Smart Contract
Security Audit Report

Table Of Contents

1 Executive Summary

On 2022.03.28, the SlowMist security team received the NFTGalaxy team's security audit application for Galaxy, 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/NFTGalaxy/GalaxyContractAudit

commit: cad2e3e6c68cde341d44e53e0c8e8b8ccfe9a4f7

Fixed Version:

https://github.com/NFTGalaxy/GalaxyContractAudit

commit: 776932b3e58743ef3225875e325e6b9be8886032

3.2 Vulnerability Information

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

NO	Title	Category	Level	Status
N1	Risk of excessive authority	Authority Control Vulnerability	Medium	Ignored
N2	Compiler version too low	Integer Overflow and Underflow Vulnerability	Low	Ignored
N3	Lacks a zero-check	Design Logic Audit	Suggestion	Ignored
N4	Reentrancy risks	Reentrancy Vulnerability	Critical	Fixed
N5	Event fraud risk	Malicious Event Log Audit	Low	Fixed
N6	Uses timestamp for compariso	Block data Dependence Vulnerability	Suggestion	Ignored

4 Code Overview

4.1 Contracts Description

The main network address of the contract is as follows:

The code was not deployed to the mainnet.

4.2 Visibility Description

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

MerkleDistributor			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
isClaimed	Public	-	-
_setClaimed	Private	Can Modify State	-
claim	External	Can Modify State	-
withdrawRemaining	External	Can Modify State	onlyOwner

TokenVesting			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
beneficiary	Public	-	-
start	Public	-	-
duration	Public	-	-
vestingCount	Public	-	-

TokenVesting			
revocable	Public	-	-
released	Public	-	-
revoked	Public	-	-
release	Public	Can Modify State	-
revoke	Public	Can Modify State	onlyOwner
releasableAmount	Public	-	-
_vestedAmount	Private	-	-

Ownable			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
owner	Public	-	-
renounceOwnership	Public	Can Modify State	onlyOwner
transferOwnership	Public	Can Modify State	onlyOwner
_setOwner	Private	Can Modify State	-

Context			
Function Name	Visibility	Mutability	Modifiers
_msgSender	Internal	-	-
_msgData	Internal	-	-

Ownable			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Internal	Can Modify State	-
owner	Public	-	-
renounceOwnership	Public	Can Modify State	onlyOwner
transferOwnership	Public	Can Modify State	onlyOwner

GAL			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC20 ERC20Permit
mint	External	Can Modify State	onlyOwner

ERC20			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
name	Public	-	-
symbol	Public	-	-
decimals	Public	-	-
totalSupply	Public	-	-
balanceOf	Public	-	-
transfer	Public	Can Modify State	-
allowance	Public	-	-

ERC20			
approve	Public	Can Modify State	-
transferFrom	Public	Can Modify State	-
increaseAllowance	Public	Can Modify State	-
decreaseAllowance	Public	Can Modify State	-
_transfer	Internal	Can Modify State	-
_mint	Internal	Can Modify State	-
_burn	Internal	Can Modify State	-
_approve	Internal	Can Modify State	-
_beforeTokenTransfer	Internal	Can Modify State	-
_afterTokenTransfer	Internal	Can Modify State	-

ERC20Permit			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	EIP712
permit	Public	Can Modify State	-
nonces	Public	-	-
DOMAIN_SEPARATOR	External	-	-
_useNonce	Internal	Can Modify State	-

