Nitro Collection Smart Contract final Audit Report

Overview

Scope of Audit

The scope of this audit was to analyse and document the **Nitro Collection** smart contract codebase for quality, security, and correctness.

Check Vulnerabilities

- Re-entrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Use of tx.origin
- Exception disorder
- Gasless send
- Balance equality
- Byte array
- Transfer forwards all gas
- ERC20 API violation
- Malicious libraries
- Compiler version not fixed
- Redundant fallback function
- Send instead of transfer
- Style guide violation
- Unchecked external call
- Unchecked maths
- Unsafe type inference
- Implicit visibility level

Techniques and Methods

Throughout the audit of smart contracts, care was taken to ensure:

- The overall quality of code.
- Use of best practises.
- Code documentation and comments match logic and expected behaviour.
- Token distribution and calculations are as per the intended behaviour mentioned in the whitepaper.
- Implementation of ERC-20 token standards.
- Efficient use of gas.
- Code is safe from re-entrancy and other vulnerabilities.

The following techniques, methods, and tools were used to review all the smart contracts.

Structural Analysis

In this step, we have analysed the design patterns and structure of smart contracts. A thorough check was done to ensure the smart contract is structured in a way that will not result in future problems.

Static Analysis

A static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step, a series of automated tools are used to test the security of smart contracts.

Code Review / Manual Analysis

Manual Analysis or review of code was done to identify new vulnerabilities or verify the vulnerabilities found during the static analysis. Contracts were completely manually analysed, their logic was checked and compared with the one described in the whitepaper. Besides, the results of the automated analysis were manually verified.

Gas Consumption

In this step, we have checked the behaviour of smart contracts in production. Checks were done to know how much gas gets consumed and the possibilities of optimization of code to reduce gas consumption.

Tools and Platforms used for Audit

Remix IDE, Truffle, Truffle Team, Solhint, Mythril, Slither, Solidity statistic analysis.

Issue Categories

Every issue in this report has been assigned to a severity level. There are four levels of severity, and each of them has been explained below.

High Severity Issues

A high severity issue or vulnerability means that your smart contract can be exploited. Issues on this level are critical to the smart contract's performance or functionality, and we recommend these issues be fixed before moving to a live environment.

Medium Severity Issues

The issues marked as medium severity usually arise because of errors and deficiencies in the smart contract code. Issues on this level could potentially bring problems, and they should still be fixed.

Low Severity Issues

Low-level severity issues can cause minor impact and or are just warnings that can remain unfixed for now. It would be better to fix these issues at some point in the future.

Informational Issues

These are four severity issues that indicate an improvement request, a general question, a cosmetic or documentation error, or a request for information. There is low-to-no impact.

Number of security issues per severity.

TYPE	HIGH	MEDIUM	LOW	INFORMATIONAL
Open	0	0	0	0
Acknowledged	0	0	2	1
Closed	0	0	0	1

Introduction

During the period of **Feb 7, 2022 to Feb 12, 2022** - QuillAudits Team performed a security audit for **Nitro Collection** smart contracts.

The code for the audit was taken from following the official link**:**

Codebase: b75498b80d8e07da636f52858c793690e5cd5b78

Issues Found – Code Review / Manual Testing

High Severity Issues

none

Medium Severity Issues

none

Low Severity Issues

- **[L1] redundant code in tokenURI**

  The view function ***tokenURI*** has redundancy code and extra use of storage and gas.

  We suggest to make the tokenURI function as:

```
function tokenURI(uint256 tokenId)
    public
    view
    virtual
    override
    returns (string memory)
{
    require(_exists(tokenId), "tokenURI: URI query for nonexistent token");
    string memory _tokenURI = _tokenURIs[tokenId];
    string memory base = _baseURI();
    if (bytes(base).length == 0) {
        return _tokenURI;
    }
    return string(abi.encodePacked(base, tokenId.toString()));
}
```

  Status: **Acknowledged**

- **[L2] tier variable is not used anywhere**

  tier variable is assigned but not used anywhere in the contract.

