freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

horgi

$HORGI

11/01/2022

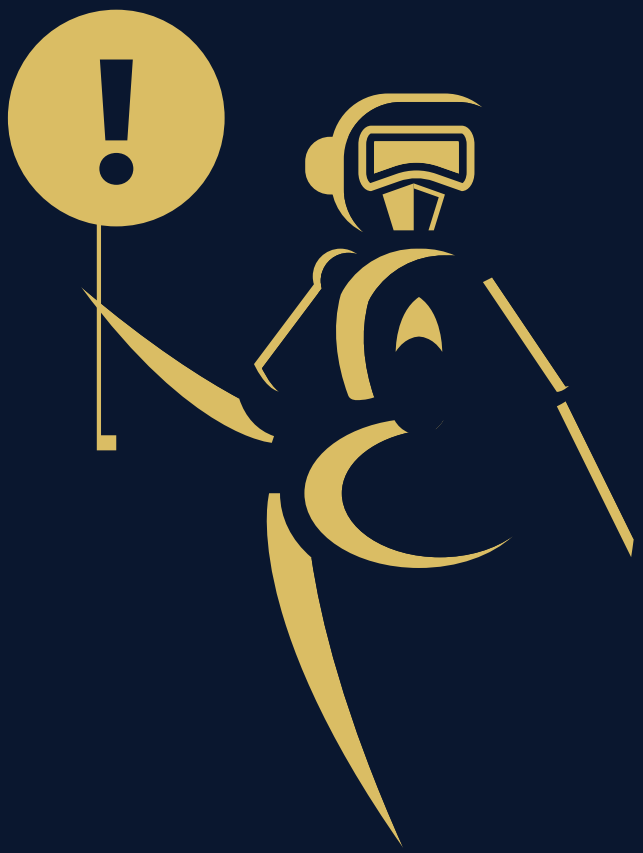
TABLE OF CONTENTS

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

0x4F5C381861333097AFA97E98a8D6DC0eB0D69ec4

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 11/01/2022

WEBSITE DIAGNOSTIC

https://horgitoken.org/

0-49 50-89 90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
88	90	100	86	NA

Metrics

First Contentful Paint
2.3 s

Time to interactive
3.8 s

Speed Index
3.6 s

Total Blocking Time
220 ms

Large Contentful Paint
7.8 s

Cumulative Layout Shift
0

WEBSITE IMPROVEMENTS

Properly size images

Eliminate render-blocking resources

Reduce the impact of third-party code Third-party code blocked the main thread for 340 ms

Image elements do not have explicit width and height

Image elements do not have [alt] attributes

Background and foreground colors do not have a sufficient contrast ratio

[id] attributes on active, focusable elements are not unique

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 fee

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

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

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(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");
    _transferOwnership(newOwner);
}
```

Contract owner can change max tx & max wallet amount

```
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
    require(maxTxAmount > _tTotal.div(10000), "Amount must be greater than 0.01% of supply");
    require(maxTxAmount <= _tTotal, "Amount must be less than or equal to totalSupply");
    _maxTxAmount = maxTxAmount;
    emit MaxTxAmountUpdated(_maxTxAmount);
}
.
.
.
function setMaxWalletAmount(uint256 maxWalletAmount) external onlyOwner() {
    require(maxWalletAmount > _tTotal.div(200), "Amount must be greater than 0.5% of supply");
    require(maxWalletAmount <= _tTotal, "Amount must be less than or equal to totalSupply");
    _maxWalletAmount = maxWalletAmount;
    emit MaxWalletAmountUpdated(_maxWalletAmount);
}
```

Contract owner can change the fees

```
function setTaxes(uint256 marketingFee, uint256 liquidityFee, uint256 teamFee, uint256 stakingPoolFee)
external onlyOwner() {
    uint256 totalFee = marketingFee.add(liquidityFee).add(teamFee).add(stakingPoolFee);
    require(totalFee <= 15, "Sum of fees must be less than or equals to 15");

