WHITE HAT DAO

Carbon Collectible℠ NFTs

Smart Contract Audit Report

carboncollectiblenfts.com

By White Hat DAO

www.whitehatdao.com

Date: 11/03/2022

WHITE HAT DAO

Table of Contents

WHITE HAT DAO

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report.

In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions.

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and White Hat DAO and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives)

White Hat DAO owes no duty of care towards you or any other person, nor does White Hat DAO make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and White Hat DAO hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report.

Except and only to the extent that it is prohibited by law, White Hat DAO hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against White Hat DAO, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on

this report. The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security.

No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.

WHITE HAT DAO

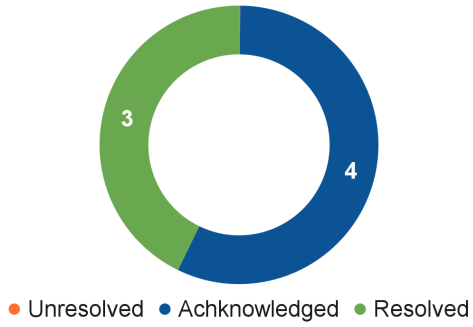
Executive Summary

White Hat DAO was contracted by "The Carbon Collectible NFTs" team to conduct a smart contract security audit. This report presents the findings of the security assessment conducted between Feb. 16, 2022 and March 4th, 2022. During this audit, the team manually reviewed one (1) smart contract and analyzed it with static analysis tools.

Based on results of the audit, the customers' smart contract safety rating is:

The White Hat DAO audit team has found 7 issues which include five (5) Major issues, one (1) Minor issue and one (1) Informational suggestion. Most of the Major issues fell within the **"Centralization/Privilege" & "Reentrancy"** categories. For a full list of these issues please refer to the Findings section of the report.

Total Issues	7 (3 Resolved)
Critical Issues	0 (0 Resolved)
Major Issues	5 (1 Resolved)
Medium Issues	0 (0 Resolved)
Minor Issues	1 (1 Resolved)
Informational	1 (1 Resolved)



WHITE HAT DAO

Summary of Findings

The most prominent audit findings were the Major issues around the **"Centralization/Privilege" & "Reentrancy"**. There were also some Minor/Informational vulnerabilities around "Gas Optimization" and "Coding Standards". All issues regarding severities & vulnerabilities discovered by the audit process are listed below:

Issue ID	Issue Title	Category	Severity	Status
CCNFT-01	Withdraws could technically fail	Loss of Funds	Major	Acknowledged
CCNFT-02	Centralized unbound setter functions	Centralization/ Privilege	Major	Acknowledged (Plan to resolve)
CCNFT-03	Centralized unbound setter functions	Centralization/ Privilege	Major	Acknowledged (Plan to resolve)
CCNFT-04	Reentrancy	Centralization/ Privilege	Major	Resolved
CCNFT-05	Single Point of Failure	Centralization/ Privilege	Major	Acknowledged (Plan to resolve)
CCNFT-06	No receive function	Code Standards	Minor	Resolved
CCNFT-07	Redundant import	Gas Optimization	Informational	Resolved

WHITE HAT DAO

Introduction

This security audit assessment has been prepared for "The Carbon Collectible NFTs". The purpose of this audit is to document and expose any safety concerns and vulnerabilities found in the source code which has been reviewed by the White Hat DAO Audit team. This review includes any contract dependencies in scope that were not part of an officially recognized library. Comprehensive tests have been conducted, utilizing manual code review, static analysis, and techniques.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors
- Assessing the codebase to ensure compliance with current best practices and industry standards
- Ensuring contract logic meets the specifications and intentions of the client. Cross-referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Thorough line-by-line manual review of the entire codebase by industry experts
- Reviewing unit tests to ensure full coverage of the codebase

The Project Summary, Scope, Audit Details and Methodology of the audit is described within this document.

The security audit assessment can result in findings that range from critical to informational. The White HAt DAO Audit Team recommends that the client address findings contained herein to ensure a high level of security standards. This information can be found in the Findings section of the report.