SpaceStationV1			
Function Name	Visibility	Mutability	Modifiers

SpaceStationV1			
<Constructor>	Public	Can Modify State	EIP712
activateCampaign	External	Can Modify State	onlyCampaignSetter
claim	External	Payable	onlyNoPaused
claimBatch	External	Payable	onlyNoPaused
forge	External	Payable	onlyNoPaused
<Receive Ether>	External	Payable	-
<Fallback>	External	Payable	-
updateGalaxySigner	External	Can Modify State	onlyManager
updateCampaignSetter	External	Can Modify State	onlyManager
updateManager	External	Can Modify State	onlyManager
updateTreasureManager	External	Can Modify State	onlyTreasuryManager
setPause	External	Can Modify State	onlyManager
_hash	Public	-	-
_hashBatch	Public	-	-
_hashForge	Public	-	-
_verify	Public	-	-
_setFees	Private	Can Modify State	-
_payFees	Private	Can Modify State	-
_validateOnlyCampaignSetter	Internal	-	-
_validateOnlyManager	Internal	-	-

SpaceStationV1			
_validateOnlyTreasuryManager	Internal	-	-
_validateOnlyNotPaused	Internal	-	-

EIP712			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
_domainSeparatorV4	Internal	-	-
_buildDomainSeparator	Private	-	-
_hashTypedDataV4	Internal	-	-

EIP712			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Internal	Can Modify State	-
_domainSeparatorV4	Internal	-	-
_buildDomainSeparator	Private	-	-
_hashTypedDataV4	Internal	-	-
_getChainId	Private	-	-

SpaceStationV1Meta			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	BaseRelayRecipient EIP712 ERC1155Receiver

SpaceStationV1Meta			
updateGalaxySigner	External	Can Modify State	onlyManager
claim	External	Payable	-
claimBatch	External	Payable	-
forge	External	Payable	-
<Receive Ether>	External	Payable	-
<Fallback>	External	Payable	-
_hash	Public	-	-
_hashBatch	Public	-	-
_hashForge	Public	-	-
_verify	Public	-	-
_validateOnlyManager	Internal	-	-
onERC1155Received	External	Can Modify State	-
onERC1155BatchReceived	External	Can Modify State	-
transferInternal	Internal	Can Modify State	-
batchTransferInternal	Internal	Can Modify State	-

BaseRelayRecipient			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-

BaseRelayRecipient			
isTrustedForwarder	Public	-	-
_msgSender	Internal	-	-
_msgData	Internal	-	-

MinimalForwarder			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	EIP712
getNonce	Public	-	-
verify	Public	-	-
execute	Public	Payable	-

ERC1155Receiver			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Internal	Can Modify State	-

ERC165			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Internal	Can Modify State	-
supportsInterface	Public	-	-
_registerInterface	Internal	Can Modify State	-

SpaceStationV2

SpaceStationV2			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	EIP712
activateCampaign	External	Can Modify State	onlyCampaignSetter
claim	Public	Payable	onlyNoPaused
claim	External	Payable	onlyNoPaused
claimBatch	Public	Payable	onlyNoPaused
claimBatch	External	Payable	onlyNoPaused
claimCapped	Public	Payable	onlyNoPaused
claimCapped	External	Payable	onlyNoPaused
claimBatchCapped	Public	Payable	onlyNoPaused
claimBatchCapped	External	Payable	onlyNoPaused
forge	Public	Payable	onlyNoPaused
forge	External	Payable	onlyNoPaused
<Receive Ether>	External	Payable	-
<Fallback>	External	Payable	-
updateGalaxySigner	External	Can Modify State	onlyManager
updateCampaignSetter	External	Can Modify State	onlyManager
updateManager	External	Can Modify State	onlyManager
updateTreasureManager	External	Can Modify State	onlyTreasuryManager
setPause	External	Can Modify State	onlyManager

SpaceStationV2			
_hash	Public	-	-
_hashCapped	Public	-	-
_hashBatch	Public	-	-
_hashBatchCapped	Public	-	-
_hashForge	Public	-	-
_verify	Public	-	-
_setFees	Private	Can Modify State	-
_payFees	Private	Can Modify State	-
_validateOnlyCampaignSetter	Internal	-	-
_validateOnlyManager	Internal	-	-
_validateOnlyTreasuryManager	Internal	-	-
_validateOnlyNotPaused	Internal	-	-

