freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

PyramiDaoToken

$PYRA

15/04/2022

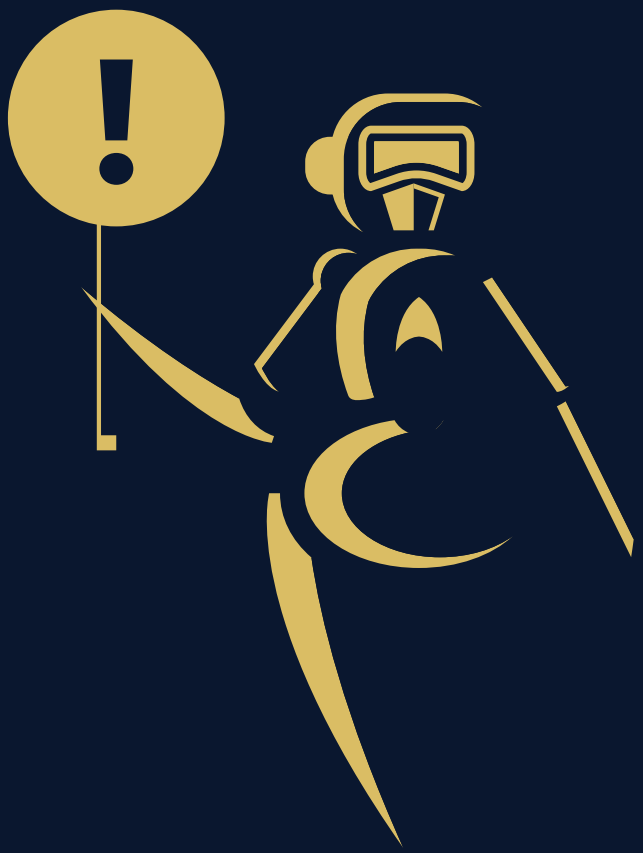
TABLE OF CONTENTS

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

0xf62D479c63Ba5196FCE6F22210080F9a2D6a4D47

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 15/04/2022

AUDIT OVERVIEW

97

Security Score

96

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 exclude/include an address from transactions

```
function updateBotBlacklist(address _user, bool _flag) public onlyOwner{
        blacklist[_user] = _flag;
}
```

Contract owner can change rebase settings

```
function rebase(int256 supplyDelta)
        external
        onlyRelayer
        returns (uint256)
    {
        if (shouldRebase()) {
            return _rebase(supplyDelta);
        }
}

function setRebaseTime(uint256 _rebaseTime) external onlyOwner {
        rebaseTime = _rebaseTime;
}

function setLastRebasedTime(uint256 lastRebasedTime_)
        public
        onlyOwner
    {
        lastRebasedTime = lastRebasedTime_;
}

function setPercentInOneRebaseRound(uint256 percentInOneRebaseRound_)
        public
        onlyOwner
    {
        percentInOneRebaseRound = percentInOneRebaseRound_;
}
```

Contract owner can change swap settings

```
function setSwapBackSettings(
        bool _enabled,
        uint256 _num,
        uint256 _denom
    ) external onlyOwner {
        swapEnabled = _enabled;
        pydSwapThreshold = TOTAL_PYDS.div(_denom).mul(_num);
}
```

Contract owner can exclude wallet from fees

```
function setFeeExempt(address _addr) external onlyOwner {
    _isFeeExempt[_addr] = true;
}
```

Contract owner can change liquidityFee, Insurance, Treasury up to 25% and sellFee up to 100%

```
function setFees(
    uint256 _liquidityFee,
    uint256 _Insurance,
    uint256 _Treasury,
    uint256 _sellFee,
    uint256 _feeDenominator
  ) external onlyOwner {
    liquidityFee = _liquidityFee;
    Insurance = _Insurance;
    Treasury = _Treasury;
    sellFee = _sellFee;
    totalFee = liquidityFee.add(Treasury).add(Insurance);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator / 4);
}
```

Contract owner can change _autoLiquidityReceiver, _EgyptTreasury, _pharaohsInsurance, AutoLiquidityReceiver, EgyptTreasury, PharaohsInsurance and pairAddress addresses

```
function setFeeReceivers(
    address _autoLiquidityReceiver,
    address _EgyptTreasury,
    address _pharaohsInsurance
  ) external onlyOwner {
    AutoLiquidityReceiver = _autoLiquidityReceiver;
    EgyptTreasury = _EgyptTreasury;
    PharaohsInsurance = _pharaohsInsurance;
}

function setLP(address _address) external onlyOwner {
    pairContract = ILiquidityPool(_address);
    _isFeeExempt[_address];
}

function setAutoLiquidityReceiver(address _AutoLiquidityReceiver) external onlyOwner {
    AutoLiquidityReceiver = _AutoLiquidityReceiver;
}

function setEgyptTreasury(address _EgyptTreasury) external onlyOwner {
    EgyptTreasury = _EgyptTreasury;
}

function setPharaohsInsurance(address _PharaohsInsurance) external onlyOwner {
    PharaohsInsurance = _PharaohsInsurance;
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public onlyOwner {
    emit OwnershipRenounced(_owner);
    _owner = address(0);
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public onlyOwner {
    _transferOwnership(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:	8%
Sell fees:	13%
Max TX:	N/A
Max Sell:	N/A

Honeypot Risk

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

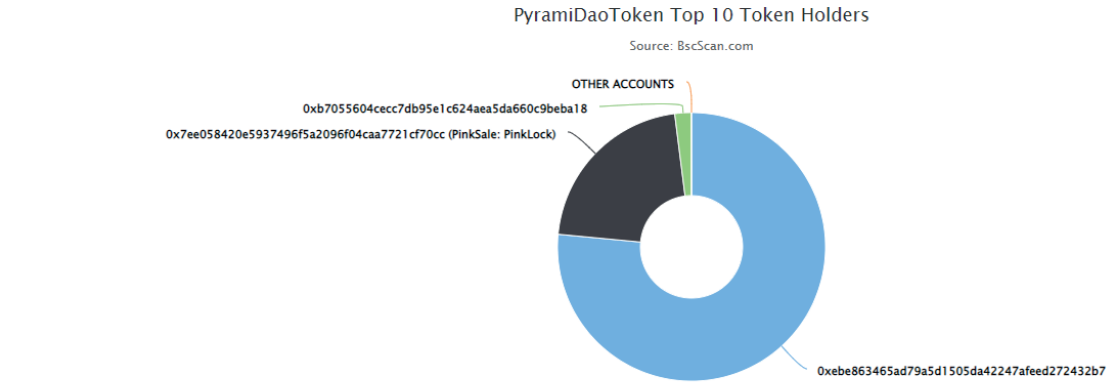
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

PYRAMIDAOTOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000.00 Tokens) of PyramiDaoToken

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 3



(A total of 1,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xebe863465ad79a5d1505da42247afeed272432b7	765,000,000	76.5000%
2	PinkSale: PinkLock	215,000,000	21.5000%
3	0xb7055604cecc7db95e1c624aea5da660c9beba18	20,000,000	2.0000%

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.