WHITE HAT DAO

Project Summary

Project	The Carbon Collectible Nfts
Description	This Project provides digital and virtual rights to 1 dedicated hectare of mature forestry and 5 Web3 carbon offsets but no physical rights to the land or trees.
Website	https://carboncollectiblenfts.com/
Platform	Polygon
Language used	Solidity
Codebase	https://github.com/crypto-dev-nft/carbon-collective_contract/blob/main/contract.sol
Commit:	5de1fc11d9d3b2f0a5eb13e1649b4cd4820c1671

Project Scope

White Hat DAO was commissioned by The Carbon Collectible NFTs to perform security assessments on the smart contract listed below:

Source Code	Acknowledgement	SHA-256
carbon-collective_contract/contract.sol	Accepted	e1d3d1d1b4ba968120efc7ccdba7f4bd00056a8994beb19c2f937cf02b839906

WHITE HAT DAO

Audit Details

Delivery Date	11/03/2022
Received Date	16/02/2022
Key Components	carbon-collective_contract/contract.sol

Methodology

The White Hat DAO Audit team reviewed the code base provided by The Carbon Collectible NFTs team between Feb. 16, 2022 and March 4, 2022. The team conducted the assessment based on the repository at commit [13bae6d6e49f8d186e19fa5b243f85f379b99903](13bae6d6e49f8d186e19fa5b243f85f379b99903).

The White Hat DAO Audit team launched the audit by analyzing the specifications of the project and focusing on the key areas of interest while evaluating the documentation. The code was then manually reviewed in an attempt to identify potential vulnerabilities and verify that the adh code has good unit tests coverage. The White Hat DAO Audit team wrote some unit cases to test some edge cases. Automated analysis of the codebase was performed and results were reviewed.

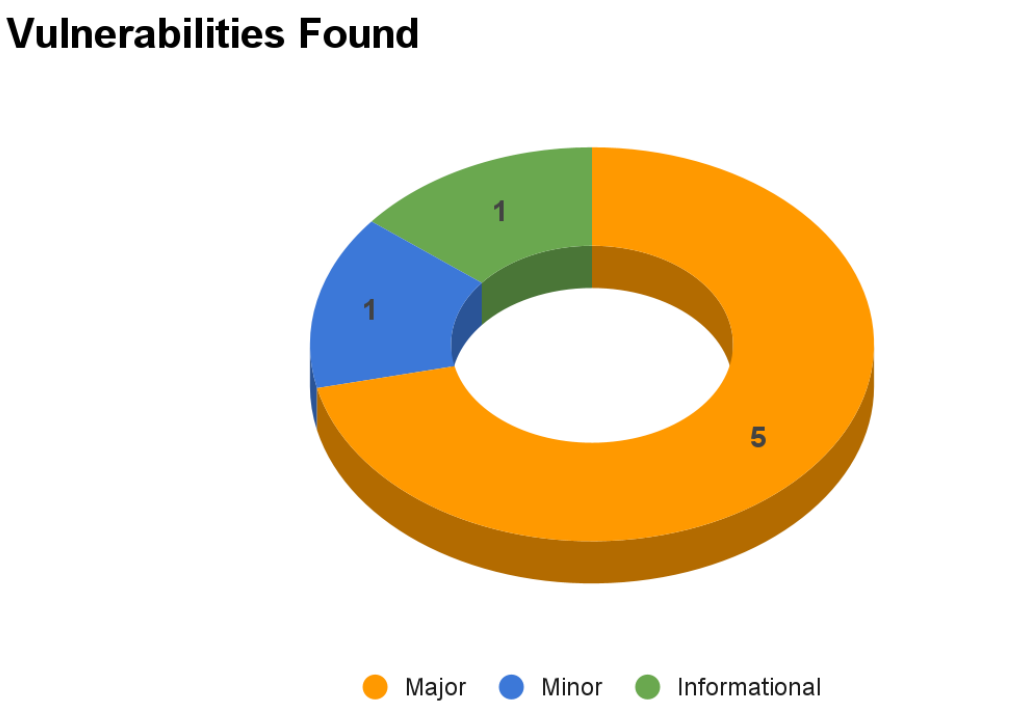
WHITE HAT DAO

The smart contract provided by The Carbon Collectible NFTs team was scanned for vulnerabilities. The following is the list of some of the vulnerabilities that were considered during the audit of the smart contract:

No	Vulnerability Tests	Status
# 1	Access Control	Passed
# 2	Arbitrary token minting	Passed
# 3	Business Logics Review	Passed
# 4	Centralization of power	Passed
# 5	Code clones, functionality duplication	Passed
# 6	Conditional Completion attack	Passed
# 7	Costly Loop	Passed
# 8	Ownership Takeover	Passed
# 9	Redundant fallback function	Passed
# 10	Reentrancy	Passed
# 11	Remote code execution	Passed
# 12	User Balances manipulation	Passed
# 13	Logic Flaws	Passed
# 14	Scoping and Declarations	Passed
# 15	Integer Overflow and Underflow attacks	Passed

Findings

The White Hat DAO Audit team found five (5) Major issues, one (1) Minor issue and one (1) Informational suggestion. Most of the Major issues fell under the “Centralization/Privilege" & “Reentrancy" categories. The one Minor vulnerability was in the “Coding Standards" category. Additional information on these vulnerabilities is provided in the following sections.



