Cyberscope

# Audit Report

# Hubin Network LFT

October 2022

| | |
|---|---|
| Type | BEP20 |
| Network | BSC |
| Address | 0x3a4aD9F27F9162de6104CF768074ea24F7f74CbA |
| Audited by | © cyberscope |

# Table of Contents

# Contract Review

| | |
|---|---|
| **Contract Name** | HubinNetworkNFT |
| **Compiler Version** | v0.8.17+commit.8df45f5f |
| **Optimization** | 200 runs |
| **Licence** | MIT |
| **Explorer** | https://bscscan.com/token/0x3a4aD9F27F9162de6104CF768074ea24F7f74CbA |
| **Symbol** | HBNNFT |

# Source Files

| **Filename** | **SHA256** |
|---|---|
| **contract.sol** | 9c42b70612ec12ac14307afb7f69d011671051382473863 1a9df4ddb028f8ea5 |

# Audit Updates

| | |
|---|---|
| **Initial Audit** | 27th October 2022 |
| **Corrected** | |

# Introduction

The Contract HubinNetworkNFT implements an NFT functionality.

# Roles

The Contract has an admin role. The admin is authorized to

- Configure mint cost.
- Configure minimum mint capability.
- Configure maximum tokes per wallet.
- Configure tokens Uri and file base extension.
- Pause mint functionality.
- Whitelist users.
- Withdraw liquidity from the contract.

Users can mint new tokens.

# Contract Diagnostics

Critical Medium Minor / Informative

| Severity | Code | Description | Status |
|---|---|---|---|
| Medium | MWMV | Max Wallet Mint Violation | Unresolved |
| Minor / Informative | SMT | Stops Mint Transactions | Unresolved |
| Minor / Informative | MC | Missing Check | Unresolved |
| Minor / Informative | L02 | State Variables could be Declared Constant | Unresolved |
| Minor / Informative | L04 | Conformance to Solidity Naming Conventions | Unresolved |
| Minor / Informative | L09 | Dead Code Elimination | Unresolved |
| Minor / Informative | L11 | Unnecessary Boolean equality | Unresolved |
| Minor / Informative | L12 | Using Variables before Declaration | Unresolved |
| Minor / Informative | L14 | Uninitialized Variables in Local Scope | Unresolved |
| Minor / Informative | L15 | Local Scope Variable Shadowing | Unresolved |

# MWMV - Max Wallet Mint Violation

| | |
|---|---|
| **Criticality** | medium |
| **Location** | contract.sol#L1449 |
| **Status** | Unresolved |

## Description

The mint method has some preconditions. One of the preconditions is that a user cannot mint more than the maxMintPerWallet amount. The contract is checking the owner's balance without taking in account the new amount that is going to be minted. As a result, the precondition will break when:

1. The user already holds `maxMintPerWallet - 1` NFTs.
2. The user mints another `maxMintPerWallet - 1` NFTs.
3. After the mint process, the user will hold `(maxMintPerWallet - 1) * 2` tokens.

```
function mint(address _to, uint256 _mintAmount) public payable {
    uint256 supply = totalSupply();
    require(!paused);
    require(_mintAmount > 0);
    require(_mintAmount <= maxMintAmount);
    require(supply + _mintAmount <= maxSupply);
    require(walletOfOwner(msg.sender).length < maxMintPerWallet  );
```

## Recommendation

The contract should take into account the new amount that is going to be minted when it checks the maxMintPerWallet.

# SMT - Stops Mint Transactions

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1447 |
| **Status** | Unresolved |

## Description

The contract owner has the authority to stop mint transactions for all users including the owner. The owner may take advantage of it by calling the `pause` function with the state argument set to true.

```
function mint(address _to, uint256 _mintAmount) public payable {
    uint256 supply = totalSupply();
    require(!paused);
```

## Recommendation

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism preventing a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

# MC - Missing Check

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1507,1511 |
| **Status** | Unresolved |

## Description

The contract is processing variables that have not been properly sanitized and checked that they form the proper shape. These variables may produce vulnerability issues.

