



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



06/04/2022



TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 RAPIDEFI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 10 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 RapiDefi (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x9fdcc07A32A9e2E6EF0D31933331F98dFb47943

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 06/04/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



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 tax

```
function setIsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

Contract owner can exclude/include address from wallet limitations

```
function setIsMaxWalletExempt(address holder, bool exempt)
    external
    onlyOwner
{
    isMaxWalletExempt[holder] = exempt;
}
```

Contract owner can change wallet limitations (with threshold)

```
function setMaxWalletPercent(uint256 percent) external onlyOwner {
    //prevent hp
    require(percent > 1, "Max wallet can only be more than 1%");
    _maxWalletToken = _totalSupply.mul(percent).div(100);
}
```

Contract owner can change swap settings

```
function setSwapBackSettings(
    bool _enabled,
    uint256 _amount,
    uint256 _taxBurnAmount
) external onlyOwner {
    require(
        _amount > _taxBurnAmount,
        "Swap threshold must be more than amount burned"
    );
    swapEnabled = _enabled;
    swapThreshold = _amount * 10**9;
    taxBurnAmount = _taxBurnAmount * 10**9;
}
```

Contract owner can enable trading (current value `tradingOpen = false`)

```
function tradingStatus() external onlyOwner {
    require(!tradingOpen, "Trading is already open.");
    tradingOpen = true;
}
```

Contract owner can enable/disable tax

```
function setFeesOn(bool _feesOn) external onlyOwner {  
    feesOn = _feesOn;  
}
```

Stake function allows investors to stake RDFi tokens without being transferred from the wallet (staking contract N/A)

```
function stake(address holder, uint256 amount) external authorized {  
    require(  
        _balances[holder] >= amount,  
        "You can't stake more than you own."  
    );  
    staked[holder] = amount;  
}  
  
. . .  
  
//if staking tokens, they can't be transferred  
if (staked[sender] > 0) {  
    uint256 heldTokens = balanceOf(sender);  
    require(  
        amount <= heldTokens.sub(staked[sender]),  
        "Can't send actively staked tokens, unstake first"  
    );  
}
```

Contract owner can withdraw foreign tokens

```
function emergencyWithdraw() external onlyOwner {  
    (bool success, ) = payable(msg.sender).call{  
        value: address(this).balance  
    }("");  
    require(success, "Withdraw failed");  
}  
  
function emergencyWithdrawTokens(address token) external onlyOwner {  
    IBEP20 tokenIB20 = IBEP20(token);  
    tokenIB20.transfer(msg.sender, tokenIB20.balanceOf(address(this)));  
}
```

Contract owner can transfer ownership

```
function transferOwnership(address payable adr) public onlyOwner {  
    owner = adr;  
    authorizations[adr] = true;  
    emit OwnershipTransferred(adr);  
}
```

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:	3%
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:	Clean



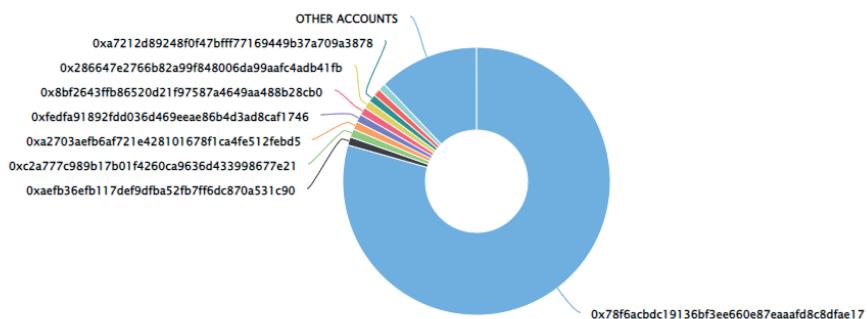
RAPIDEFI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 87.90% (87,901,435.63 Tokens) of RapiDefi

Token Total Supply: 100,000,000.00 Token | Total Token Holders: 140

RapiDefi Top 10 Token Holders

Source: BscScan.com



(A total of 87,901,435.63 tokens held by the top 10 accounts from the total supply of 100,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x78f6acbd19136bf3ee660e87eaaaf8c8dfae17	79,368,434.598636634	79.3684%
2	0xaeefb36efb117def9dfba52fb7ff6dc870a531c90	999,999.98	1.0000%
3	0xc2a777c989b17b01f4260ca9636d433998677e21	999,999.9267	1.0000%
4	0xa2703aefb6af721e428101678f1ca4fe512febd5	975,567.5298	0.9756%
5	0xfedfa91892fd036d469eeae86b4d3ad8caf1746	962,878.6159	0.9629%
6	0x8bf2643fb86520d21f97587a4649aa488b28cb0	954,838.3401	0.9548%
7	0x286647e2766b82a99f848006da99aafc4adb41fb	941,251.1471	0.9413%
8	0xa7212d89248f0f47bfff77169449b37a709a3878	932,659.6129	0.9327%
9	0x977cb24024b0e1a9e9539b457db640ab25e772de	887,902.074	0.8879%
10	0x9cefad4a8c0d1d99308e0728bfc629a64e8ff641	877,903.808	0.8779%

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.

