



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**SPEED**  
\$SPEED

24/06/2023



# TOKEN OVERVIEW

---

## Fees

- Buy fees: 0%
- Sell fees: 5%

## Fees privileges

- Can't change / set fees

## Ownership

- Owned

## Minting

- No mint function

## Max Tx Amount / Max Wallet Amount

- Can't change max tx amount and max wallet amount

## Blacklist

- Blacklist function not detected

## Other privileges

- Can exclude / include from fees
-

# TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-8

OWNER PRIVILEGES

9

CONCLUSION AND ANALYSIS

10

TOKEN DETAILS

11

SPEED TOKEN ANALYTICS &  
TOP 10 TOKEN HOLDERS

12

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

**0xF0c5a20eD7ebd785714C82AC7E6Ea3f5FE7bD31d**

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 **24/06/2023**



# WEBSITE DIAGNOSTIC

<http://www.speedbsc.com/>



0-49



50-89



90-100



Performance



Accessibility



Best  
Practices



SEO



Progressive  
Web App

## Socials



Twitter

<https://twitter.com/Speedbsc2023>



Telegram

<https://t.me/SpeedBSC>

# AUDIT OVERVIEW



Security Score



## Static Scan

Automatic scanning for common vulnerabilities



## ERC Scan

Automatic checks for ERC's conformance



High



Medium



Low



Optimizations



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 tax**

```
function excludeFromFees(address account, bool excluded) external onlyOwner {
    require(!_isExcludedFromFees[account] != excluded, "Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

- **Contract owner can exclude/include wallet from rewards**

```
function excludeFromReward(address account) public onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already included");
    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;
        }
    }
}
```

- **Contract owner can change swap settings**

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner() {
    require(newAmount > totalSupply() / 1e5, "SwapTokensAtAmount must be greater than 0.001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(newAmount);
}

function setSwapEnabled(bool _enabled) external onlyOwner {
    swapEnabled = _enabled;
    emit SwapEnabledUpdated(_enabled);
}
```

## ● Contract owner can change **marketingWallet** address

Current value:

**marketingWallet:** 0x0b2d2602f52e1ade05e4b6117c94daea23115b66

```
function changeMarketingWallet(address _marketingWallet) external onlyOwner {
    require(_marketingWallet != marketingWallet, "Marketing wallet is already that address");
    require(_marketingWallet!=address(0), "Marketing wallet is the zero address");
    marketingWallet = _marketingWallet;
    emit MarketingWalletChanged(marketingWallet);
}
```

## ● Contract owner has ability to retrieve any token held by the contract (native tokens excluded)

```
function claimStuckTokens(address token) external onlyOwner {
    require(token != address(this), "Owner cannot claim native tokens");
    if (token == address(0x0)) {
        payable(msg.sender).sendValue(address(this).balance);
        return;
    }
    IERC20 ERC20token = IERC20(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

## ● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

## ● Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
}
```

### **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 no HIGH issues during the first review.

# TOKEN DETAILS

## Details

Buy fees: 0%

Sell fees: 5%

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: 89% tokens unlocked



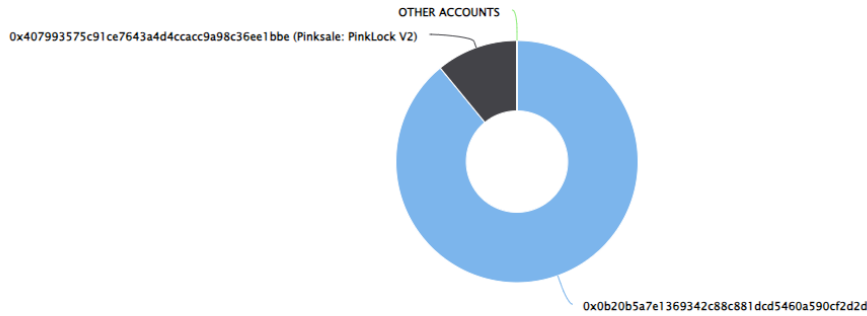
# SPEED TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (420,690,000.00 Tokens) of SPEED

Token Total Supply: 420,690,000.00 Token | Total Token Holders: 2

## SPEED Top 10 Token Holders

Source: BscScan.com



(A total of 420,690,000.00 tokens held by the top 10 accounts from the total supply of 420,690,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	<a href="#">0x0b20b5a7e1369342c88c881dcd5460a590cf2d2d</a>	374,708,583	89.0700%
2	<a href="#">Pinksale: PinkLock V2</a>	45,981,417	10.9300%

# 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.

