



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Akeelah
\$AKEELAH



01/02/2022



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 AKEELAH TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS
- 12 TECHNICAL DISCLAIMER



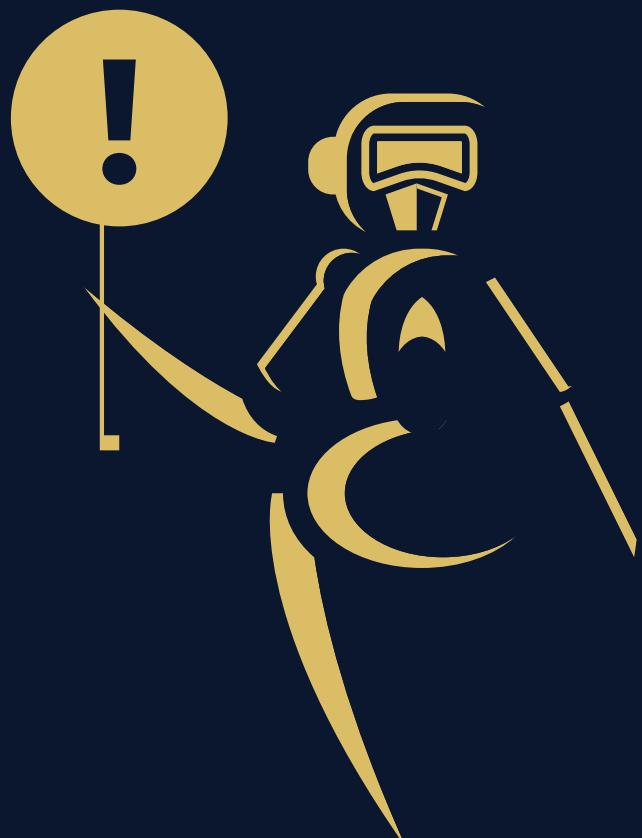
DISCLAIMER

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 Akeelah (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x5229c8d1329bAD22aE0fb5139192868540932e39

Network: Ethereum (ERC)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 01/02/2022



WEBSITE DIAGNOSTIC

<https://akeelah.io/>



0-49



50-89



90-100



Performance



Accessibility



Best Practices



SEO



Progressive
Web App

Metrics



First Contentful Paint

3.7 s



Time to interactive

80.3 s



Speed Index

10.9 s



Total Blocking Time

1,100 ms



Large Contentful Paint

17.6 s



Cumulative Layout Shift

0

WEBSITE IMPROVEMENTS

Reduce JavaScript execution time **85.2 s**

Minimize main-thread work **92.5 s**

Ensure text remains visible during webfont load

Avoid enormous network payloads Total size was 5,946 KiB

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't mint tokens after initial contract deploy.

Contract owner can exclude/include wallet from reward

```
function excludeFromReward(address account) public onlyOwner() {
    require(account != 0x10ED43C718714eb63d5aA57B78B54704E256024E, "We can not exclude Pancake router.");
    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 swap settings

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

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 change max tx amount

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    require(maxTxPercent > 10, "Cannot set transaction amount less than 10 percent!");
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}
```

Contract owner can change treasuryWallet and marketingWallet address

```
address public treasuryWallet = 0x81e5f9Dc80C1AefD0Bc008d18e03943433C4F7a5;
address public marketingWallet = 0x88090868f76E2f22116e0c1652185896CD04f25f;
```

```
function setmarketingWallet(address newWallet) external onlyOwner() {
    marketingWallet = newWallet;
}

function setTreasuryWallet(address newWallet) external onlyOwner(){
    treasuryWallet = newWallet;
}
```

Contract owner can change fees up to 100%

```
function setFee(uint256 reflectionFee, uint256 liquidityFee, uint256 treasuryFee, uint256 marketingFee)
external onlyOwner(){
    _taxFee = reflectionFee;
    _liquidityFee = liquidityFee;
    _treasuryFee = treasuryFee;
    _marketingFee = marketingFee;
}
```

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 issue during the first review.

TOKEN DETAILS

Details

Buy fees: 15%

Sell fees: 17%

Max TX: N/A

Max Sell: N/A

Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Detected with threshold

Modify Max Sell: Not detected

Disable Trading: Not detected

Rug Pull Risk

Liquidity: N/A

Holders: Clean



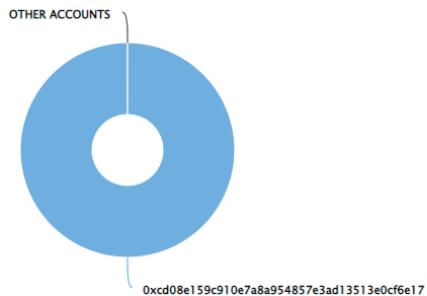
AKEELAH TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

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

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

Akeelah Top 10 Token Holders

Source: Etherscan.io



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

Rank	Address	Quantity (Token)	Percentage
1	0xcd08e159c910e7a8a954857e3ad13513e0cf6e17	1,000,000,000,000,000	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.

