freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Tindearn
$LIKE

25/05/2022

TABLE OF CONTENTS

DISCLAIMER

INTRODUCTION

FreshCoins (Consultant) was contracted by
Tindearn (Customer) to conduct a Smart Contract Code Review and
Security Analysis.

0x31eDC600F94ec8767380B16B8548EbF7464584eC

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 25/05/2022

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't mint tokens after initial contract deploy

Contract owner can't exclude an address from transactions

Contract owner can exclude/include wallet from fees

```
function excludeFromFeesDAO(address _account, bool _state) external onlyRole(DAO) {
    _exemptFromFees[_account] = _state;
}
```

Contract owner can enable/disable swap fees

```
function enableSwappingFeesDAO() external onlyRole(DAO) {
    swapFeesEnabled = true;
}

function stopSwappingFeesDAO() external onlyRole(DAO) {
    swapFeesEnabled = false;
}
```

Contract owner can withdraw BNB stored and ERC20 tokens from smart contract

```
function withdrawDAO(uint256 _amount) external onlyRole(DAO) {
    payable(msg.sender).transfer(_amount);
}

function withdrawERC20DAO(address _erc20, uint256 _amount) external onlyRole(DAO) {
    IERC20(_erc20).transfer(msg.sender, _amount);
}
```

Contract owner can change buy fees up to 15%

```
function setBuyingFeeDAO(uint256 _buyingFee) external onlyRole(DAO) {
    require(_buyingFee <= 150, "The buying fee cannot be set to a number above 15%");
    buyingFee = _buyingFee;
}
```

Contract owner can change sell fees up to 15%

```
function setSellingFeeDAO(uint256 _sellingFee) external onlyRole(DAO) {
    require(_sellingFee <= 150, "The selling fee cannot be set to a number above 15%");
    sellingFee = _sellingFee;
}
```

Contract owner can change marketingAddress address

```
function setMarketingAddressDAO(address payable _marketingAddress) external onlyRole(DAO) {
    marketingAddress = _marketingAddress;
}
```

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:	0%
Sell fees:	7%
Max TX:	N/A
Max Sell:	N/A

Honeypot Risk

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

Rug Pull Risk

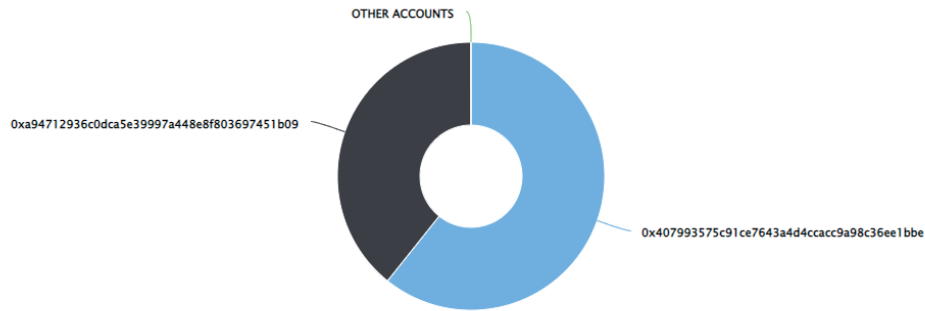|	
Liquidity:	N/A
Holders:	Clean

TINDEARN TOKEN TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (200,000,000.00 Tokens) of Tindearn   Token Total Supply: 200,000,000.00 Token | Total Token Holders: 2

Tindearn Top 10 Token Holders
Source: BscScan.com

OTHER ACCOUNTS

0xa94712936c0dca5e39997a448e8f803697451b09

0x407993575c91ce7643a4d4ccacc9a98c36ee1bbe

(A total of 200,000,000.00 tokens held by the top 10 accounts from the total supply of 200,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x407993575c91ce7643a4d4ccacc9a98c36ee1bbe	121,500,000	60.7500%
2	0xa94712936c0dca5e39997a448e8f803697451b09	78,500,000	39.2500%

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.