



# Smart Contract Security Audit Report



# Table Of Contents

<b>1 Executive Summary</b>	_____
<b>2 Audit Methodology</b>	_____
<b>3 Project Overview</b>	_____
3.1 Project Introduction	_____
3.2 Vulnerability Information	_____
<b>4 Code Overview</b>	_____
4.1 Contracts Description	_____
4.2 Visibility Description	_____
4.3 Vulnerability Summary	_____
<b>5 Audit Result</b>	_____
<b>6 Statement</b>	_____

# 1 Executive Summary

On 2022.04.27, the SlowMist security team received the team's security audit application for CheersUp, developed the audit plan according to the agreement of both parties and the characteristics of the project, and finally issued the security audit report.

The SlowMist security team adopts the strategy of "white box lead, black, grey box assists" to conduct a complete security test on the project in the way closest to the real attack.

The test method information:

Test method	Description
Black box testing	Conduct security tests from an attacker's perspective externally.
Grey box testing	Conduct security testing on code modules through the scripting tool, observing the internal running status, mining weaknesses.
White box testing	Based on the open source code, non-open source code, to detect whether there are vulnerabilities in programs such as nodes, SDK, etc.

The vulnerability severity level information:

Level	Description
Critical	Critical severity vulnerabilities will have a significant impact on the security of the DeFi project, and it is strongly recommended to fix the critical vulnerabilities.
High	High severity vulnerabilities will affect the normal operation of the DeFi project. It is strongly recommended to fix high-risk vulnerabilities.
Medium	Medium severity vulnerability will affect the operation of the DeFi project. It is recommended to fix medium-risk vulnerabilities.
Low	Low severity vulnerabilities may affect the operation of the DeFi project in certain scenarios. It is suggested that the project team should evaluate and consider whether these vulnerabilities need to be fixed.
Weakness	There are safety risks theoretically, but it is extremely difficult to reproduce in engineering.

Level	Description
Suggestion	There are better practices for coding or architecture.

## 2 Audit Methodology

The security audit process of SlowMist security team for smart contract includes two steps:

Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using automated analysis tools.

Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

Serial Number	Audit Class	Audit Subclass
1	Overflow Audit	-
2	Reentrancy Attack Audit	-
3	Replay Attack Audit	-
4	Flashloan Attack Audit	-
5	Race Conditions Audit	Reordering Attack Audit
6	Permission Vulnerability Audit	Access Control Audit
		Excessive Authority Audit

Serial Number	Audit Class	Audit Subclass
7	Security Design Audit	External Module Safe Use Audit
		Compiler Version Security Audit
		Hard-coded Address Security Audit
		Fallback Function Safe Use Audit
		Show Coding Security Audit
		Function Return Value Security Audit
		External Call Function Security Audit
		Block data Dependence Security Audit
		tx.origin Authentication Security Audit
8	Denial of Service Audit	-
9	Gas Optimization Audit	-
10	Design Logic Audit	-
11	Variable Coverage Vulnerability Audit	-
12	"False Top-up" Vulnerability Audit	-
13	Scoping and Declarations Audit	-
14	Malicious Event Log Audit	-
15	Arithmetic Accuracy Deviation Audit	-
16	Uninitialized Storage Pointer Audit	-

### 3 Project Overview

## 3.1 Project Introduction

### Audit Version

<https://github.com/Base-Labs/contracts-audit>

commit:f561ce6ab24363467ad9a95c96f4b00a042b117e

### Fixed Version

<https://github.com/Base-Labs/contracts>

commit:cca5d884360c0772976361385d974b48b8470e9b

## 3.2 Vulnerability Information

The following is the status of the vulnerabilities found in this audit:

NO	Title	Category	Level	Status
N1	Missing event record	Others	Suggestion	Fixed
N2	Access control issue	Authority Control Vulnerability	Low	Confirmed
N3	Random number problem	Others	Low	Confirmed

## 4 Code Overview

### 4.1 Contracts Description

The main network address of the contract is as follows:

The project's contracts were fully open-sourced on GitHub, and currently, only the CheersUpPeriod contract was deployed on the mainnet and verified on etherscan.