Critical - 0 | Major - 5 | Medium Issue - 0 | Minor - 1 | Informational -1

WHITE HAT DAO

Severity Definitions

Severity	Definitions
Critical	Critical vulnerabilities have a catastrophic impact on the security of the project. They can lead to loss, data manipulation, take over, etc. It is strongly recommended to fix these vulnerabilities.
Major	Major vulnerabilities have a significant impact on the security of the project. They can lead to loss, data manipulation, take over, etc. It is strongly recommended to fix these vulnerabilities.
Medium	Medium vulnerabilities are important to fix. These vulnerabilities alone can't lead to asset loss or data manipulation. However, medium vulnerabilities can be chained to create a more severe vulnerability. It is highly recommended to review and address these vulnerabilities.
Minor	Minor vulnerabilities are mostly related to outdated, unused code snippets and don't have a significant impact on execution.
Informational	Informational vulnerabilities don't pose an immediate risk but are relevant to security best practices. They could be code-style violations and informational statements that don't affect smart contract execution. They may be able to be ignored.

Critical Vulnerabilities

No Critical severity vulnerabilities were found.

Major Vulnerabilities

CCNFT-01 | Withdraws could technically fail

Type: Loss of Funds
Level: Major

Description: Withdrawals could technically fail if **withdrawAddresses** contains a contract address that reverts on receive or fallback.

Recommendation: Include a safe address and set it via constructor or function.

Details:

contract.sol: (N/A)

```
address payable ourSafe;

function withdrawFailSafe() public onlyOwner nonReentrant {
   uint256 balance = address(this).balance;
   (bool success, ) = ourSafe.call{ value: balance }("");
   require(success, "Transfer Failed");
}

function withdrawTokenFailSafe(address _tokenContract) public onlyOwner
nonReentrant {
        IERC20 tokenContract = IERC20(_tokenContract);
        uint256 balance = tokenContract.balanceOf(address(this));
        tokenContract.transfer(ourSafe, balance);
    }
```

Update: Client doesn't believe this is relevant in their case. They have full control of withdrawAddresses array, and they know it's not a contract address, so there is no possibility that there will be a revert on receive or fallback.

CCNFT-02 | Centralized unbound setter functions

Type: Centralization/Privilege
Level: Major

Description: These functions are called by the owner at any point in time with no restrictions, a misuse of these functions mid-sale or right before destruction will result in an unusable ecosystem, or changing the baseURI by the owner to manipulate the NFT allocations/order.

Recommendation: Restrict the call to these functions to only before the sale starts, once the sale has started these functions should never change, users should be aware of what they are buying on the sale with no way for the owner to change that mid or post-sale.

Details:

File: contract.sol (line 94)

```
function setBaseURI(string memory baseURI_) external onlyOwner {
    setBaseURI(baseURI_);
}
```

Update: Client will add a URI freeze function. This allows them to freeze the URI after the mint and it will not be possible to unfreeze. They will move to a multisig wallet for the Owner role.

Details of update:

File: contract.sol

```
function setBaseURI(string memory _baseTokenURI) external onlyOwner {
    require(freeze == false, "CryptoMofayas: uri is frozen");
    baseURI = _baseTokenURI;
}

function freezeURI() external onlyOwner {
    freeze = true;
}
```

CCNFT-03 | Centralized unbound setter functions

Type: Centralization/Privilege
Level: Major

Description: These functions are called by the owner at any point in time with no restrictions, a misuse of these functions mid-sale or right before destruction will result in allowing cheap mints or rendering tokens unmintable.

Recommendation: Restrict the call to these functions to only before the sale starts, once the sale has started these functions should never change, users should be aware of what they are buying on the sale with no way for the owner to change that mid or post-sale.

Details:

File: `contract.sol (line 99)`

```
function setPrice(string memory newPrice) external onlyOwner {
    price = newPrice;
}
```

Update: Client is planning to resolve this issue using a multi-sig wallet address as owner.

CCNFT-04 | Reentrancy

Type: Centralization/Privilege
Level: Major

Description: Reentrancy possible in for loop.

Recommendation: Transfer may be unreliable, recommended rewrite below also adds the already imported "nonreentrant" modifier. Although I was unable to re enter these functions with an attack contract, this feels safer.

