



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Lunar Rabbit

\$LunarRabbit

11/01/2023

TOKEN OVERVIEW

Fees

- Buy fees: 6%
- Sell fees: 6%
- Transfer fees: 10%

Fees privileges

- Can change buy fees up to 10%, sell fees up to 10% and transfer fees up to 11%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount and max wallet amount (with threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees
-

TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3-4

AUDIT OVERVIEW

5-8

OWNER PRIVILEGES

9

CONCLUSION AND ANALYSIS

10

TOKEN DETAILS

11

**LUNARRABBIT 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 **Lunar Rabbit** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xCF136913c4583aD8D12190DeA731e6FA75F45E95

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 **11/01/2023**



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) public onlyOwner {
    _isExcludedFromFees[account] = excluded;
    emit ExcludeFromFees(account, excluded);
}
```

- Contract owner can exclude/include wallet from tx limitations

```
function excludeFromMaxTransaction(address updAds, bool isEx) public onlyOwner {
    _isExcludedMaxTransactionAmount[updAds] = isEx;
}
```

- Contract owner can send tokens in multiple wallets

```
function airdropToWallets(address[] memory airdropWallets, uint256[] memory amounts) external onlyOwner
returns (bool){
    require(!tradingActive, "Trading is already active, cannot airdrop after launch.");
    require(airdropWallets.length == amounts.length, "arrays must be the same length");
    require(airdropWallets.length < 200, "Can only airdrop 200 wallets per txn due to gas limits"); // allows for
    airdrop + launch at the same exact time, reducing delays and reducing sniper input.
    for(uint256 i = 0; i < airdropWallets.length; i++){
        address wallet = airdropWallets[i];
        uint256 amount = amounts[i];
        _transfer(msg.sender, wallet, amount);
    }
    return true;
}
```

- Contract owner has to call `enableTrading()` function to enable trade

```
function enableTrading() external onlyOwner {
    tradingActive = true;
    swapEnabled = true;
}
```

- Contract owner can reset all taxes to 0

```
function resetTaxAmount() public onlyOwner {
    tokensForLiquidity = 0;
    tokensForMarketing = 0;
    tokensForStaking = 0;
    tokensForTeam = 0;
}
```


● Contract owner can enable/disable staking

```
function setStakingEnabled(bool enabled) external onlyOwner(){
    stakingEnabled = enabled;
}
```

● Contract owner can change buy fees up to 10%, sell fees up to 10% and transfer fees up to 11%

```
function updateBuyFees(uint256 _marketingFee, uint256 _liquidityFee, uint256 _devFee, uint256 _teamFee)
external onlyOwner {
    buyMarketingFee = _marketingFee;
    buyLiquidityFee = _liquidityFee;
    buyStakingFee = _devFee;
    buyTeamFee = _teamFee;
    buyTotalFees = buyMarketingFee + buyLiquidityFee + buyStakingFee + buyTeamFee;
    require(buyTotalFees <= 10, "Must keep fees at 10% or less");
}
```

```
function updateSellFees(uint256 _marketingFee, uint256 _liquidityFee, uint256 _devFee, uint256 _teamFee)
external onlyOwner {
    sellMarketingFee = _marketingFee;
    sellLiquidityFee = _liquidityFee;
    sellStakingFee = _devFee;
    sellTeamFee = _teamFee;
    sellTotalFees = sellMarketingFee + sellLiquidityFee + sellStakingFee + sellTeamFee;
    require(sellTotalFees <= 10, "Must keep fees at 10% or less");
}
```

```
function updateTransferFees(uint256 _marketingFee, uint256 _liquidityFee, uint256 _devFee, uint256 _team-
Fee) external onlyOwner {
    transferMarketingFee = _marketingFee;
    transferLiquidityFee = _liquidityFee;
    transferStakingFee = _devFee;
    transferTeamFee = _teamFee;
    transferTotalFees = transferMarketingFee + transferLiquidityFee + transferStakingFee + transferTeamFee;
    require(transferTotalFees <= 11, "Must keep fees at 11% or less");
}
```

● Contract owner can change marketingWallet, stakingWallet and teamWallet addresses

Default values:

marketingWallet : 0x6bd7f652BB574c6f3336b45448c1b4b2d09CF777

stakingWallet : owner wallet

teamWallet : 0x2ea8EE42dA22f5208905B55F15C66EdA3BFDcfA6

```
function updateMarketingWallet(address newMarketingWallet) external onlyOwner {
    emit marketingWalletUpdated(newMarketingWallet, marketingWallet);
    marketingWallet = newMarketingWallet;
}
```

```
function updateStakingWallet(address newWallet) external onlyOwner {
    emit stakingWalletUpdated(newWallet, stakingWallet);
    stakingWallet = newWallet;
}
```

```
function updateTeamWallet(address newWallet) external onlyOwner {
    emit teamWalletUpdated(newWallet, teamWallet);
    teamWallet = newWallet;
}
```

● Contract owner can change max wallet amount (with threshold)

```
function updateMaxWallet(uint256 newNum) external onlyOwner {
    require(newNum >= (totalSupply() * 1 / 100) / (10**_decimals), "Cannot set maxTransactionAmount lower than 1%");
    maxWallet = newNum * (10**_decimals);
}
```

● Contract owner can change max tx amount (with threshold)

```
function updateMaxAmount(uint256 newNum) external onlyOwner {
    require(newNum >= (totalSupply() * 1 / 100) / (10**_decimals), "Cannot set maxTransactionAmount lower than 1%");
    maxTransactionAmount = newNum * (10**_decimals);
}
```

● Contract owner can change swap settings

```
function updateSwapEnabled(bool enabled) external onlyOwner() {
    swapEnabled = enabled;
}
```

● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public 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:	6%
Sell fees:	6%
Transfer fees:	10%
Max TX:	100,000,000
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



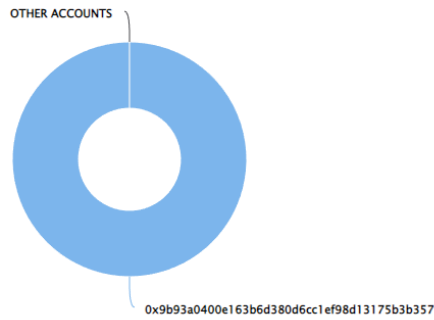
LUNARRABBIT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000.00 Tokens) of Lunar Rabbit

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 1

Lunar Rabbit Top 10 Token Holders

Source: BscScan.com



(A total of 1,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x9b93a0400e163b6d380d6cc1ef98d13175b3b357	1,000,000,000	100.0000%

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.