The contract does not sanitize the arguments of the functions.

```
function setmaxMintAmount(uint256 _newmaxMintAmount) public onlyOwner {
    maxMintAmount = _newmaxMintAmount;
}

function setmaxMintPerWallet(uint256 _maxMintPerWallet) public onlyOwner {
    maxMintPerWallet = _maxMintPerWallet;
}
```

## Recommendation

The contract should properly check the variables according to the required specifications.

- The variable `_newmaxMintAmount` could be greater than zero.
- The variable `_maxMintPerWallet` could be greater than zero but less than the `_newmaxMintAmount`.

# L02 - State Variables could be Declared Constant

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1425 |
| **Status** | Unresolved |

## Description

Constant state variables should be declared constant to save gas.

```
maxSupply
```

## Recommendation

Add the constant attribute to state variables that never change.

# L04 - Conformance to Solidity Naming Conventions

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1519,1526,1534,863,1445,1507,1515,1464,1511,1503,1530 |
| **Status** | Unresolved |

## Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_newBaseExtension
_state
_user
_data
_to
_newmaxMintAmount
_newBaseURI
_mintAmount
_owner
...
```

## Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions.

# L09 - Dead Code Elimination

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L576,239,398,461,559,223,1002,541,481,505,764,442,524,429 |
| **Status** | Unresolved |

## Description

Functions that are not used in the contract, and make the code's size bigger.

```
verifyCallResult
toHexString
sendValue
functionCallWithValue
functionDelegateCall
_burn
functionStaticCall
_baseURI
functionCall
...
```

## Recommendation

Remove unused functions.

# L11 - Unnecessary Boolean equality

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1445 |
| **Status** | Unresolved |

## Description

The comparison to boolean constants is redundant. Boolean constants can be used directly and do not need to be compared to true or false.

```
whitelisted[msg.sender] != true
```

## Recommendation

Remove the equality to the boolean constant.

# L12 - Using Variables before Declaration

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1084,1086 |
| **Status** | Unresolved |

## Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
retval
reason
```

## Recommendation

The variables should be declared before any usage of them.

# L14 - Uninitialized Variables in Local Scope

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1471 |
| **Status** | Unresolved |

## Description

The are variables that are defined in the local scope and are not initialized.

```
i
```

## Recommendation

All the local scoped variables should be initialized.

# L15 - Local Scope Variable Shadowing

| | |
|---|---|
| **Criticality** | minor / informative |
| **Location** | contract.sol#L1464,1433,1432 |
| **Status** | Unresolved |

## Description

The are variables that are defined in the local scope containing the same name from an upper scope.

```
_owner
_symbol
_name
```

## Recommendation

The local variables should have different names from the upper scoped variables.

