



freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



ZAT Project
\$ZPRO



11/03/2022



TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-7 OWNER PRIVILEGES
- 8 CONCLUSION AND ANALYSIS
- 9 TOKEN DETAILS
- 10 ZAT PROJECT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 11 TECHNICAL DISCLAIMER



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

0xc8c488fDbBB2E72E41710Ade67784f0812160210

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 11/03/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



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't mint tokens after initial contract deploy

Contract owner can exclude/include wallet from tax

```
function S01_includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}

function S02_excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}
```

Contract owner can exclude/include wallet from rewards

```
function S03_includeInReward(address account) external onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}

function S04_excludeFromReward(address account) public onlyOwner() {
    // require(account != 0x7a250d5630B4cf539739dF2C5dAcb4c659F2488D, 'We can not exclude Uniswap router');
    require(!_isExcluded[account], "Account is already excluded");
    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

Contract owner can change max tx amount

```
function A_setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(10**_decimals);
    emit MaxTxAmountUpdated(_maxTxAmount);
}
```

Contract owner can enable/disable trading

```
function E4_enableTrading(bool truefalse) external onlyOwner() {
    tradingEnabled = truefalse;
}
```

Contract owner can change the fees up to 100%

```
function F01_setMarketingFee(uint256 marketingfee) external onlyOwner() {
    _marketingFee = marketingfee;
}
function F02_setTreasuryfee(uint256 treasuryfee) external onlyOwner() {
    _treasuryFee = treasuryfee;
}
function F06_setBuyTaxfee(uint256 buyTaxFee) external onlyOwner() {
    _buyTaxFee = buyTaxFee;
}
function F02_setLiqfee(uint256 liqfee) external onlyOwner() {
    _liqFee = liqfee;
}
function F02_setcharityfee(uint256 charityfee) external onlyOwner() {
    _charityFee = charityfee;
}
function F07_setTaxFee(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}
function F08_setProjectFee(uint256 ProjectFee) external onlyOwner() {
    _liquidityFee = ProjectFee;
}
function F09_setSellTaxFee(uint256 sellTaxFee) external onlyOwner() {
    _sellTaxFee = sellTaxFee;
}
function F11_setTransferTaxFee(uint256 transferTaxFee) external onlyOwner() {
    _transferTaxFee = transferTaxFee;
}
function F12_setBuyProjectFee(uint256 buyProjectFee) external onlyOwner() {
    _buyLiquidityFee = buyProjectFee;
}
function F13_setSellProjectFee(uint256 sellProjectFee) external onlyOwner() {
    _sellLiquidityFee = sellProjectFee;
}
function F15_setTransferProjectFee(uint256 transferProjectFee) external onlyOwner() {
    _transferLiquidityFee = transferProjectFee;
}
function F16_setExchangeTaxFee(uint256 exchangeTaxFee) external onlyOwner() {
    _exchangeTaxFee = exchangeTaxFee;
}
function F17_setExchangeProjectFee(uint256 exchangeProjectFee) external onlyOwner() {
    _exchangeLiquidityFee = exchangeProjectFee;
}
```

Contract owner can exclude/include address from transactions

```
function S05_addToBlacklist(address _address) public onlyOwner {
    require(!_isBlacklisted[_address], "Address is already blacklisted");
    _isBlacklisted[_address] = true;
}

function S06_removeFromBlacklist(address _address) public onlyOwner {
    require(_isBlacklisted[_address], "Address is already whitelisted");
    _isBlacklisted[_address] = false;
}
```

Contract owner can change _marketingWallet, _treasuryWallet, _liquidityWallet and _charityWallet addresses

Current values:

_marketingWallet : 0x667f6ddf92333ff6de2a0d6064271939bae95922

_treasuryWallet : 0x5b1092bb157ffd843252511ec75c1a058075cea8

_liquidityWallet : 0xfe330d8f5a1ef9527a8e7702fd2d4889229c24d8

_charityWallet : 0x9237ac325547ed35d4a7ccf301a56df25b70a0aa

```
function W1_setMarketingWallet(address _address) public onlyOwner() {
    _marketingWallet = _address;
}

function W2_setTreasuryWallet(address _address) public onlyOwner() {
    _treasuryWallet = _address;
}

function W3_setLiquidityWallet(address _address) public onlyOwner() {
    _liquidityWallet = _address;
}

function W4_setcharityWallet(address _address) public onlyOwner() {
    _charityWallet = _address;
}
```

Contract owner can enable/disable tax

```
function E1_enableFees(bool truefalse) external onlyOwner() {
    feesEnabled = truefalse;
}
```

Contract owner can transfer ownership

```
function Z_transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = 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 1 LOW issue during the first review.

TOKEN DETAILS

Details

Buy fees:	10%
Sell fees:	15%
Max TX:	100,000,000
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



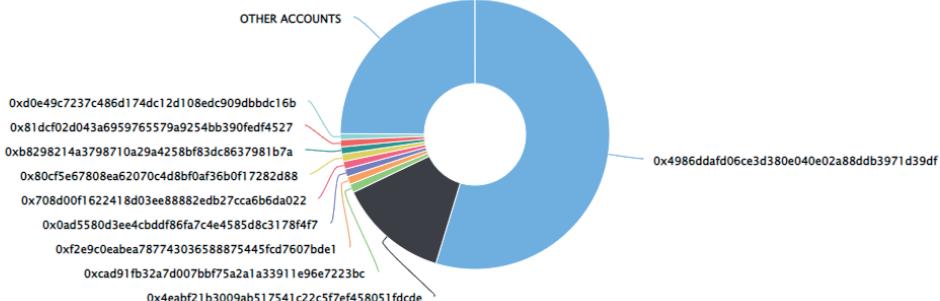
ZAT PROJECT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 75.09% (75,086,950.20 Tokens) of ZAT Project

Token Total Supply: 100,000,000.00 Token | Total Token Holders: 349

ZAT Project Top 10 Token Holders

Source: BscScan.com



(A total of 75,086,950.20 tokens held by the top 10 accounts from the total supply of 100,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x4986ddaf06ce3d380e040e02a88ddb3971d39df	54,703,930.9428714	54.7039%
2	0x4ebf21b3009ab517541c22c5f7ef458051fdcce	13,243,049.1083607	13.2430%
3	0xcad91fb32a7d007bbf75a2a1a33911e96e7223bc	1,010,614.2891639	1.0106%
4	0xf2e9c0eabea787743036588875445fcf7607bde1	950,078.0536732	0.9501%
5	0x0ad5580d3ee4cbddf86fa7c4e4585d8c3178f4f7	946,048.8045588	0.9460%
6	0x708d00f1622418d03ee88882edb27cca6b6da022	922,903.0946316	0.9229%
7	0x80cf5e67808ea62070c4db0af36b0f17282d88	883,292.7469598	0.8833%
8	0xb8298214a3798710a29a4258bf83dc8637981b7a	872,938.5071116	0.8729%
9	0x81dcf02d043a695976579a9254bb390fedf4527	850,718.2435556	0.8507%
10	0xd0e49c7237c486d174dc12d108edc909dbbd16b	703,376.413794	0.7034%

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.

