freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Wizardz V2

$WIZ

10/08/2022

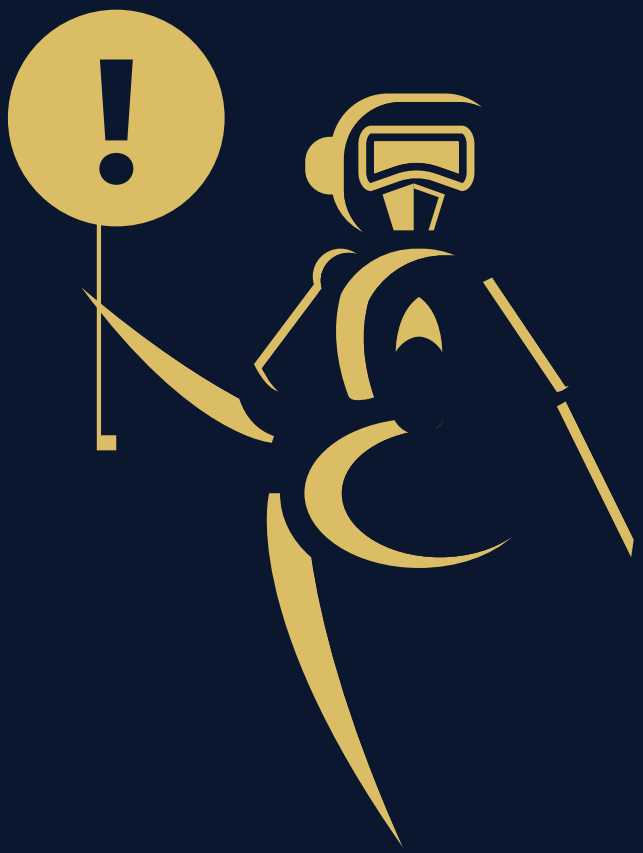
TABLE OF CONTENTS

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 Wizardz V2 (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x034Daf02cDFee63A35F20D03C2f81080A12B0A7D

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 10/08/2022

AUDIT OVERVIEW

82

Security Score

76

Static Scan
Automatic scanning for common vulnerabilities

90

ERC Scan
Automatic checks for ERC's conformance

5 High

6 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 exclude an address from transactions

```
function enable_blacklist(bool _status) public onlyOwner {
    blacklistMode = _status;
}

function manage_blacklist(address[] calldata addresses, bool status) public onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
        isBlacklisted[addresses[i]] = status;
    }
}
```

- Contract owner can exclude/include wallet from tax

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

- Contract owner can exclude/include wallet from tx limitations

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

- Contract owner can change max wallet amount (with threshold)

```
function setMaxWallet(uint256 amount) external onlyOwner {
    require(amount >= _totalSupply / 1000 );
    _maxWalletSize = amount;
}
```

- Contract owner can change max tx amount

```
function setTxLimit(uint256 amountBuy) external onlyOwner {
    _maxTxAmount = amountBuy;
}
```

- Contract owner can enable/disable cooldown between trades

Current value (interval): 5 uint8

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

Contract owner can enable/disable additional tax for certain wallets

```
.
.
.
uint256 multiplier = isSell ? _sellMultiplier : 100; //dont touch this section
if(taxMode && !istaxed[receiver] && !isSell){
    multiplier = 800;
}
.
.
.
```

Normal multiplier on buy is 1 (value 100), if taxMode is enabled, for wallets included in istaxed array multiplier will be 8

```
function enable_tax(bool _status) public onlyOwner {
    taxMode = _status;
}

function manage_tax(address[] calldata addresses, bool status) public onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
        istaxed[addresses[i]] = status;
    }
}
```

Contract owner can set sell multiplier (without threshold)

```
function setSellMultiplier(uint256 multiplier) external onlyOwner{
    _sellMultiplier = multiplier;
}
```

Contract owner can enable/disable trade

```
function tradingstatus(bool state) public onlyOwner {
    tradingOpen = state;
}

function OpenTrading(uint256 _swapAt, uint256 _swapDelay) public onlyOwner {
    tradingOpen = true;
    launchBlock = block.number;
    swapAt = _swapAt * (10 ** 9);
    swapDelay = _swapDelay;
}
```

