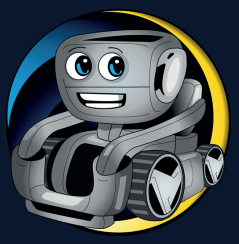
freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



WAL-E Coin

$WAL-E

04/03/2022

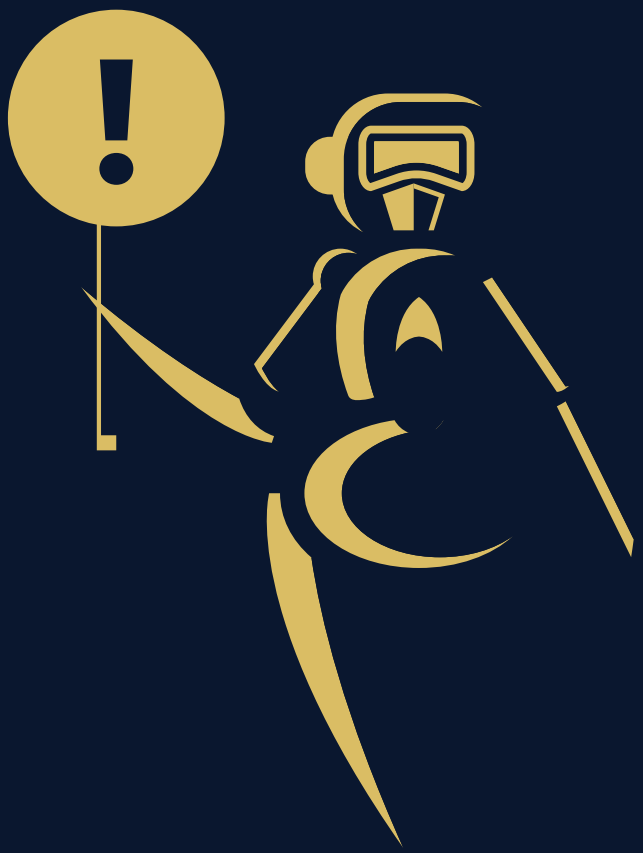
TABLE OF CONTENTS

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
WAL-E Coin (Customer) to conduct a Smart Contract Code Review
and Security Analysis.

0x67CeA5e25903c3022EBaf99e67e1898F1De6a75E

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 04/03/2022

WEBSITE DIAGNOSTIC

https://walerewards.com/

0-49 50-89 90-100

82	92	93	98	NA
Performance	Accessibility	Best Practices	SEO	Progressive Web App

Metrics

First Contentful Paint
3.4 s

Time to interactive
10.2 s

Speed Index
5.6 s

Total Blocking Time
610 ms

Large Contentful Paint
8.3 s

Cumulative Layout Shift
0

WEBSITE IMPROVEMENTS

Reduce unused CSS

Reduce the impact of third-party code Third-party code blocked the main thread for 410 ms

Ensure text remains visible during webfont load

Reduce JavaScript execution time 1.6 s

Avoid enormous network payloads Total size was 3,128 KiB

Background and foreground colors do not have a sufficient contrast ratio

Heading elements are not in a sequentially-descending order

Image elements do not have explicit width and height

AUDIT OVERVIEW

98

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

98

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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet from tax

```
function setFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

Contract owner can exclude/include wallet from tx limitations

```
function setTxLimitExempt(address holder, bool exempt) external authorized {
    isTxLimitExempt[holder] = exempt;
}
```

Contract owner can change buyback settings

```
function setAutoBuybackSettings(bool _enabled, uint256 _cap, uint256 _amount, uint256 _period) external
authorized {
    autoBuybackEnabled = _enabled;
    autoBuybackCap = _cap;
    autoBuybackAccumulator = 0;
    autoBuybackAmount = _amount;
    autoBuybackBlockPeriod = _period;
    autoBuybackBlockLast = block.number;
}

function setBuybackMultiplierSettings(uint256 numerator, uint256 denominator, uint256 length) external
authorized {
    require(numerator / denominator <= 2 && numerator > denominator);
    buybackMultiplierNumerator = numerator;
    buybackMultiplierDenominator = denominator;
    buybackMultiplierLength = length;
}
```

Contract owner can change max tx amount