GitHub address:

<https://github.com/Base-Labs/contracts>

CheersUpPeriod contract address:

<https://etherscan.io/address/0xa5bb28eccc6134f89745e34ec6ab5d5bcb16dad7#code>

## 4.2 Visibility Description

The SlowMist Security team analyzed the visibility of major contracts during the audit, the result as follows:

CheersUpPeriod			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC721A
giveaway	External	Can Modify State	onlyOwner nonReentrant
whitelistSale	External	Payable	callerIsUser nonReentrant
publicSale	External	Payable	callerIsUser nonReentrant
_sale	Internal	Can Modify State	-
reveal	External	Can Modify State	callerIsUser nonReentrant
refundExcessPayment	Private	Can Modify State	-
withdraw	External	Can Modify State	onlyOwner nonReentrant
isWhitelistSaleEnabled	Public	-	-
isPublicSaleEnabled	Public	-	-
isRevealEnabled	Public	-	-
isWhitelistAddress	Public	-	-

CheersUpPeriod			
getPublicSalePrice	Public	-	-
getWhitelistSalePrice	Public	-	-
totalMinted	Public	-	-
_baseURI	Internal	-	-
setBaseURI	External	Can Modify State	onlyOwner
setWhitelistSaleConfig	External	Can Modify State	onlyOwner
setPublicSaleConfig	External	Can Modify State	onlyOwner
setRevealConfig	External	Can Modify State	onlyOwner
_beforeTokenTransfers	Internal	Can Modify State	-
emergencyPause	External	Can Modify State	onlyOwner notSealed
unpause	External	Can Modify State	onlyOwner notSealed
sealContract	External	Can Modify State	onlyOwner

Crypto			
Function Name	Visibility	Mutability	Modifiers
base58	Public	-	-
cidv0	Public	-	-

CheersUp			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC721



CheersUp			
mint	Public	Can Modify State	-
getRandomTokenId	Internal	Can Modify State	-
unsafeRandom	Internal	-	-
refund	External	Can Modify State	callerIsUser nonReentrant
deposit	External	Payable	onlyOwner nonReentrant
withdraw	External	Can Modify State	onlyOwner nonReentrant
totalMinted	Public	-	-
isRefundEnabled	Public	-	-
tokenURI	Public	-	-
setProvenance	External	Can Modify State	onlyOwner
setRevealingURI	External	Can Modify State	onlyOwner
setCUCPContractAddress	External	Can Modify State	onlyOwner
setMaintainerAddress	External	Can Modify State	onlyOwner
setRefundConfig	External	Can Modify State	onlyOwner
setTokenHash	Public	Can Modify State	atLeastMaintainer
batchSetTokenHash	External	Can Modify State	atLeastMaintainer
_beforeTokenTransfer	Internal	Can Modify State	-
emergencyPause	External	Can Modify State	onlyOwner notSealed
unpause	External	Can Modify State	onlyOwner notSealed
sealContract	External	Can Modify State	onlyOwner

## 4.3 Vulnerability Summary

### [N1] [Suggestion] Missing event record

Category: Others

#### Content

Modifying sensitive parameters in contracts lacks corresponding event records.

```
function setProvenance(string memory provenance_) external onlyOwner {
    provenance = provenance_;
}
function setCUCPContractAddress(address address_) public onlyOwner {
    cucpContractAddress = address_;
}
```

```
function setWhitelistSaleConfig(WhitelistSaleConfig calldata config_) external
onlyOwner {
    require(config_.price > 0, "sale price must greater than zero");
    whitelistSaleConfig = config_;
}
function setPublicSaleConfig(PublicSaleConfig calldata config_) external onlyOwner {
    require(config_.price > 0, "sale price must greater than zero");
    publicSaleConfig = config_;
}
function setRevealConfig(RevealConfig calldata config_) external onlyOwner {
    revealConfig = config_;
}
```

#### Solution

It is recommended to add corresponding event records.

#### Status

Fixed; The project team has added an event record.

### [N2] [Low] Access control issue

## Category: Authority Control Vulnerability

### Content

The Owner role has the right to call the giveaway function to airdrop any number of blind boxes to any address at any time.

```
function giveaway(address address_, uint64 numberOfTokens_) external onlyOwner
nonReentrant {
    require(address_ != address(0), "zero address");
    require(numberOfTokens_ > 0, "invalid number of tokens");
    require(totalMinted() + numberOfTokens_ <= MAX_TOKEN, "max supply exceeded");
    _safeMint(address_, numberOfTokens_);
}
```

The Owner role has the right to modify the cucpContractAddress contract address through the setCUCPContractAddress function, and unauthorized modification will result in the user's blind box being unable to be opened or arbitrarily modifying the blind box.

```
function setCUCPContractAddress(address address_) public onlyOwner {
    cucpContractAddress = address_;
}
```