    _marketingFee = marketingFee;
    _liquidityFee = liquidityFee;
    _teamFee = teamFee;
    _stakingPoolFee = stakingPoolFee;

    _previousMarketingFee = _marketingFee;
    _previousLiquidityFee = _liquidityFee;
    _previousTeamFee = _teamFee;
    _previousStakingPoolFee = _stakingPoolFee;

    uint256 totalBNBfees = _marketingFee.add(_teamFee).add(_stakingPoolFee);

    _marketingPercent = (_marketingFee.mul(1000)).div(totalBNBfees);
    _teamPercent = (_teamFee.mul(1000)).div(totalBNBfees);
    _stakingPoolPercent = (_stakingPoolFee.mul(1000)).div(totalBNBfees);

    emit FeesUpdated(_marketingFee, _liquidityFee, _teamFee, _stakingPoolFee);
  }
```

Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    require(swapAndLiquifyEnabled != _enabled, "Value already exists!");
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

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:	10%
Sell fees:	10%
Max TX:	100000000000000000000000000000
Max Sell:	N/A

Honeypot Risk

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

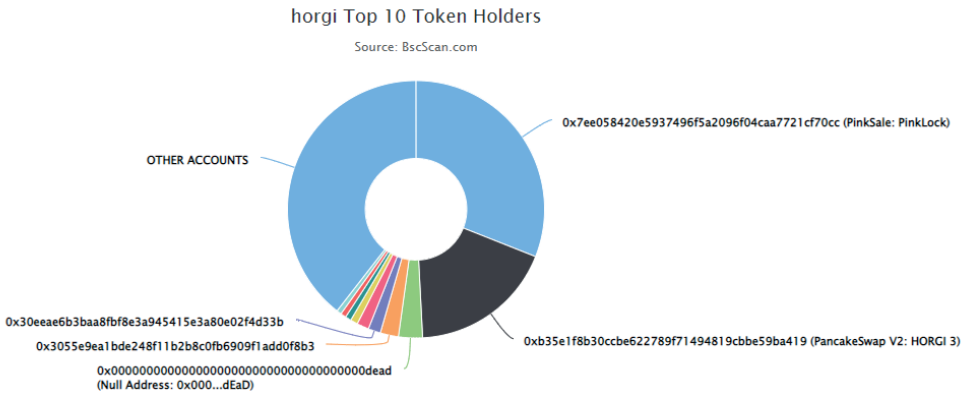
Rug Pull Risk

Liquidity:	18.2499% Locked (Unlock Date: 2023.01.06 20:02 UTC / in a year)
Holders:	Clean

HORGI TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 60.54% (60,541,357,429.32 Tokens) of horgi

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 941



(A total of 60,541,357,429.32 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	PinkSale: PinkLock	31,000,000,000	31.0000%
2	PancakeSwap V2: HORGI 3	18,193,751,036.066608278653143465	18.1938%
3	Null Address: 0x000...dEaD	3,000,000,000	3.0000%
4	0x3055e9ea1bde248f11b2b8c0fb6909f1add0f8b3	2,300,000,000	2.3000%
5	0x30eeae6b3baa8fbf8e3a945415e3a80e02f4d33b	1,557,081,927.876064930961886393	1.5571%
6	0x45487a5e8131bef4133a6884420c40ad23efa1fd	1,509,377,463.10391036324758554	1.5094%
7	0x4eebe388eb7e35474ad858d0f9c3754a3c1dd39a	929,520,967.013385411466302889	0.9295%
8	0x49513bc6590c3ed764ed115e14ef2b258b0fda17	747,035,100.0027	0.7470%
9	0x1c25243210d8b64c35fe776e390b82f126b77b90	704,066,897.191258494460323891	0.7041%
10	0xc480bf019ab71a6dc6984f5eb02c622ae62e2100	600,524,038.06778	0.6005%

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.