```
function setTxLimit(uint256 amount) external authorized {
    require(amount >= _totalSupply / 1000);
    _maxTxAmount = amount;
}
```

Contract owner can change the fees up to 25%

```
function setFees(uint256 _liquidityFee, uint256 _buybackFee, uint256 _reflectionFee, uint256 _marketingFee,
uint256 _feeDenominator) external authorized {
    liquidityFee = _liquidityFee;
    buybackFee = _buybackFee;
    reflectionFee = _reflectionFee;
    marketingFee = _marketingFee;
    totalFee = _liquidityFee.add(_buybackFee).add(_reflectionFee).add(_marketingFee);
    feeDenominator = _feeDenominator;
    require(totalFee < feeDenominator/4);
}
```

Contract owner can change autoLiquidityReceiver and marketingFeeReceiver address

Current values:

autoLiquidityReceiver : 0xdec2493410f08d95bebea87a9ead8e5ec3e102c8

marketingFeeReceiver : 0x8e1bc3e5c5a6af6a3991441c62170edc8b2a2040

```
function setFeeReceivers(address _autoLiquidityReceiver, address _marketingFeeReceiver) external authorized
{
    autoLiquidityReceiver = _autoLiquidityReceiver;
    marketingFeeReceiver = _marketingFeeReceiver;
}
```

Contract owner can change swap settings

```
function setSwapBackSettings(bool _enabled, uint256 _amount) external authorized {
    swapEnabled = _enabled;
    swapThreshold = _amount;
}
```

Contract owner can transfer ownership

```
function transferOwnership(address payable adr) public onlyOwner {
    owner = adr;
    authorizations[adr] = true;
    emit OwnershipTransferred(adr);
}
```

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:	15%
Max TX:	1,250,000,000,000
Max Sell:	N/A

Honeypot Risk

Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

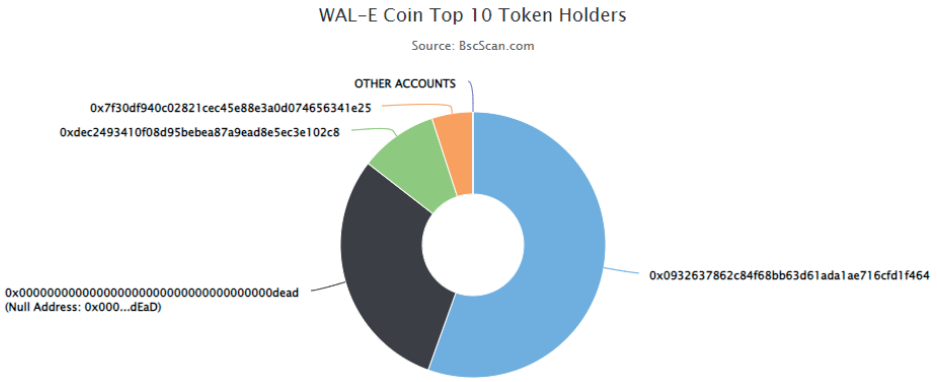
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

WAL-E TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000,000,000.00 Tokens) of WAL-E Coin

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 4



(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	0x0932637862c84f68bb63d61ada1ae716cfd1f464	555,000,000,000,000	55.5000%
2	Null Address: 0x000...dEaD	300,000,000,000,000	30.0000%
3	0xdec2493410f08d95bebea87a9ead8e5ec3e102c8	95,000,000,000,000	9.5000%
4	0x7f30df940c02821cec45e88e3a0d074656341e25	50,000,000,000,000	5.0000%

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.