freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

LONKAR GLOBAL LTD

$LGL

24/06/2025

TOKEN OVERVIEW

Fees

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

Fees privileges

- Can change fees up to 20%

Ownership

- Owned

Minting

- Mint function detected (not more than the total supply)

Max Tx Amount / Max Wallet Amount

- Can’t change max tx amount and / or max wallet amount

Blacklist

- Blacklist function not detected

Other privileges

- N/A

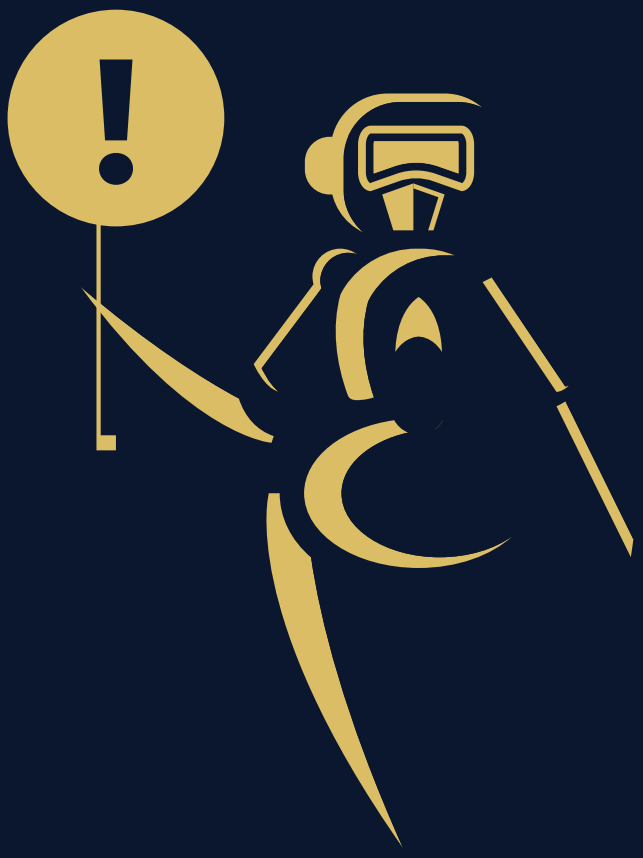
TABLE OF CONTENTS

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
LONKAR GLOBAL LTD (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xb7f4EE705Ae6F7E92579550D419fAA722B8FDD4C

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 24/06/2025

WEBSITE DIAGNOSTIC

https://www.lonkargloballtd.uk

0-49 50-89 90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
92	91	86	88	NA

Socials

X (Twitter)

N/A

Telegram

https://t.me/LGLCoinOfficial

AUDIT OVERVIEW

91

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

92

ERC Scan
Automatic checks for ERC's conformance

0 High

0 Medium

1 Low

1 Optimizations

0 Informational

No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Low
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 mint tokens after initial contract deploy

The owner can mint new tokens if minting is enabled through mint, but not more than the total supply

```
function mint(address to, uint256 amount) external onlyOwner {
  if (!isMintable()) {
   revert MintingNotEnabled();
  }
  if (isMaxAmountOfTokensSet() && !isFeesAndLimitsExcluded[to]) {
   if (balanceOf(to) + amount > maxTokenAmountPerAddress) {
    revert DestBalanceExceedsMaxAllowed(to);
   }
  }
  if (isMaxSupplySet()) {
   if (totalSupply() + amount > maxTotalSupply) {
    revert TotalSupplyExceedsMaxAllowedAmount();
   }
  }

  super._mint(to, amount);
}
```

Validation Order in Tax Configuration

In the setTaxConfig function, input validation for the _taxAddress should precede the isTaxable() flag check to avoid unnecessary gas usage if input is invalid.

Recommendation:

Reorder the input validation logic in the setTaxConfig function so that the _taxAddress is validated before checking the isTaxable() flag. This reduces unnecessary gas consumption by ensuring that invalid inputs are rejected early, avoiding further execution and state reads when they are not needed.

```
function setTaxConfig(
  address _taxAddress,
  uint256 _taxBPS
 ) external onlyOwner {
  if (!isTaxable()) {
   revert TokenIsNotTaxable();
  }

  uint256 totalBPS = deflationBPS + tFeeBPS + _taxBPS;
  if (totalBPS > MAX_ALLOWED_BPS) {
   revert InvalidTotalBPS(totalBPS);
  }
  LibCommon.validateAddress(_taxAddress);
  taxAddress = _taxAddress;
  taxBPS = _taxBPS;
  emit TaxConfigSet(_taxAddress, _taxBPS);
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(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");
    _transferOwnership(newOwner);
}

function _transferOwnership(address newOwner) internal virtual {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
}
```

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:	0%
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:	100% unlocked tokens

LGL TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00%
(100,010,000,000.00 Tokens) of LONKAR GLOBAL COIN

Token Total Supply: 100,010,000,000.00 Token | Total Token Holders: 7

(A total of 100,010,000,000.00 tokens held by the top 10 accounts from the total supply of 100,010,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x47910225...c0a963eB1	99,935,435,269.425	99.9254%
2	0xFeA9c74B...C271BAccD	62,500,000	0.0625%
3	0xb7f4EE70...22B8FDD4C	10,000,000	0.0100%
4	0x63170C39...3C04A4263	1,000,000	0.0010%
5	0x424eDcF2...6BBA4C01f	999,900	0.0010%
6	PancakeSwap V2: LGL 16	64,482.357322	0.0001%
7	0x5D750D03...88b51bF46	348.217678	0.0000%

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.