# Contract Functions

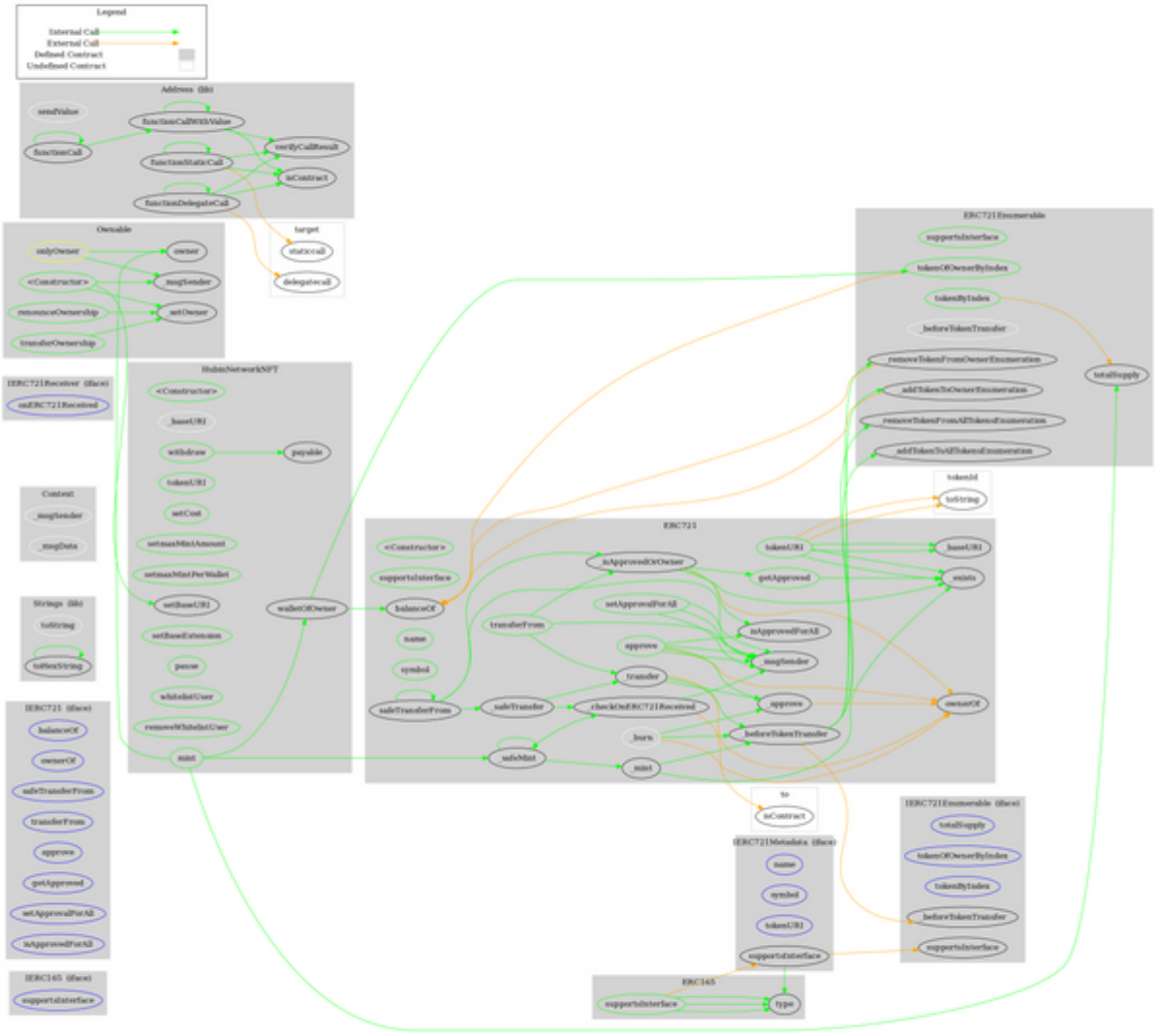
| Contract | Type | Bases | | |
|---|---|---|---|---|
| | Function Name | Visibility | Mutability | Modifiers |
| | | | | |
| IERC165 | Interface | | | |
| | supportsInterface | External | | - |
| | | | | |
| IERC721 | Interface | IERC165 | | |
| | balanceOf | External | | - |
| | ownerOf | External | | - |
| | safeTransferFrom | External | ✓ | - |
| | transferFrom | External | ✓ | - |
| | approve | External | ✓ | - |
| | getApproved | External | | - |
| | setApprovalForAll | External | ✓ | - |
| | isApprovedForAll | External | | - |
| | safeTransferFrom | External | ✓ | - |
| | | | | |
| Strings | Library | | | |
| | toString | Internal | | |
| | toHexString | Internal | | |
| | toHexString | Internal | | |
| | | | | |
| Context | Implementation | | | |
| | _msgSender | Internal | | |
| | _msgData | Internal | | |
| | | | | |
| IERC721Receiver | Interface | | | |
| | onERC721Received | External | ✓ | - |
| | | | | |
| IERC721Metadata | Interface | IERC721 | | |

