freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Godlike Hacks

$GDHK

14/03/2023

TOKEN OVERVIEW

Fees

- Buy fees: 5%
- Sell fees: 5%

Fees privileges

- Can change fees up to 100%

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 fees

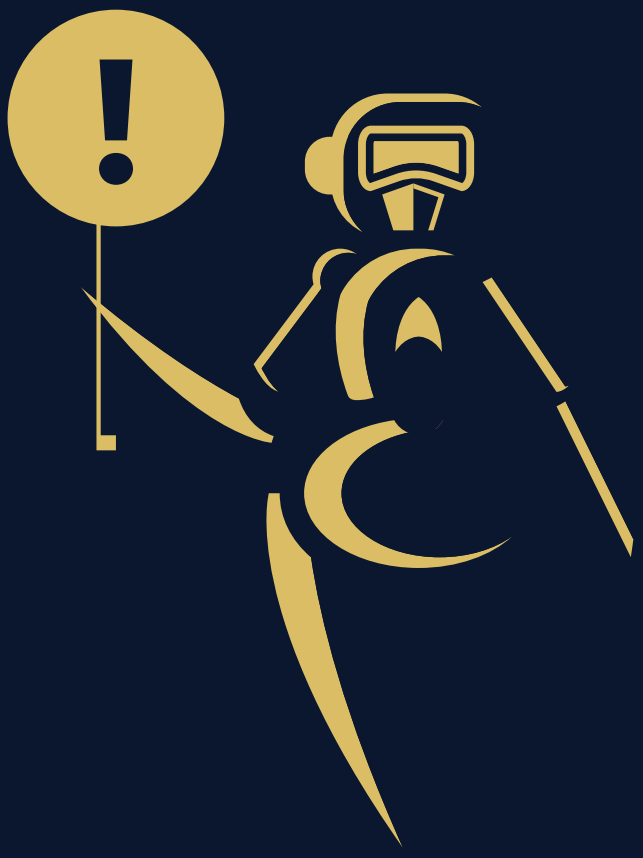
TABLE OF CONTENTS

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

0xe6dcB9A749D0513D4C9889659D5D4116db19Ef08

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 14/03/2023

WEBSITE DIAGNOSTIC

https://www.godlikehacks.com/


0-49


50-89


90-100

Performance

Accessibility

Best Practices

SEO

Progressive Web App

Socials


Twitter

https://twitter.com/GodlikeHacks


Telegram

https://t.me/GodlikehacksPortal

AUDIT OVERVIEW

72

Security Score
HIGH RISK
Audit FAIL

93

Static Scan
Automatic scanning for common vulnerabilities

88

ERC Scan
Automatic checks for ERC's conformance

2	High
3	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't exclude an address from transactions
- Contract owner can exclude wallet from tax

```
function updateExcludedFromFees(address _address, bool state) external onlyOwner {
    excludedFromFees[_address] = state;
}
```

- Contract owner can change swap settings (without threshold)

```
function setSwapEnabled(bool state) external onlyOwner {
    swapEnabled = state;
}

function setSwapThreshold(uint256 new_amount) external onlyOwner {
    swapThreshold = new_amount * 10**decimals();
}
```

- Contract owner has to call confirmLpFilled function to enable trade

```
function confirmLpFilled() external onlyOwner{
    lpFilled = true;
    swapEnabled = true;
}
.
.
.
require(lpFilled || excludedFromFees[sender] || excludedFromFees[recipient], "Trading disabled");
.
.
.
```

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

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

```
function setMaxBuy(uint256 amount) external onlyOwner{
    maxBuy = amount * 10**decimals();
}
```

Contract owner can change fees up to 100%

```
function setBuyTaxes(uint256 _marketing, uint256 _dev, uint256 _liquidity) external onlyOwner{
        buyTaxes = Taxes(_marketing, _dev, _liquidity);
        totalBuyTax = _marketing + _dev + _liquidity;
}

function setSellTaxes(uint256 _marketing, uint256 _dev, uint256 _liquidity) external onlyOwner{
        sellTaxes = Taxes(_marketing, _dev, _liquidity);
        totalSellTax = _marketing + _dev + _liquidity;
}
```

Contract owner can change marketingWallet and devFunds addresses

Current values:

marketingWallet : 0x846bc285856046804c8b331623d31bae2e16188f

devFunds : 0xd77450acef9afd7a985839c537771d5372b6c420

```
function updateDevFunds(address newWallet) external onlyOwner{
        devFunds = newWallet;
}

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

Contract owner can withdraw tokens from smart contract

except own tokens

```
function rescueBEP20(address tokenAddress, uint256 amount) external onlyOwner{
        require(tokenAddress != address(this), "Can't take self token");
        IERC20(tokenAddress).transfer(owner(), amount);
}

function rescueBNB(uint256 weiAmount) external onlyOwner{
        payable(owner()).sendValue(weiAmount);
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        _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 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:	5%
Sell fees:	5%
Max TX:	0
Max Sell:	N/A

Honeypot Risk

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

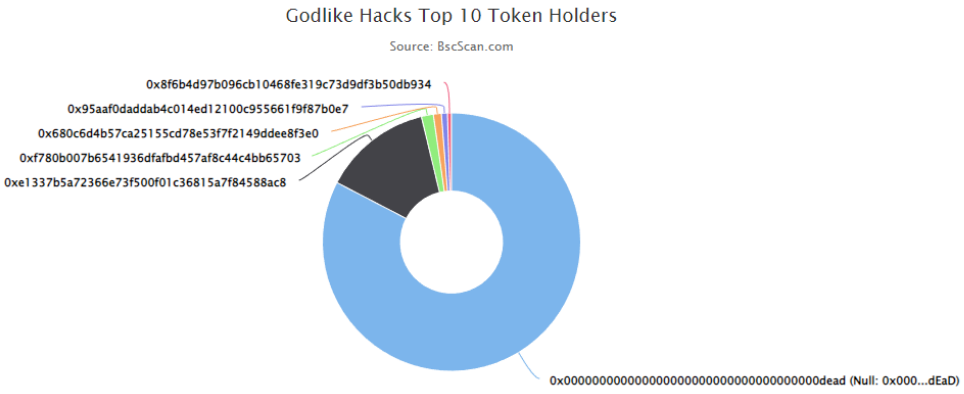
Rug Pull Risk

Liquidity:	N/A
Holders:	Unlocked tokens

GDHK TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000,000.00 Tokens) of Godlike Hacks

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 6



(A total of 100,000,000,000.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Null: 0x000...dEaD	82,633,000,000	82.6330%
2	0xe1337b5a72366e73f500f01c36815a7f84588ac8	13,617,000,000	13.6170%
3	0xf780b007b6541936dfafbd457af8c44c4bb65703	1,500,000,000	1.5000%
4	0x680c6d4b57ca25155cd78e53f7f2149ddee8f3e0	1,000,000,000	1.0000%
5	0x95aaf0daddab4c014ed12100c955661f9f87b0e7	750,000,000	0.7500%
6	0x8f6b4d97b096cb10468fe319c73d9df3b50db934	500,000,000	0.5000%

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.