The Owner role has the right to modify the price of the blind box purchased by the user through the setWhitelistSaleConfig function and the setPublicSaleConfig function at any time. If the price of the blind box is modified after the sale starts, the user's transaction will fail.

```
function setWhitelistSaleConfig(WhitelistSaleConfig calldata config_) external
onlyOwner {
    require(config_.price > 0, "sale price must greater than zero");
    whitelistSaleConfig = config_;
}
function setPublicSaleConfig(PublicSaleConfig calldata config_) external onlyOwner {
    require(config_.price > 0, "sale price must greater than zero");
    publicSaleConfig = config_;
}
```

The Owner has the right to modify the revealConfig parameter through the setRevealConfig function, which includes the address of the contract that opens the blind box, and the time and closing time of the blind box sale.

```
function setRevealConfig(RevealConfig calldata config_) external onlyOwner {
    revealConfig = config_;
}
```

## Solution

1. It is recommended to hand over the authority of the Owner role to the governance contract or TimeLock management, at least multi-sign should be used and a limit on the number of airdrops that can be added.
2. It is recommended to limit the time for modifying parameters, and parameters cannot be modified during the period from the start of the sale to the end of the sale.

## Status

Confirmed; The project team's management of the owner role adopts the form of multi signing under the chain. After communication, the project team promises that all sensitive parameter modifications are only used to deal with emergencies and will not be modified privately during the normal operation of the contract.

## [N3] [Low] Random number problem

### Category: Others

### Content

The tokenId in the contract is related to the properties of the NFT. The tokenId is randomly generated through the random number on the chain, which will cause the random number to be predicted.

```
function mint(address address_, uint256 cucpTokenId_) public returns(uint256) {
    require(_msgSender() == cucpContractAddress, "not authorized");
    require(_numberMinted + 1 <= MAX_TOKEN, "mint would exceed max supply");
    uint256 tokenId = randomToken(address_);
    _safeMint(address_, tokenId);
    unchecked {
        _numberMinted += 1;
    }
}
```

```

    }
    emit CheersUpRevealed(cucpTokenId_, tokenId);
    return tokenId;
}

function randomToken(address address_) internal returns (uint256) {
    unchecked {
        uint totalSize = MAX_TOKEN - _numberMinted;
        uint index = uint(keccak256(abi.encodePacked(_numberMinted, address_,
block.difficulty, block.timestamp))) % totalSize;
        uint value = 0;
        if (randIndices[index] != 0) {
            value = randIndices[index];
        } else {
            value = index;
        }
        if (randIndices[totalSize - 1] == 0) {
            randIndices[index] = totalSize - 1;
        } else {
            randIndices[index] = randIndices[totalSize - 1];
        }
        return value;
    }
}

```

## Solution

It is recommended to use chainlink to generate random numbers or use hashes of future blocks to generate random numbers.

## Status

Confirmed; After communication, it is learned that the project team will keep the rarity corresponding to the randomly selected tokenId during the sale and not disclose it. In this way, the attacker will not be able to judge the value corresponding to his randomly selected tokenId without obtaining the rarity data, so cannot launch the attack.

## 5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002204290003	SlowMist Security Team	2022.04.27 - 2022.04.29	Low Risk

Summary conclusion: The SlowMist security team use a manual and SlowMist team's analysis tool to audit the project, during the audit work we found 2 low risks and 1 suggestion vulnerability. And 2 low risks were confirmed 1 suggestion vulnerability was fixed; The code was deployed to the mainnet.

## 6 Statement

SlowMist issues this report with reference to the facts that have occurred or existed before the issuance of this report, and only assumes corresponding responsibility based on these.

For the facts that occurred or existed after the issuance, SlowMist is not able to judge the security status of this project, and is not responsible for them. The security audit analysis and other contents of this report are based on the documents and materials provided to SlowMist by the information provider till the date of the insurance report (referred to as "provided information"). SlowMist assumes: The information provided is not missing, tampered with, deleted or concealed. If the information provided is missing, tampered with, deleted, concealed, or inconsistent with the actual situation, the SlowMist shall not be liable for any loss or adverse effect resulting therefrom. SlowMist only conducts the agreed security audit on the security situation of the project and issues this report. SlowMist is not responsible for the background and other conditions of the project.



**Official Website**  
[www.slowmist.com](http://www.slowmist.com)



**E-mail**  
[team@slowmist.com](mailto:team@slowmist.com)



**Twitter**  
[@SlowMist\\_Team](https://twitter.com/SlowMist_Team)



**Github**  
<https://github.com/slowmist>