freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

MiniDoge2022

$MD22

03/01/2022

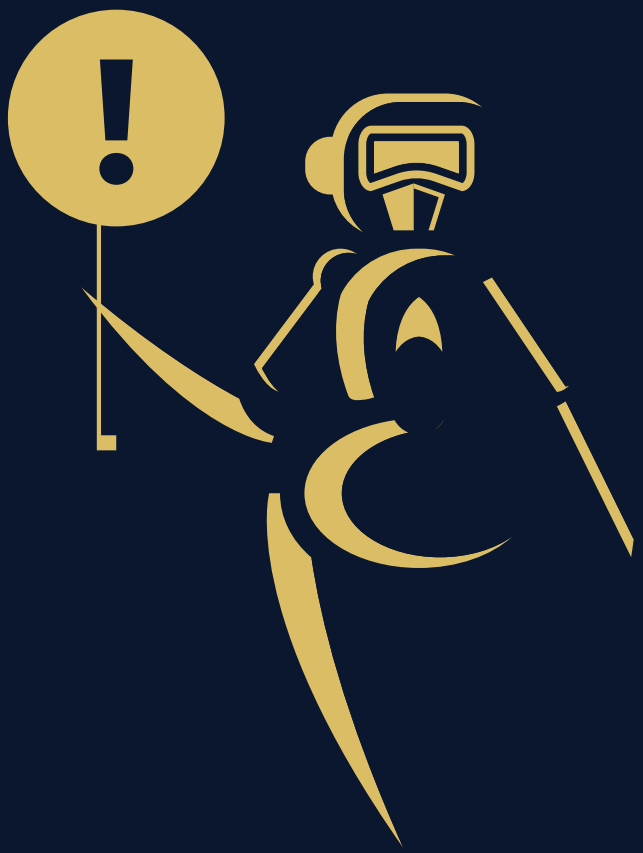
TABLE OF CONTENTS

DISCLAIMER

The information provided on this analysis decoument 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 MiniDoge2022 (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xbdfa2231e5944f5fe21491aec4d303ff78a95f61

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 03/01/2022

WEBSITE DIAGNOSTIC

https://minidoge2022.info/

0-49 50-89 90-100

82	95	100	92	NA
Performance	Accessability	Best Practices	SEO	Progressive Web App

Metrics

First Contentful Paint
1.2 s

Time to interactive
3.8 s

Speed Index
4.0 s

Total Blocking Time
0 ms

Large Contentful Paint
4.1 s

Cumulative Layout Shift
0

Issues found

Eliminate render-blocking resources

Reduce unused CSS

Reduce unused JavaScript

Reduce the impact of third-party code - Third-party code blocked the main thread for 450 ms

Image elements do not have explicit width and height

Background and foreground colors do not have a sufficient contrast ratio.

Heading elements are not in a sequentially-descending order

Document does not have a meta description

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

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 change cooldown time between transactions

```
function cooldownEnabled(bool _status, uint8 _interval) public onlyOwner {
    buyCooldownEnabled = _status;
    cooldownTimerInterval = _interval;
}
```

Contract owner can change fee receivers addresses

```
function setFeeReceivers(address _autoLiquidityReceiver, address _marketingFeeReceiver, address
_buybackFeeReceiver) external authorized {
    autoLiquidityReceiver = _autoLiquidityReceiver;
    marketingFeeReceiver = _marketingFeeReceiver;
    buybackFeeReceiver = _buybackFeeReceiver;
}
```

Contract owner can change taxes

```
function setFees(uint256 _liquidityFee, uint256 _buybackFee, uint256 _marketingFee, uint256 _devFee,
uint256 _feeDenominator) external authorized {
    liquidityFee = _liquidityFee;
    buybackFee = _buybackFee;
    devFee = _devFee;
    marketingFee = _marketingFee;
    totalFee = _liquidityFee.add(_buybackFee).add(_devFee).add(_marketingFee);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator/5);
    require(devFee > 2);
}
```

Contract owner can exclude from fees a wallet

```
function setIsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

Contract owner can set tx limit for a specific address

```
function setIsTxLimitExempt(address holder, bool exempt) external authorized {
    isTxLimitExempt[holder] = exempt;
}
```

Contract owner can set max wallet percent for a specific address

```
function setMaxWalletPercent(uint256 maxWallPercent) external onlyOwner() {
    _maxWalletToken = (_totalSupply * maxWallPercent ) / 100;
}
```

Contract owner can change swap settings

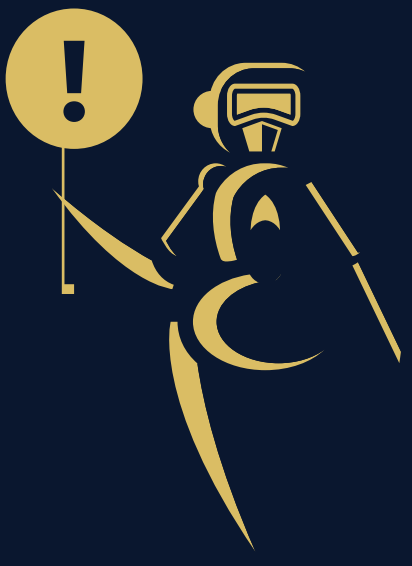
```
function setSwapBackSettings(bool _enabled, uint256 _amount) external authorized {
    swapEnabled = _enabled;
    swapThreshold = _amount;
}
```

Contract owner can change tx limit

```
function setTxLimit(uint256 amount) external authorized {
    _maxTxAmount = amount;
}
```

Contract owner can change sell multiplier

```
function set_sell_multiplier(uint256 Multiplier) external onlyOwner{
    sellMultiplier = Multiplier;
}
```



CONCLUSION AND ANALYSIS

Smart Contracts within the scope were manually reviewd 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:	15%
Max TX:	1%
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

MINIDOGE2022 TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 100 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of MiniDoge2022

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

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

Rank	Address	Quantity (Token)	Percentage
1	0x4ce471ddf977c5b037099862facd16d493de7869	1,000,000,000,000	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.