freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Moon Rocket Cash

$MRCH

02/01/2022

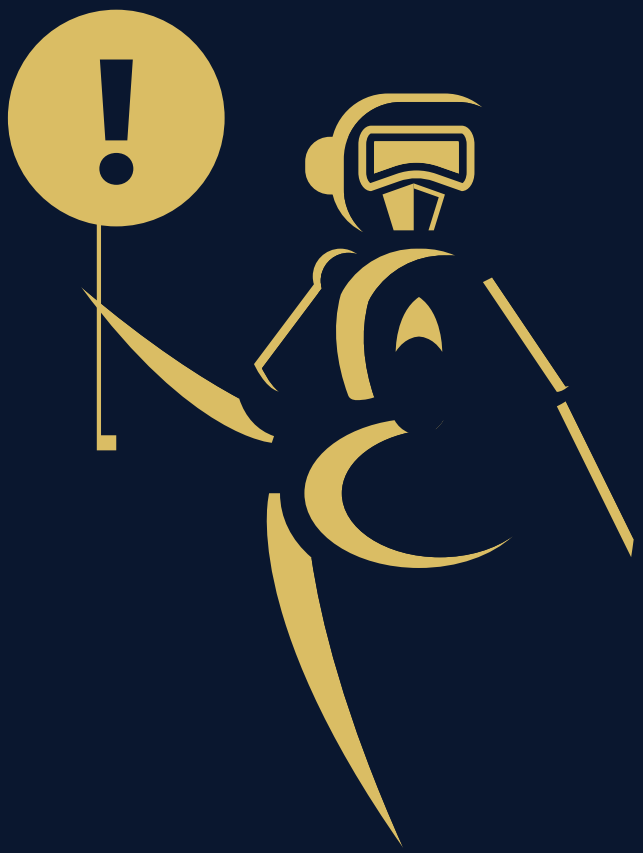
TABLE OF CONTENTS

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
Moon Rocket Cash (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xf52ec7fb30e9febddb00dfceeb9f21dcf0280ecf

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 02/01/2022

WEBSITE DIAGNOSTIC

https://moonrocketcoin.net/

0-49 50-89 90-100

72	98	86	93	NA
Performance	Accessibility	Best Practices	SEO	Progressive Web App

Metrics

First Contentful Paint
1.2 s

Time to interactive
2.5 s

Speed Index
6.4 s

Total Blocking Time
22 ms

Large Contentful Paint
12.1 s

Cumulative Layout Shift
0

WEBSITE IMPROVEMENTS

Serve images in next-gen formats

Reduce unused JavaScript

Reduce initial server response time

Reduce unused CSS

Eliminate render-blocking resources

Ensure text remains visible during webfont load

Reduce the impact of third-party code

Does not use passive listeners to improve scrolling performance.

AUDIT OVERVIEW

96

Security Score

94

Static Scan
Automatic scanning for common vulnerabilities

98

ERC Scan
Automatic checks for ERC's conformance

0 High

0 Medium

1 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 change cooldown time between transactions

```
function changeCooldownSettings(bool newStatus, uint256 newInterval) external onlyOwner {
        cooldownEnabled = newStatus;
        cooldownTimerInterval = newInterval;
    }
```

Contract owner can change swapTokensAtAmount

```
function SetSwapTokensAtAmount(uint256 _newAmount) external onlyOwner {
                swapTokensAtAmount = _newAmount * (10**18);
}
```

Contract owner can change setMaxWalletTokend

```
function setMaxWalletTokend(uint256 _maxToken) external onlyOwner {
                maxWalletToken = _maxToken * (10**18);
 }
```

Contract owner can change marketingWalletAddress

```
function setMarketingWallet(address payable wallet) external onlyOwner{
        _marketingWalletAddress = wallet;
}
```

Contract owner can set taxes (Without limit)

```
function updateFees(uint256 rewardFee, uint256 _liquidityFee, uint256 _marketingFee, uint256 _MRCBurnFee,
uint256 _mrchBurnFee) external onlyOwner{
        MRCRewardsFee = rewardFee;
        LiquidityFee = _liquidityFee;
        MarketingFee = _marketingFee;
        MRCBurnFee = _MRCBurnFee;
        MRCHBurnFee = _mrchBurnFee;
        totalFees = MRCRewardsFee.add(LiquidityFee).add(MarketingFee).add(MRCBurnFee).add(MRCHBurnFee);
}
```

Contract owner can disable trade for a specific address (blacklist)

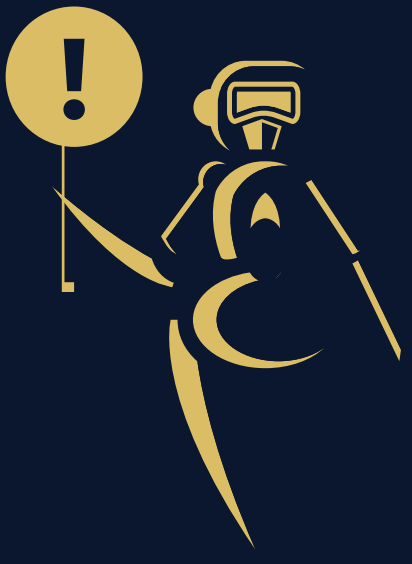
```
function blacklistAddress(address account, bool value) external onlyOwner{
        _isBlacklisted[account] = value;
}
```

Contract owner can exclude from fees a wallet or multiple addresses.

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
        require(_isExcludedFromFees[account] != excluded,  "MRCH: Account is already the value of 'excluded'");
        _isExcludedFromFees[account] = excluded;

        emit ExcludeFromFees(account, excluded);
}

function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
        for(uint256 i = 0; i < accounts.length; i++) {
            _isExcludedFromFees[accounts[i]] = excluded;
        }

        emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

Contract owner can exclude from dividends a wallet

```
function excludeFromDividends(address account) external onlyOwner {
        require(!excludedFromDividends[account]);
        excludedFromDividends[account] = true;

        _setBalance(account, 0);
        tokenHoldersMap.remove(account);

        emit ExcludeFromDividends(account);
}
```



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 low issue during the first review.

TOKEN DETAILS

Details

Buy fees:	12%
Sell fees:	12%
Max TX:	N/A
Max Sell:	N/A

Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

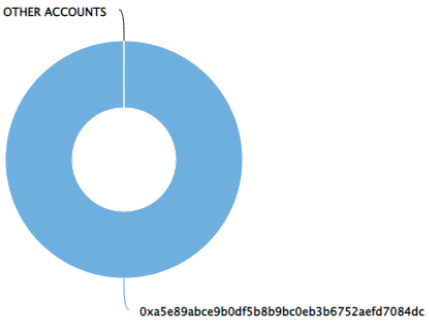
MOON ROCKET CASH TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 100 holders collectively own 100.00%
(1,000,000,000,000,000.00 Tokens) of Moon Rocket Cash

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

Moon Rocket Cash Top 100 Token Holders

Source: BscScan.com



(A total of 1,000,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xa5e89abce9b0df5b8b9bc0eb3b6752aefd7084dc	1,000,000,000,000,000	100.0000%

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.