SpaceStationV2Meta			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	BaseRelayRecipient
claim	External	Payable	-
claimBatch	External	Payable	-
claimCapped	External	Payable	-
claimBatchCapped	External	Payable	-

SpaceStationV2Meta			
forge	External	Payable	-
claim	External	Payable	-
claimBatch	External	Payable	-
claimCapped	External	Payable	-
claimBatchCapped	External	Payable	-
forge	External	Payable	-
<Receive Ether>	External	Payable	-
<Fallback>	External	Payable	-

EIP712			
Function Name	Visibility	Mutability	Modifiers
initialize	Internal	Can Modify State	-
_domainSeparatorV4	Internal	-	-
_buildDomainSeparator	Private	-	-
_hashTypedDataV4	Internal	-	-
_getChainId	Private	-	-

StarNFT721NaiveFactory			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
allNFTsLength	External	-	-

StarNFT721NaiveFactory			
getOneNFT	External	-	-
getRecentNFT	External	-	-
createStarNFT	External	Can Modify State	onlyNoPaused
<Receive Ether>	External	Payable	-
setPause	External	Can Modify State	onlyManager
_validateOnlyManager	Internal	-	-
_validateOnlyNotPaused	Internal	-	-
toAsciiString	Internal	-	-
char	Internal	-	-

StarNFT721			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC721
transferFrom	Public	Can Modify State	-
safeTransferFrom	Public	Can Modify State	-
safeTransferFrom	Public	Can Modify State	-
name	Public	-	-
symbol	Public	-	-
mint	External	Can Modify State	onlyMinter
mintBatch	External	Can Modify State	onlyMinter

StarNFT721			
burn	External	Can Modify State	onlyMinter
burnBatch	External	Can Modify State	onlyMinter
isOwnerOf	Public	-	-
getNumMinted	External	-	-
tokenURI	Public	-	-
setURI	External	Can Modify State	onlyOwner
setTransferable	External	Can Modify State	onlyOwner
setName	External	Can Modify State	onlyOwner
setSymbol	External	Can Modify State	onlyOwner
addMinter	External	Can Modify State	onlyOwner
removeMinter	External	Can Modify State	onlyOwner
uint2str	Internal	-	-

ERC721			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
balanceOf	Public	-	-
ownerOf	Public	-	-
name	Public	-	-
symbol	Public	-	-

ERC721			
tokenURI	Public	-	-
baseURI	Public	-	-
tokenOfOwnerByIndex	Public	-	-
totalSupply	Public	-	-
tokenByIndex	Public	-	-
approve	Public	Can Modify State	-
getApproved	Public	-	-
setApprovalForAll	Public	Can Modify State	-
isApprovedForAll	Public	-	-
transferFrom	Public	Can Modify State	-
safeTransferFrom	Public	Can Modify State	-
safeTransferFrom	Public	Can Modify State	-
_safeTransfer	Internal	Can Modify State	-
_exists	Internal	-	-
_isApprovedOrOwner	Internal	-	-
_safeMint	Internal	Can Modify State	-
_safeMint	Internal	Can Modify State	-
_mint	Internal	Can Modify State	-
_burn	Internal	Can Modify State	-
_transfer	Internal	Can Modify State	-

ERC721			
_setTokenURI	Internal	Can Modify State	-
_setBaseURI	Internal	Can Modify State	-
_checkOnERC721Received	Private	Can Modify State	-
_approve	Private	Can Modify State	-
_beforeTokenTransfer	Internal	Can Modify State	-

StarNFTV3NaiveFactory			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	-
allNFTsLength	External	-	-
getOneNFT	External	-	-
getRecentNFT	External	-	-
createStarNFT	External	Can Modify State	onlyNoPaused
<Receive Ether>	External	Payable	-
setPause	External	Can Modify State	onlyManager
_validateOnlyManager	Internal	-	-
_validateOnlyNotPaused	Internal	-	-
toAsciiString	Internal	-	-
char	Internal	-	-