| | | | | |
|---|---|---|---|---|
| | name | External | | - |
| | symbol | External | | - |
| | tokenURI | External | | - |
| | | | | |
| **Address** | Library | | | |
| | isContract | Internal | | |
| | sendValue | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCall | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | functionCallWithValue | Internal | ✓ | |
| | functionStaticCall | Internal | | |
| | functionStaticCall | Internal | | |
| | functionDelegateCall | Internal | ✓ | |
| | functionDelegateCall | Internal | ✓ | |
| | verifyCallResult | Internal | | |
| | | | | |
| **ERC165** | Implementation | IERC165 | | |
| | supportsInterface | Public | | - |
| | | | | |
| **ERC721** | Implementation | Context, ERC165, IERC721, IERC721Metadata | | |
| | <Constructor> | Public | ✓ | - |
| | supportsInterface | Public | | - |
| | balanceOf | Public | | - |
| | ownerOf | Public | | - |
| | name | Public | | - |
| | symbol | Public | | - |
| | tokenURI | Public | | - |
| | _baseURI | Internal | | |
| | approve | Public | ✓ | - |
| | getApproved | Public | | - |
| | setApprovalForAll | Public | ✓ | - |

| | | | | |
|---|---|---|---|---|
| | isApprovedForAll | Public | | - |
| | transferFrom | Public | ✓ | - |
| | safeTransferFrom | Public | ✓ | - |
| | safeTransferFrom | Public | ✓ | - |
| | _safeTransfer | Internal | ✓ | |
| | _exists | Internal | | |
| | _isApprovedOrOwner | Internal | | |
| | _safeMint | Internal | ✓ | |
| | _safeMint | Internal | ✓ | |
| | _mint | Internal | ✓ | |
| | _burn | Internal | ✓ | |
| | _transfer | Internal | ✓ | |
| | _approve | Internal | ✓ | |
| | _checkOnERC721Received | Private | ✓ | |
| | _beforeTokenTransfer | Internal | ✓ | |
| | | | | |
| **IERC721Enumerable** | Interface | IERC721 | | |
| | totalSupply | External | | - |
| | tokenOfOwnerByIndex | External | | - |
| | tokenByIndex | External | | - |
| | | | | |
| **ERC721Enumerable** | Implementation | ERC721, IERC721Enumerable | | |
| | supportsInterface | Public | | - |
| | tokenOfOwnerByIndex | Public | | - |
| | totalSupply | Public | | - |
| | tokenByIndex | Public | | - |
| | _beforeTokenTransfer | Internal | ✓ | |
| | _addTokenToOwnerEnumeration | Private | ✓ | |
| | _addTokenToAllTokensEnumeration | Private | ✓ | |
| | _removeTokenFromOwnerEnumeration | Private | ✓ | |
| | _removeTokenFromAllTokensEnumeration | Private | ✓ | |

| | | | | |
|---|---|---|---|---|
| **Ownable** | Implementation | Context | | |
| | <Constructor> | Public | ✓ | - |
| | owner | Public | | - |
| | renounceOwnership | Public | ✓ | onlyOwner |
| | transferOwnership | Public | ✓ | onlyOwner |
| | _setOwner | Private | ✓ | |
| | | | | |
| **HubinNetwork NFT** | Implementation | ERC721Enumerable, Ownable | | |
| | <Constructor> | Public | ✓ | ERC721 |
| | _baseURI | Internal | | |
| | mint | Public | Payable | - |
| | walletOfOwner | Public | | - |
| | tokenURI | Public | | - |
| | setCost | Public | ✓ | onlyOwner |
| | setmaxMintAmount | Public | ✓ | onlyOwner |
| | setmaxMintPerWallet | Public | ✓ | onlyOwner |
| | setBaseURI | Public | ✓ | onlyOwner |
| | setBaseExtension | Public | ✓ | onlyOwner |
| | pause | Public | ✓ | onlyOwner |
| | whitelistUser | Public | ✓ | onlyOwner |
| | removeWhitelistUser | Public | ✓ | onlyOwner |
| | withdraw | Public | Payable | onlyOwner |

# Contract Flow



# Summary

The HubinNetworkNFT contract implements an NFT mechanism. This audit investigates security issues and mentions business logic concerns and potential improvements.

# Disclaimer

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment.

Cyberscope team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed.

The Cyberscope team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Cyberscope receive a payment to manipulate those results or change the awarding badge that we will be adding in our website.

Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token.

The Cyberscope team disclaims any liability for the resulting losses.

# About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io