



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



ALF.FINANCE
\$ALF



10/01/2022



TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 WEBSITE DIAGNOSTIC
- 5-6 AUDIT OVERVIEW
- 7-8 OWNER PRIVILEGES
- 9 CONCLUSION AND ANALYSIS
- 10 TOKEN DETAILS
- 11 ALF.FINANCE TOKEN DISTRIBUTION & 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 ALF.FINANCE (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xf666108FDaa6204B35fD90D6d459e77D3B36139b

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 10/01/2022



WEBSITE DIAGNOSTIC

<https://alf.finance/>



0-49



50-89



90-100



Performance



Accessability



Best Practices



SEO



Progressive
Web App

Metrics



First Contentful Paint

2.1 s



Time to interactive

2.6 s



Speed Index

2.7 s



Total Blocking Time

0 ms



Large Contentful Paint

5.1 s



Cumulative Layout Shift

0

Issues found

Avoid enormous network payloads - Total size was 9,073 KiB

Does not have a `<meta name="viewport">` tag with width or initial-scale

Background and foreground colors do not have a sufficient contrast ratio

Links do not have a discernible name

Document does not have a meta description

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 exclude/include wallet from fee

```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

Contract owner can exclude/include wallet from reward

```
function excludeFromReward(address account) public onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    if (_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tOwned[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

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;
}
```

Contract owner can change the fees

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}

function setBurnFeePercent(uint256 burnFee) external onlyOwner() {
    _burnFee = burnFee;
}

function setMarketingAndDevFeePercent(uint256 mFee) external onlyOwner() {
    marketingAndDevelopment = mFee;
}
```

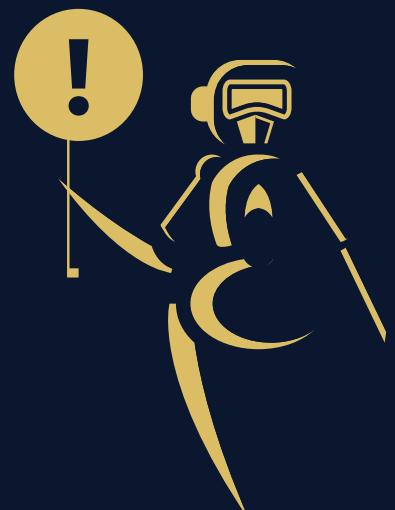
Contract owner can change swap settings

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

Contract owner can change max tx percent & max wallet amount

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}

function setMaxWalletAmount(uint256 setMaxWallet) public onlyOwner {
    _maxWalletToken = setMaxWallet;
}
```



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: 0%

Sell fees: 0%

Max TX: N/A

Max Sell: N/A

Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Detected

Modify Max Sell: Not detected

Disable Trading: Not detected

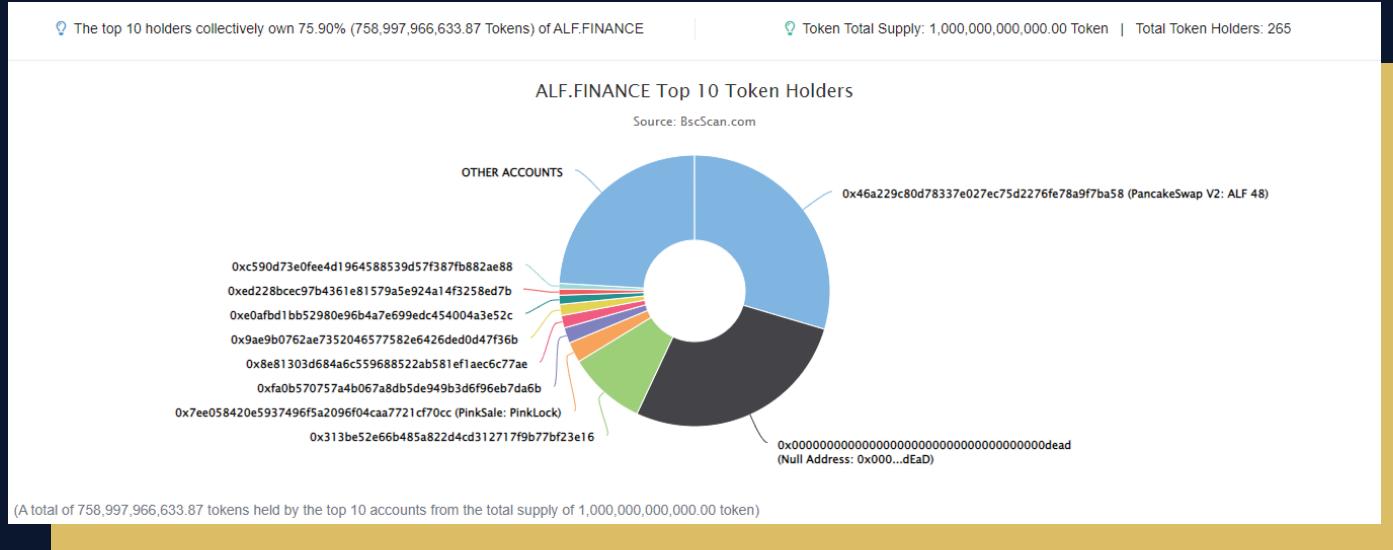
Rug Pull Risk

Liquidity: 2.45% Locked

Holders: Clean



ALF.FINANCE TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: ALF 48	295,708,116,684.120356334893288577	29.5708%
2	Null Address: 0x000...dEaD	274,132,707,562.767638137317739772	27.4133%
3	0x313be52e66b485a822d4cd312717f9b77bf23e16	93,033,312,415.36333542500000004	9.3033%
4	PinkSale: PinkLock	24,500,000,000	2.4500%
5	0xfa0b570757a4b067a8db5de949b3d6f96eb7da6b	18,665,314,009.303789593309894092	1.8665%
6	0x8e81303d684a6c559688522ab581ef1aec6c77ae	14,800,462,611.056721368074449205	1.4800%
7	0x9ae9b0762ae7352046577582e6426ded0d47f36b	13,358,199,402.044332017158187896	1.3358%
8	0xe0afbd1bb52980e96b4a7e699edc454004a3e52c	10,810,651,722.606078627622825918	1.0811%
9	0xed228bcec97b4361e81579a5e924a14f3258ed7b	7,210,021,013.728468418394950306	0.7210%
10	0xc590d73e0fee4d1964588539d57f387fb882ae88	6,779,181,212.883109269843478131	0.6779%

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.