StarNFTV3

StarNFTV3			
Function Name	Visibility	Mutability	Modifiers
<Constructor>	Public	Can Modify State	ERC721
transferFrom	Public	Can Modify State	-
safeTransferFrom	Public	Can Modify State	-
safeTransferFrom	Public	Can Modify State	-
name	Public	-	-
symbol	Public	-	-
cid	Public	-	-
mint	External	Can Modify State	onlyMinter
mintBatch	External	Can Modify State	onlyMinter
burn	External	Can Modify State	onlyMinter
burnBatch	External	Can Modify State	onlyMinter
isOwnerOf	Public	-	-
getNumMinted	External	-	-
tokenURI	Public	-	-
setURI	External	Can Modify State	onlyOwner
setTransferable	External	Can Modify State	onlyOwner
setName	External	Can Modify State	onlyOwner
setSymbol	External	Can Modify State	onlyOwner
addMinter	External	Can Modify State	onlyOwner

StarNFTV3			
removeMinter	External	Can Modify State	onlyOwner
uint2str	Internal	-	-

4.3 Vulnerability Summary

[N1] [Medium] Risk of excessive authority

Category: Authority Control Vulnerability

Content

```
modifier onlyOwner() {
    require(owner() == _msgSender(), "Ownable: caller is not the owner");
    _;
}
modifier onlyMinter() {
    require(minters[msg.sender], "must be minter");
    _;
}
modifier onlyManager() {
    _validateOnlyManager();
    _;
}
```

`Owner/Manager/Minter` have special authority in the contracts.

Solution

It is recommended to set `Owner/Manager/Minter` address to timelock contract, governance contract, or multi-sign contract to reduce the risk of private key loss.

Status

Ignored

[N2] [Low] Compiler version too low

Category: Integer Overflow and Underflow Vulnerability

Content

```
pragma solidity 0.7.6;
```

Compiler version 0.7.6 is too low, there is a risk of numerical overflow.

Solution

Upgrade the compiler to above 0.8.0

Status

Ignored

[N3] [Suggestion] Lacks a zero-check

Category: Design Logic Audit

Content

- GalaxyContractAudit-main/contracts/Distributor/MerkleDistributor.sol

```
constructor(address token_, bytes32 merkleRoot_, uint256 endBlock_) public {
    token = token_; //SlowMist// lacks a zero-check
    merkleRoot = merkleRoot_;
    endBlock = endBlock_;
}
```

Solution

Check variable != address(0)

Status

Ignored

[N4] [Critical] Reentrancy risks

Category: Reentrancy Vulnerability

Content

- GalaxyContractAudit-main/contracts/SpaceStation/SpaceStationV2.sol

```
function claimCapped(
    uint256 _cid,
    IStarNFT _starNFT,
    uint256 _dummyId,
    uint256 _powah,
    uint256 _cap,
    address _mintTo,
    bytes calldata _signature
) public payable onlyNoPaused {
    require(!hasMinted[_dummyId], "Already minted");
    require(numMinted[_cid] < _cap, "Reached cap limit");
    require(
        _verify(
            _hashCapped(_cid, _starNFT, _dummyId, _powah, _cap, _mintTo),
            _signature
        ),
        "Invalid signature"
    );
    hasMinted[_dummyId] = true;
    _payFees(_cid, 1);
    uint256 nftID = _starNFT.mint(_mintTo, _powah);
    numMinted[_cid] = numMinted[_cid] + 1; //SlowMist// Change state after call
    emit EventClaim(_cid, _dummyId, nftID, _mintTo);
}

function claimBatchCapped(
    uint256 _cid,
    IStarNFT _starNFT,
    uint256[] calldata _dummyIdArr,
    uint256[] calldata _powahArr,
    uint256 _cap,
    address _mintTo,
    bytes calldata _signature
) public payable onlyNoPaused {

//...
```

```
    _payFees(_cid, _dummyIdArr.length);
    uint256[] memory nftIdArr = _starNFT.mintBatch(
        _mintTo,
        _powahArr.length,
        _powahArr
    );
    numMinted[_cid] = numMinted[_cid] + _dummyIdArr.length; //SlowMist// Change state
after call
    emit EventClaimBatch(_cid, _dummyIdArr, nftIdArr, _mintTo);
}

function forge(
    uint256 _cid,
    IStarNFT _starNFT,
    uint256[] calldata _nftIDs,
    uint256 _dummyId,
    uint256 _powah,
    address _mintTo,
    bytes calldata _signature
) public payable onlyNoPaused {
    require(!hasMinted[_dummyId], "Already minted");
    require(
        _verify(
            _hashForge(
                _cid,
                _starNFT,
                _nftIDs,
                _dummyId,
                _powah,
                _mintTo
            ),
            _signature
        ),
        "Invalid signature"
    );
    hasMinted[_dummyId] = true;
    for (uint256 i = 0; i < _nftIDs.length; i++) {
        require(
            _starNFT.isOwnerOf(_mintTo, _nftIDs[i]),
            "Not the owner"
        );
    }
    _payFees(_cid, 1);
    _starNFT.burnBatch(_mintTo, _nftIDs); //SlowMist// Change state after call
```

```
    uint256 nftID = _starNFT.mint(_mintTo, _powah);
    emit EventForge(_cid, _dummyId, nftID, _mintTo);
}
```