Details:

File: contract.sol(line 119) ***Existing Funcion***

```
 function withdraw() public {
    uint256 balance = address(this).balance;
    uint256 remainingBalance = balance;
    for (uint8 i = 0; i < withdrawAddresses.length - 1; i++) {
        uint256 valueToTransfer = (perThousandPerAddress[i] * balance) / 1000;
        withdrawAddresses[i].transfer(valueToTransfer);
        remaningBalance -= valueToTransfer;
    }
    withdrawAddresses[withdrawAddresses.length -1].transfer(remainingBalance);
 }
```

File: contract.sol(line 119) ***Recommended Funcion***

```
function withdraw() public nonReentrant{
        uint256 balance = address(this).balance;
        uint256 remainingBalance = balance;
        require(remainingBalance > 0, "There is no token to withdraw.");
        for (uint8 i = 0; i < withdrawAddresses.length - 1; i++) {
            uint256 valueToTransfer = (perThousandPerAddress[i] * balance) /
                1000;
            (bool successGroup, ) =
withdrawAddresses[i].call{value:valueToTransfer}("");
                require(successGroup, "Transfer failed.");
            remainingBalance -= valueToTransfer;
        }
        (bool success, ) = withdrawAddresses[withdrawAddresses.length - 1].call
        {value: remainingBalance}("");
        require(success, "Transfer failed.");
    }
```

Update: Client has resolved this in this commit:
https://github.com/crypto-dev-nft/carbon-collective_contract/commit/a37859837b46e2025f49db56f18d933bfad4754d

CCNFT-05 | Single Point of Failure

Type: Centralization/Privilege
Level: Major

Description: The whole project is designed around one level of access control (owner), the owner has all the privileged calls if the owner's wallet gets compromised an attacker can take over the whole project.

Recommendation: We strongly recommend centralized privileges or roles in the protocol to be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., Multisignature wallets

Here are some feasible action items:

1. Assignment of privileged roles (owner) to multi-signature wallets to prevent a single point of failure due to the private key

Details:

File: `All files`

Update: Client will move to a multi signature wallet for the owner role.

Medium Vulnerabilities

No Medium severity vulnerabilities were found.

Minor Vulnerabilities

CCNFT-06 | No receive function

Type: Coding Standards
Level: Minor

Description: There is no receiver function in case of donation or user error.

Recommendation: Add receive function.

Details:

File: `contract.sol (new line)`

```
    receive() external payable {}
```

Update: Client has resolved this in this commit:
https://github.com/crypto-dev-nft/carbon-collective_contract/commit/341294b68cf107862f6d260bef888d2146798371

Informational Vulnerabilities

CCNFT-07 | Redundant imports

Type: Gas Optimization
Level: Minor

Description: The contract is importing both ERC721 and ERC721Enumerable.

Recommendation: Remove the ERC721 import and use only the ERC721Enumerable. This will allow for the removal of **beforeTokenTransfer(line 85)** and **supportsInterface(line 109)** overrides.

Details:

File: `contract.sol (line 11)`

```
contract CarbonCollectibleCharacters is
    ERC721Enumerable,
    Ownable,
    ReentrancyGuard
{
```

Update: Client has resolved this in this commit:
https://github.com/crypto-dev-nft/carbon-collective_contract/commit/438badf138435687b5ddee71b96a3e87121e6997

Conclusion

White Hat DAO has worked with The Carbon Collectible NFTs team to perform this audit. There 1 smart contract reviewed during this audit. The smart contracts were manually reviewed and analyzed with static analysis tools. The findings of these reviews were provided in this report.

No unit tests were provided. We constructed some unit tests to test edge cases. The code was commented well. Comments are helpful in understanding the overall architecture and the logic flow of the contracts.

During the audit of the smart contract provided by The Carbon Collectible NFTs team, the White Hat DAO audit team found a total of five (5) Major, one (1) Minor, and one (1) Informational vulnerabilities, which have been detailed in this report.

Update: Three of the issues have been resolved (CCNFT-04, CCNFT-06, CCNFT-07). The Carbon Collectible NFTs team has acknowledged the remaining issues and are planning to resolve CCNFT-02, CCNFT-03 and CCNFT-05 by using a multi-sig wallet address as the owner. Please refer to the findings above for more details.

WHITE HAT DAO

Change Log

- 28-02-2022 - Initial report.
- 02-03-2022 - Updated Summary of Findings.
- 11-03-2022 - Added client updates.

Audited by

www.whitehatdao.com