freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Starlink Coin

$Starlink

28/02/2022

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

0x518281F34dbf5B76e6cdd3908a6972E8EC49e345

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 28/02/2022

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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet from fees

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

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

Contract owner can burn tokens from any wallet

```
function burnFrom(address account, uint256 amount) public virtual {
        uint256 currentAllowance = allowance(account, _msgSender());
        require(currentAllowance >= amount, "ERC20: burn amount exceeds allowance");
        unchecked {
    _approve(account, _msgSender(), currentAllowance - amount);
    }
        _burn(account, amount);
}

function _burn(address account, uint256 amount) internal virtual {
        require(account != address(0), "ERC20: burn from the zero address");

        _beforeTokenTransfer(account, address(0), amount);

        uint256 accountBalance = _balances[account];
        require(accountBalance >= amount, "ERC20: burn amount exceeds balance");
        unchecked {
    _balances[account] = accountBalance - amount;
    }
        _totalSupply -= amount;

        emit Transfer(account, address(0), amount);

        _afterTokenTransfer(account, address(0), amount);
}
```

Contract owner can change max tx amount

```
function SetMaxTxAmount(uint256 newMaxAmount) external onlyOwner {
        _maxTxAmount = newMaxAmount;
}
```

Contract owner can change the fees up to 100%

```
function SetBuyValues(uint lp,uint market, uint burn) external onlyOwner {
        LPBuyFees = lp;
        marketBuyFees = market;
        burnBuyFees = burn;
}

function SetSellValues(uint lp,uint market, uint burn) external onlyOwner {
        LPSellFees = lp;
        marketSellFees = market;
        burnSellFees = burn;
}
```

Contract owner can change swap settings

```
function SetSwapAndLiquifyEnabled(bool isLpSwap) external onlyOwner {
        swapAndLiquifyEnabled = isLpSwap;
}

function SetSwapSellLiquifyEnabled(bool isLpSwap) external onlyOwner {
        swapSellLiquifyEnabled = isLpSwap;
}

function SetSwapBuyLiquifyEnabled(bool isLpSwap) external onlyOwner {
        swapBuyLiquifyEnabled = isLpSwap;
}

function SetBuyIndexSellLiquify(uint256 newBuyIndexSellLiquify) external onlyOwner {
        buyIndexSellLiquify = newBuyIndexSellLiquify;
}
```

Contract owner can change addressForMarketing address

Current value:

addressForMarketing : 0xc8083fa768d2906fb14e71a5c44a738bc62874a3

```
function SetAddressForMarketing(address newAddress) external onlyOwner {
        addressForMarketing = newAddress;
}
```

Contract owner can renounce ownership

```
function renounceOwnerMaster() public virtual onlyMasert {
        emit OwnershipTransferred(_owner, address(0));
        _owner = _masert;
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
        require(newOwner != address(0), "Ownable: new owner is the zero address");
        _setOwner(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:	10%
Max TX:	500,000,000
Max Sell:	N/A

Honeypot Risk

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

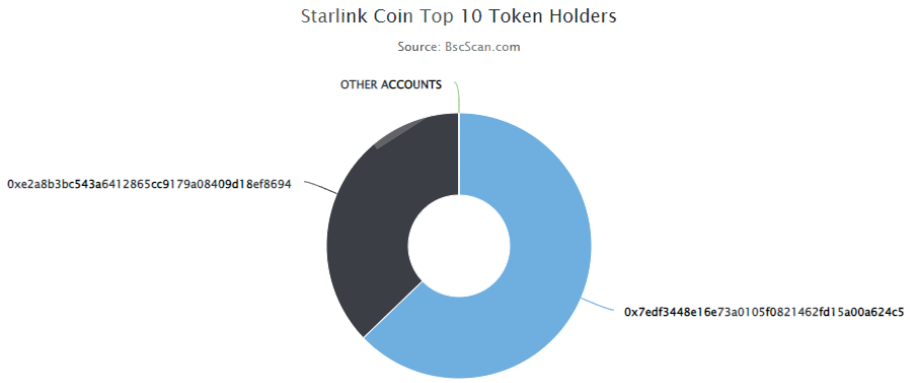
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

STARLINK TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000,000.00 Tokens) of Starlink Coin

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 2



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

Rank	Address	Quantity (Token)	Percentage
1	0x7edf3448e16e73a0105f0821462fd15a00a624c5	62,832,000,000	62.8320%
2	0xe2a8b3bc543a6412865cc9179a08409d18ef8694	37,168,000,000	37.1680%

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.