Contract owner can change swap settings

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

- Contract owner can change marketingFeeReceiver and ecosystemFeeReceiver addresses

  Current values:

  marketingFeeReceiver : 0x631540fcb592903f6bd420ce8ae54c7258ea3ba6

  ecosystemFeeReceiver : 0x03a8d9308e5f9b02690fea4b2c793f4c6b8a0244

```
function setFeeReceiver(address _marketingFeeReceiver, address _ecosystemFeeReceiver) external onlyOwner {
        marketingFeeReceiver = _marketingFeeReceiver;
        ecosystemFeeReceiver = _ecosystemFeeReceiver;
}
```

- Contract owner can change fees up to 100%

```
function setFees(uint256 _liquidityFee, uint256 _marketingFee, uint256 _ecosystemFee, uint256 _feeDenomi-
nator) external onlyOwner {
        liquidityFee = _liquidityFee;
        marketingFee = _marketingFee;
        ecosystemFee = _ecosystemFee;
        totalFee = _liquidityFee.add(_marketingFee).add(_ecosystemFee);
        feeDenominator = _feeDenominator;
    }
```

- Contract owner can renounce ownership

```
function renounceOwnership() public onlyOwner {
        _setOwner(address(0));
    }
```

- Contract owner can transfer ownership

```
function transferOwnership(address payable adr) public onlyOwner {
        owner = adr;
        authorizations[adr] = true;
        emit OwnershipTransferred(adr);
    }
```

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 5 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	9%
Sell fees:	9% (more details on page 6)
Max TX:	200,000,000
Max Sell:	N/A

Honeypot Risk

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

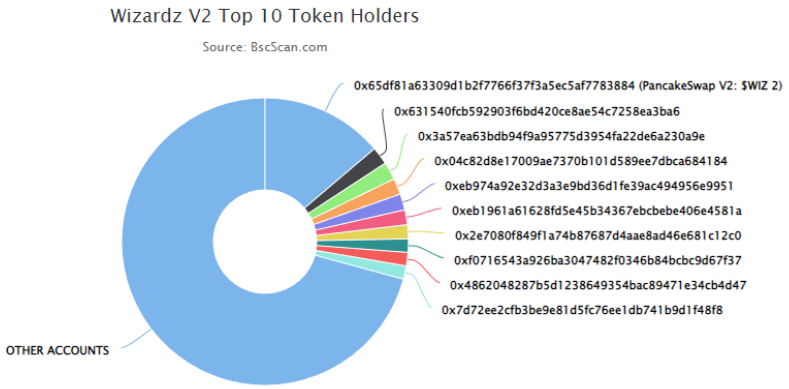
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

WIZARDZ V2 TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 29.22% (2,921,575,998.14 Tokens) of Wizardz V2

Token Total Supply: 10,000,000,000.00 Token | Total Token Holders: 377



(A total of 2,921,575,998.14 tokens held by the top 10 accounts from the total supply of 10,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: $WIZ 2	1,381,517,602.095932079	13.8152%
2	0x631540fcb592903f6bd420ce8ae54c7258ea3ba6	200,000,000	2.0000%
3	0x3a57ea63bdb94f9a95775d3954fa22de6a230a9e	199,969,017.114883689	1.9997%
4	0x04c82d8e17009ae7370b101d589ee7dbca684184	182,000,000	1.8200%
5	0xeb974a92e32d3a3e9bd36d1fe39ac494956e9951	182,000,000	1.8200%
6	0xeb1961a61628fd5e45b34367ebcbebe406e4581a	166,133,358.010301199	1.6613%
7	0x2e7080f849f1a74b87687d4aae8ad46e681c12c0	157,395,038.019168574	1.5740%
8	0xf0716543a926ba3047482f0346b84bcbc9d67f37	152,560,982.899885648	1.5256%
9	0x4862048287b5d1238649354bac89471e34cb4d47	150,000,000	1.5000%
10	0x7d72ee2cfb3be9e81d5fc76ee1db741b9d1f48f8	150,000,000	1.5000%

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.