External calls sending eth, and state variables written after the calls.

Solution

Write state variables before calls.

Status

Fixed; Fixed PR:

https://github.com/NFTGalaxy/GalaxyContractAudit/commit/d1de1209a8999ad08c4f781fee5cedb302751d5e

[N5] [Low] Event fraud risk

Category: Malicious Event Log Audit

Content

- GalaxyContractAudit-main/contracts/SpaceStation/SpaceStationV2.sol

```
emit EventClaim(_cid, _dummyId, nftID, _mintTo);
```

Event `EventClaim` is the same in `claim` and `claimBatchCapped` functions.

- GalaxyContractAudit-main/contracts/SpaceStation/SpaceStationV2.sol

```
emit EventClaimBatch(_cid, _dummyIdArr, nftIdArr, _mintTo);
```

Event `EventClaimBatch` is the same in `claimBatch` and `claimBatchCapped` functions.

The client may not be able to tell which function was called

Solution

Don't use the same event in different functions.

Status

Fixed; Fixed PR:

https://github.com/NFTGalaxy/GalaxyContractAudit/commit/776932b3e58743ef3225875e325e6b9be8886032

[N6] [Suggestion] Uses timestamp for compariso

Category: Block data Dependence Vulnerability

Content

- GalaxyContractAudit-main/contracts/GAL/TokenVesting.sol

```
function _vestedAmount(IERC20 token) private view returns (uint256) {
    uint256 currentBalance = token.balanceOf(address(this));
    uint256 totalBalance = currentBalance.add(_released[address(token)]);

    if (block.timestamp < _start) {
        // not start
        return 0;
    } else if (block.timestamp >= _start.add(_duration.mul(_vestingCount)) ||
_revoked[address(token)]) {
        // end or revoke
        return totalBalance;
    } else {
        // ((now - start) / duration / vestingCount) * totalBalance
        return totalBalance.mul(
            block.timestamp.sub(_start).div(_duration).mul(10000).div(_vestingCount)
        ).div(10000);
    }
}
```

Dangerous usage of block.timestamp, block.timestamp can be manipulated by miners.

Solution

Avoid relying on block.timestamp.

Status

Ignored

5 Audit Result

Audit Number	Audit Team	Audit Date	Audit Result
0X002204080001	SlowMist Security Team	2022.03.28 - 2022.04.08	Medium 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 1 critical risk, 2 medium risk, 1 low risk, 2 suggestion vulnerabilities. The code was not 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

E-mail

team@slowmist.com

Twitter

@SlowMist_Team

Github

https://github.com/slowmist