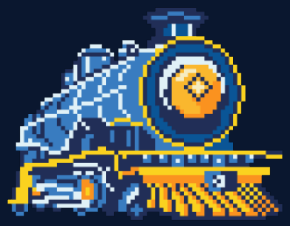
freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Happy Train

$HTR

25/10/2023

TOKEN OVERVIEW

Fees

- Buy fees: 5%
- Sell fees: 5%

Fees privileges

- Can’t change / set fees

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max wallet amount (with threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees

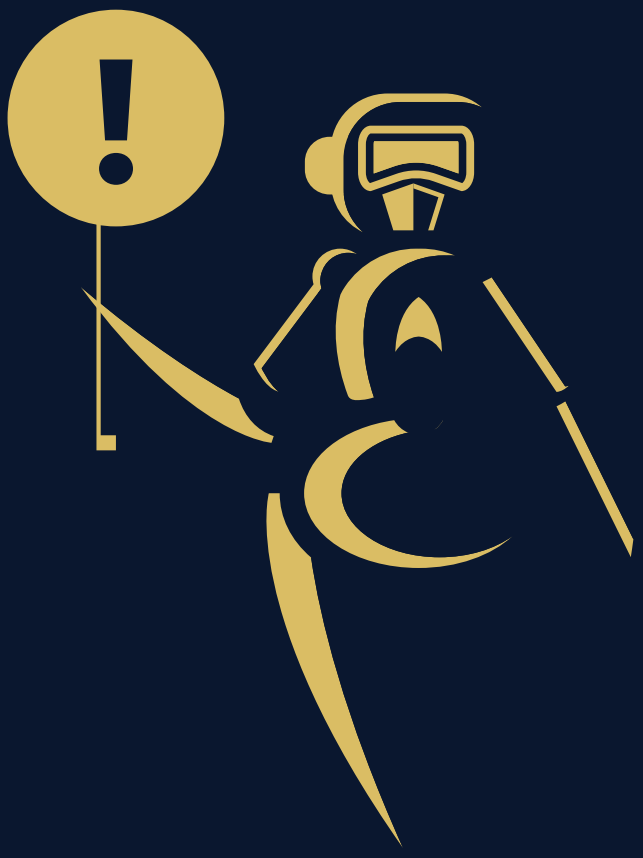
TABLE OF CONTENTS

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

0x0aF2ec0957Cb0FAA0D449C6326c4dD73d78436e7

Network: Ethereum (ETH)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 25/10/2023

WEBSITE DIAGNOSTIC

https://happytrain.io/

0-49 50-89 90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
94	97	95	94	NA

Socials

Twitter

https://twitter.com/HappyTrain_Game

Telegram

https://t.me/happy_train_game

AUDIT OVERVIEW

91

Security Score

96

Static Scan
Automatic scanning for common vulnerabilities

90

ERC Scan
Automatic checks for ERC's conformance

0 High

2 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(s) from tax

```
function excludeFromSwapFee(
      address[] calldata addressesToAdd
   ) public onlyOwner {
      for (uint256 i = 0; i < addressesToAdd.length; i++) {
         isExcludedFromFee[addressesToAdd[i]] = true;
      }
}
```

- Contract owner can change swap settings

```
function changeMinSwapSpreadSum(uint newSumUint256) public onlyOwner {
      minSwapSpreadSum = newSumUint256;
}
```

- Contract owner can change max wallet amount limitation (with threshold)

```
function changeMaxHoldSum(uint _newMaxHoldSum) public onlyOwner {
      require(_newMaxHoldSum >= (totalSupply() / 1000), "Must be at least 0.1%");
      maxHoldSum = _newMaxHoldSum;
}
```

- Contract owner can change wagonbuyerAddress and tokenBuyerAddress addresses

  Current values:

  wagonbuyerAddress : 0x7F70FB57AF2DC84FbA9E105Ed5A2F57cD6Ac2146

  tokenBuyerAddress : 0x0000000000000000000000000000000000000000

```
function changeWagonbuyerAddress(address _address) public onlyOwner {
      wagonbuyerAddress = _address;
}

function changeTokenBuyerAddress(address _address) public onlyOwner {
      tokenBuyerAddress = _address;
}
```

- Missing Zero Address Check
  - HTR: changeWagonbuyerAddress()
  - HTR: changeTokenBuyerAddress()

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

```
function withdrawal(address receiver) public onlyOwner {
    payable(receiver).transfer(address(this).balance);
}
```

- Contract owner has ability to transfer tokens from contract to any address

```
function burnContractTokenBalance() public onlyOwner {
    uint currentTokenBalance = IERC20(tokenAddress).balanceOf(tokenAddress);
    _transfer(address(this), deadAddress, currentTokenBalance);
}
```

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:	5%
Sell fees:	5%
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

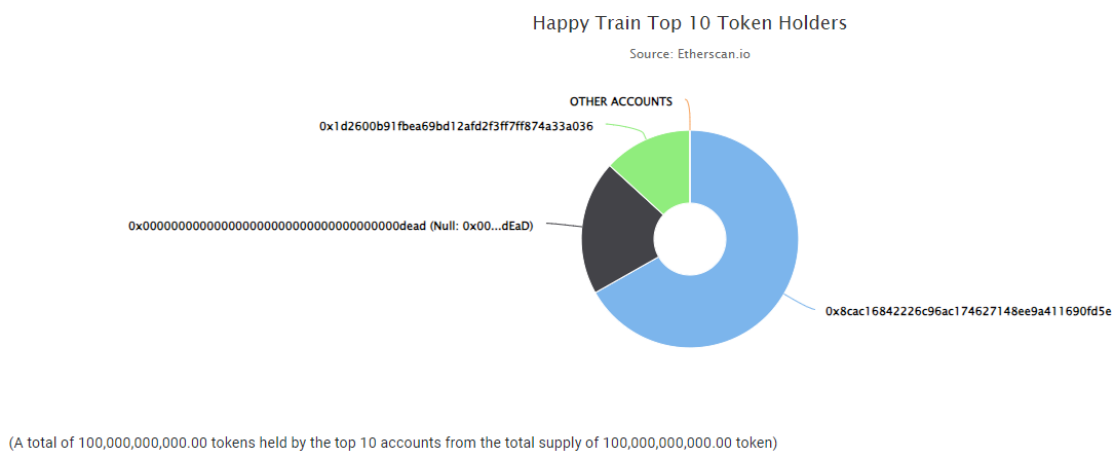
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

HTR TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000,000.00 Tokens) of Happy Train

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 3



(A total of 100,000,000,000.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x8CAC16...1690FD5E	66,802,500,000	66.8025%
2	Null: 0x00...dEaD	20,000,000,000	20.0000%
3	0x1d2600...4a33a036	13,197,500,000	13.1975%

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.