freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

GreenAir
$GREEN

24/12/2021

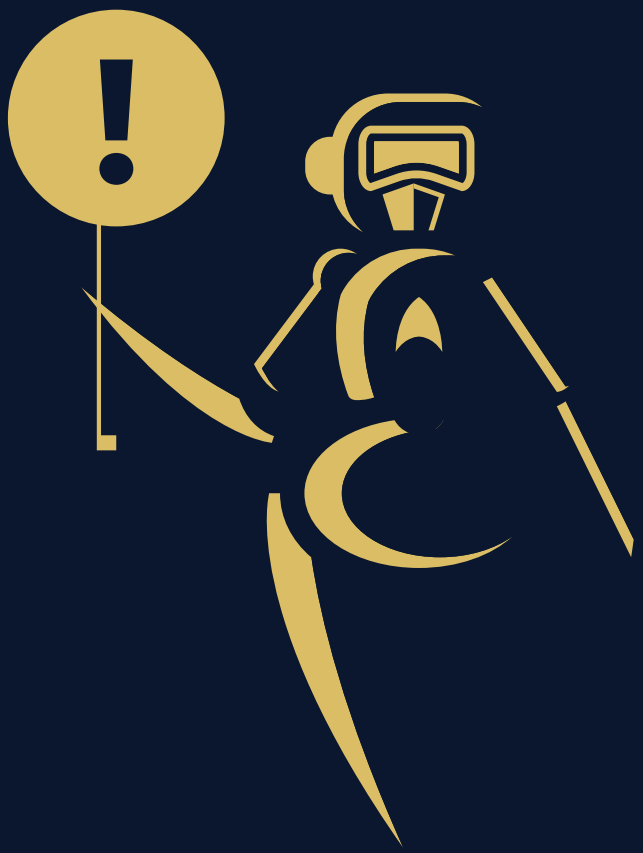
TABLE OF CONTENTS

DISCLAIMER

The information provided on this analysis decoument 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 GreenAir (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xafcd56e0d0ad1a769db98f14d4149a78f52ce620

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 24/12/2021

WEBSITE DIAGNOSTIC

https://thegreenair.com/

0-49 50-89 90-100

55	81	92	93	NA
Performance	Accessability	Best Practices	SEO	Progressive Web App

Metrics

First Contentful Paint
9.1 s

Time to interactive
23.9 s

Speed Index
20.5 s

Total Blocking Time
1,450 ms

Large Contentful Paint
17.6 s

Cumulative Layout Shift
0

Issues found

Eliminate render-blocking resources

Reduce unused JavaScript

Reduce unused CSS

Defer offscreen images

Ensure text remains visible during webfont load

Reduce the impact of third-party code Third-party code blocked the main thread for 1,600 ms

Background and foreground colors do not have a sufficient contrast ratio.

Heading elements are not in a sequentially-descending order

AUDIT OVERVIEW

92

Security Score

94

Static Scan
Automatic scanning for common vulnerabilities

99

ERC Scan
Automatic checks for ERC's conformance

0 High

0 Medium

1 Low

0 Optimizations

0 Informational

No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passsed
2	Reentrancy and Cross-function	Passsed
3	Front running	Passsed
4	Timestamp dependence	Passsed
5	Integer Overflow and Underflow	Passsed
6	Reverted DoS	Passsed
7	DoS with block gas limit	Low
8	Methods execution permissions	Passsed
9	Exchange rate impact	Passsed
10	Malicious Event	Passsed
11	Scoping and Declarations	Passsed
12	Uninitialized storage pointers	Passsed
13	Design Logic	Passsed
14	Safe Zeppelin module	Passsed

OWNER PRIVILEGES

Contract owner can add any wallet address as smart contract admin.

```
function addContractSaler(address _adminUser, bool isAllowed)
public returns (bool ) {
require(contractAdmin[msg.sender], "Only Admin can act
with this");
contractAdmin[_adminUser] = isAllowed;
return true;
}
```

Contract owner can add any wallet address to transfer tokens after presale closing and before trade is open

```
function addAllowedWallet(address _wallet, bool isAllowed)
public returns (bool ) {
require(contractAdmin[msg.sender], "Only Admin can act with this");
allowedWallet[_wallet] = isAllowed;
return isAllowed;
}
```

Contract owner can set pivateSaleIsOpen value True or False by calling setPrivateSaleStatus function

```
function setPrivateSaleStatus( bool isOpen) public returns (bool
) {
require(contractAdmin[msg.sender], "Only Admin can act
with this");
pivateSaleIsOpen = isOpen;
return isOpen;
}
```

Contract owner can set token price in private sale by calling setTokenPrice function

```
function setTokenPrice(uint256 priceWeiTokens) public returns
(bool ) {
require(msg.sender == owner,"Only Admin can set the
price");
tokenPrice = priceWeiTokens;
return true;
}
```

```
require(amount >= tokenPrice * (tokens / (10 **
uint256(_decimals))), "Wrong Token price!");
```

Contract owner can open trade by calling openTrade function, without this action, trade remain closed (False value)

```
function openTrade() public returns (bool ) {
require(msg.sender == owner,"Only Admin can open Trade");
tradeIsOpen = true;
return true;
}
```

CONCLUSION AND ANALYSIS

Smart Contracts within the scope were manually reviewd and analyzed with static tools.

Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.

Found 1 low issue during the first review.

Trade is disabled for the safety of the project, contract owner have to call openTrade function and setPrivateSaleStatus with the required values

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

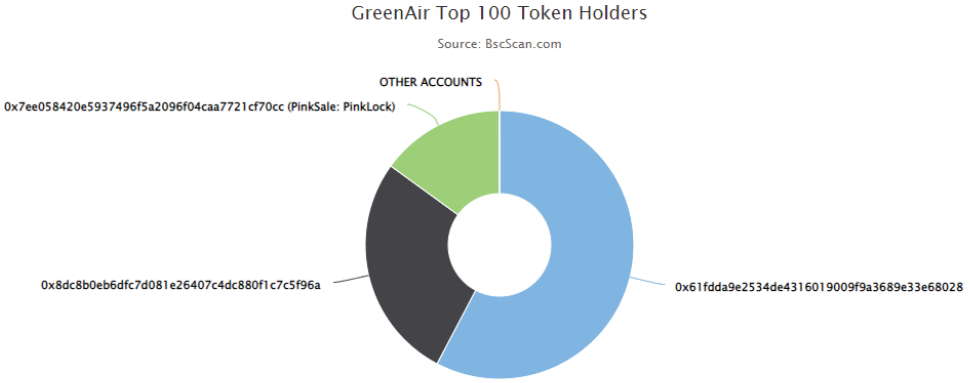
Rug Pull Risk

Liquidity:	15% locked
Holders:	Clean

GREENAIR TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 100 holders collectively own 100.00% (1,000,000,000.00 Tokens) of GreenAir

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 3



(A total of 1,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x61fdda9e2534de4316019009f9a3689e33e68028	577,091,666.66648574	57.7092%
2	0x8dc8b0eb6dfc7d081e26407c4dc880f1c7c5f96a	272,908,333.33351426	27.2908%
3	PinkSale: PinkLock	150,000,000	15.0000%

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.