freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

FORG LORD 2049

$FROGLORD2049

25/09/2023

TOKEN OVERVIEW

Fees

- Buy fees: 2%
- Sell fees: 2%

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

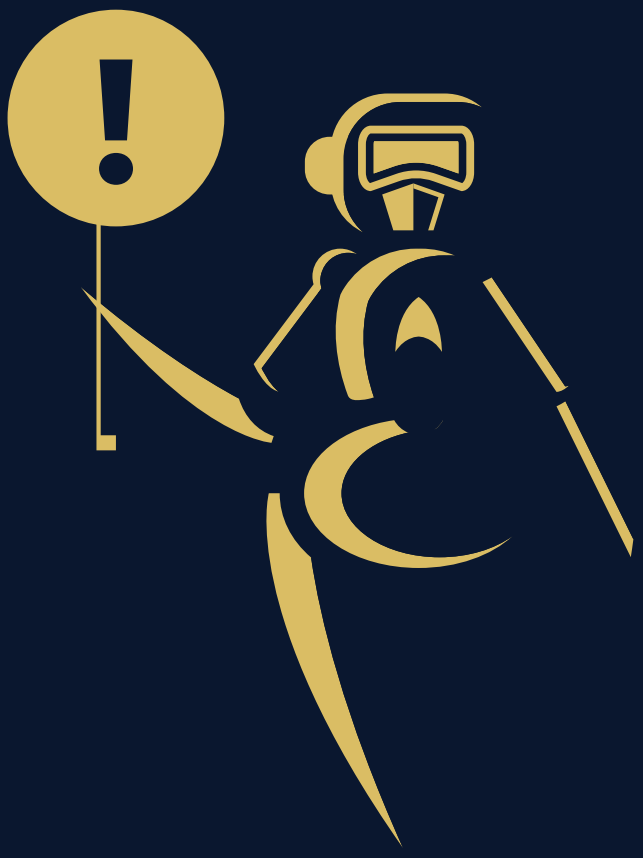
TABLE OF CONTENTS

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
FORG LORD 2049 (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xb39F40dae73Bc731485DcA1D6E484795173327 58

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 25/09/2023

WEBSITE DIAGNOSTIC

N/A


0-49


50-89


90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
NA	NA	NA	NA	NA

Socials


Twitter

https://twitter.com/froglord2049


Telegram

https://t.me/froglord2049chat

AUDIT OVERVIEW

94

Security Score

96

Static Scan
Automatic scanning for common vulnerabilities

92

ERC Scan
Automatic checks for ERC's conformance

0 High

0 Medium

0 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	Low
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;
}
```

- Contract owner can change swap settings

```
function setAutoSwap(bool state_) external onlyOwner {
    autoSwapEnable = state_;
}

function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner {
    require(
      newAmount > totalSupply() / 100000,
      "SwapTokensAtAmount must be greater than 0.001% of total supply"
    );
    swapTokensAtAmount = newAmount;
}
```

- Contract owner can change wallet01 address

Default value:

wallet01 : 0x438Ec64016D3d28c2842E90547e68754D6471a14

```
function changeWallet(address _wallet01) external onlyOwner {
    wallet01 = address(_wallet01);
}
```

- Missing Zero Address Check

- FROGLORD2049::changeWallet()

Contract owner has ability to retrieve any token held by the contract

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

function sendBNB(address payable recipient, uint256 amount) internal returns(bool) {
    require(
        address(this).balance >= amount,
        "Address: insufficient balance"
    );

    (bool success, ) = recipient.call{value: amount}("");
    return success;
}
```

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:	2%
Sell fees:	2%
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

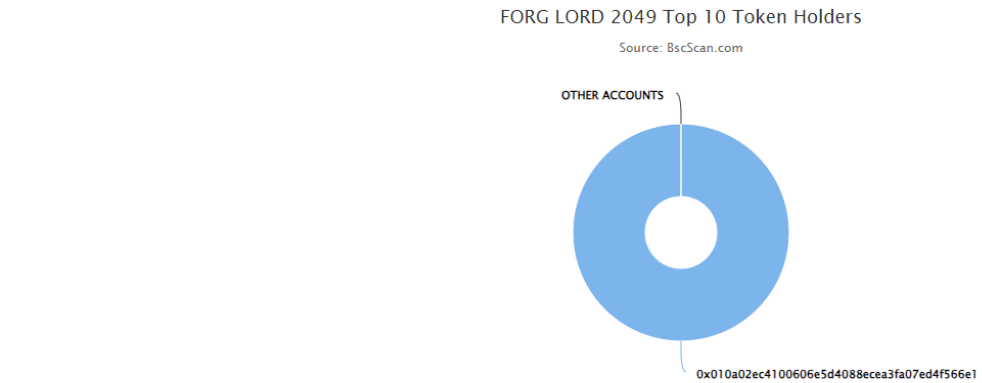
Rug Pull Risk

Liquidity:	N/A
Holders:	100% unlocked tokens

FROGLORD2049 TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (420,000,000,000.00 Tokens) of FORG LORD 2049

Token Total Supply: 420,000,000,000.00 Token | Total Token Holders: 1



(A total of 420,000,000,000.00 tokens held by the top 10 accounts from the total supply of 420,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x010A02...d4F566E1	420,000,000,000	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.