



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Retro INU
\$RINU

30/07/2022

TABLE OF CONTENTS

- 1 **DISCLAIMER**
- 2 **INTRODUCTION**
- 3-4 **AUDIT OVERVIEW**
- 5 **OWNER PRIVILEGES**
- 6 **CONCLUSION AND ANALYSIS**
- 7 **TOKEN DETAILS**
- 8 **RETRO INU TOKEN ANALYTICS &
TOP 10 TOKEN HOLDERS**
- 9 **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 Honeygot etc)



INTRODUCTION

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

0x06784F9dfB9D70389713eE8af40A4386e6157EE0

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 **30/07/2022**



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 change swap status

```
function setSwapAndLiquifyEnabled(bool _enabled) external onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

- Contract owner can exclude/include wallet from tax

```
function setIsExcludedFromFee(address _address, bool _isExcluded)
    external
    onlyOwner
{
    _isExcludedFromFee[_address] = _isExcluded;
}
```

- Contract owner can change `_marketingWallet` address
- Current value:

`_marketingWallet` : `0x2eb2e0f7beb2e43e6098aee117932063bf384495`

```
function setMarketingWallet(address payable _address) external onlyOwner {
    _marketingWallet = _address;
}
```

- Contract owner can renounce ownership

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

- 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;
}
```

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 0 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	6%
Sell fees:	18%
Max TX:	1,000,000,000,000
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



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.

