freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Genshin King

$GENK

03/04/2022

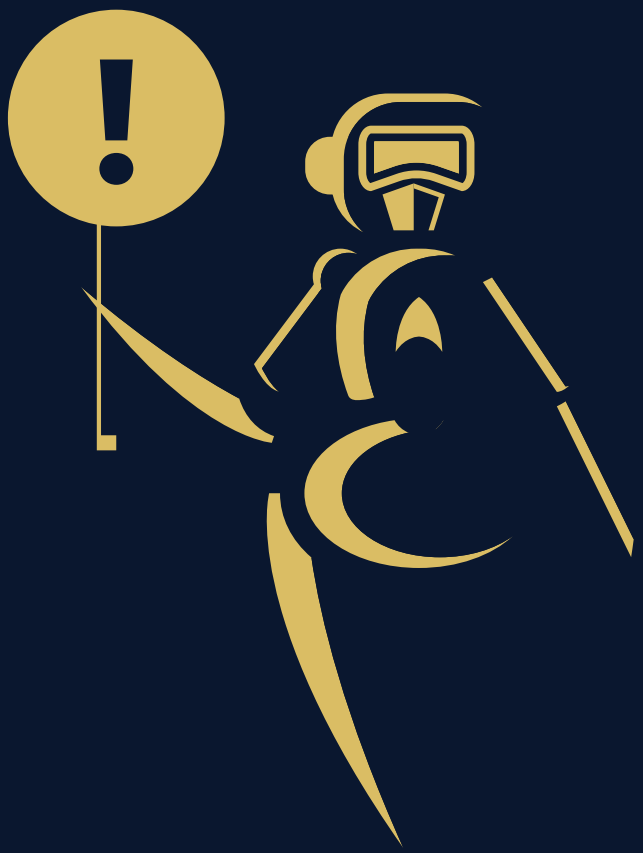
TABLE OF CONTENTS

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

0xce4C89B1E8512941cD594d0d6C1d628d1dA67f37

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 03/04/2022

AUDIT OVERVIEW

98

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

98

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 mint tokens after initial contract deploy

Contract owner can't exclude an address from transactions

Contract owner can exclude/include wallet from tax

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

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

Contract owner can change cooldown value between trades

```
function setCoolDownTime(uint256 timeForContract) external onlyOwner {
        require(swapCoolDownTime != timeForContract);
        swapCoolDownTime = timeForContract;
        emit UpdatedCoolDowntime(timeForContract);
}
```

Contract owner can change swap settings

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

Contract owner can set max tx amount

```
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
        _maxTxAmount = maxTxAmount;
        emit UpdatedMaxTxAmount(maxTxAmount);
}
```

Contract owner can change fees up to 100%

```
function setFees(uint256 bnbFee, uint256 liquidityFee) external onlyOwner() {
        require(_BNBFee != bnbFee || _liquidityFee != liquidityFee);
        _BNBFee = bnbFee;
        _liquidityFee = liquidityFee;
        emit UpdateFees(bnbFee, liquidityFee);
}
```

Contract owner can change bnbPoolAddress address

Current value:

bnbPoolAddress : 0xe655abca3199058d53b1cf7f273ef905d41ad03f

```
function setBNBPoolAddress(address account) external onlyOwner {
    require(account != bnbPoolAddress, 'This address was already used');
    bnbPoolAddress = account;
    emit UpdateBNBPoolAddress(account);
}
```

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:	12%
Sell fees:	12%
Max TX:	200,000
Max Sell:	N/A

Honeypot Risk

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

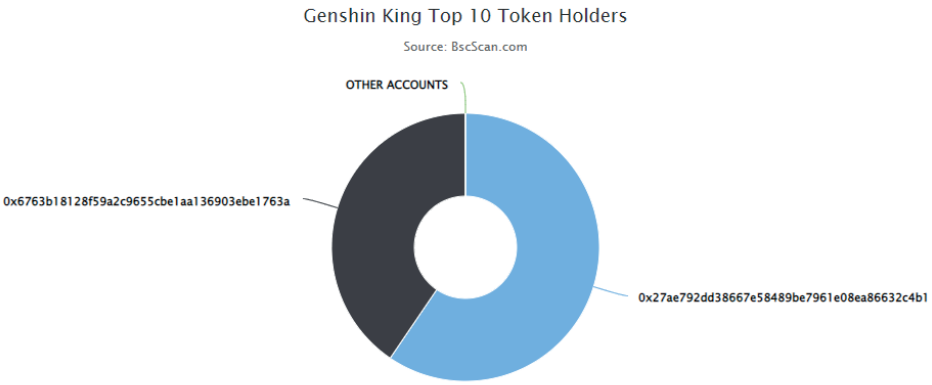
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

GENSHIN KING TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

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

Token Total Supply: 100,000,000.00 Token | Total Token Holders: 2



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

Rank	Address	Quantity (Token)	Percentage
1	0x27ae792dd38667e58489be7961e08ea86632c4b1	59,499,991.4	59.5000%
2	0x6763b18128f59a2c9655cbe1aa136903ebe1763a	40,500,008.6	40.5000%

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.