  We suggest removing the tier variable from the contract. It's just a waste of storage and gas.

  Status: **Acknowledged**

Informational Issues

- **[INF1] Missing comments and description:**

  Comments and Description of the methods and the variables are missing, it's hard to read and understand the purpose of the variables and the methods in context of the whole picture

  Recommendation: Consider adding NatSpec format comments for the comments and state variables

  Status: **Acknowledged**

- **[INF2] Public methods only being used externally**

  'public' functions that are never used within the contract should be declared 'external' to save gas.

  Recommendation: Make these methods external

  updateERC20, updateOperator, updatePrice, superMint, supportsInterface, setBaseURI, tokenURI, tokenExists.

  Status: **Closed**

Kovan Testnet Test Contract

Contract : 0x85503ae1869dd51cbd6be4e5f6a0061ccc1b004f

Functional Tests

- updateERC20
  - Successfully update the token state PASS
  - Only Owner can update the tokens PASS
- updateOperator
  - SuccessFully update the operator PASS
  - Only Owner can update the operator PASS
- updatePrice
  - SuccessFully update the price of token PASS
  - Only Owner can update the price of token PASS
- updateBeneficiary
  - Update Beneficiary successfully PASS
  - only owner can update the beneficiary PASS
- updateTeller
  - successfully update the teller PASS
  - only owner can update the teller PASS
- pay
  - pay ERC20 price and mint ERC721 token PASS
  - transfer not allowed for insufficient allowance of ERC20 (DAI) PASS
- superMint
  - successfully superMint the token PASS
  - Only operator can super mint PASS
- setBaseURI
  - successfully set the URI PASS
  - Only owner can set the URI PASS
- payPending
  - successfully transfer the ERC20 tokens PASS
  - balance should be greater than the pay amount PASS
- transferOwnership
  - SuccessFully transfer the ownership PASS
  - Only Owner can transfer the ownership PASS

Automated Tests

Slither:

```
ERC721._checkOnERC721Received(address,address,uint256,bytes) (NitroCollection_flat.sol#1281-1302) ignores return value by IERC721Receiver(to).onERC721Received
(_msgSender(),from,tokenId,_data) (NitroCollection_flat.sol#1288-1298)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-return

NitroCollection.updateERC20(address,bool)._status (NitroCollection_flat.sol#1633) shadows:
        - ReentrancyGuard._status (NitroCollection_flat.sol#86) (state variable)
NitroCollection.updateOperator(address,bool)._status (NitroCollection_flat.sol#1639) shadows:
        - ReentrancyGuard._status (NitroCollection_flat.sol#86) (state variable)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).retval (NitroCollection_flat.sol#1288)' in ERC721._checkOnERC721Received(address,addres
s,uint256,bytes) (NitroCollection_flat.sol#1281-1302) potentially used before declaration: retval == IERC721Receiver.onERC721Received.selector (NitroCollectio
n_flat.sol#1289)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (NitroCollection_flat.sol#1290)' in ERC721._checkOnERC721Received(address,addres
s,uint256,bytes) (NitroCollection_flat.sol#1281-1302) potentially used before declaration: reason.length == 0 (NitroCollection_flat.sol#1291)
Variable 'ERC721._checkOnERC721Received(address,address,uint256,bytes).reason (NitroCollection_flat.sol#1290)' in ERC721._checkOnERC721Received(address,addres
s,uint256,bytes) (NitroCollection_flat.sol#1281-1302) potentially used before declaration: revert(uint256,uint256)(32 + reason,mload(uint256)(reason)) (NitroC
ollection_flat.sol#1295)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#pre-declaration-usage-of-local-variables

Reentrancy in NitroCollection.pay(address) (NitroCollection_flat.sol#1655-1676):
        External calls:
        - require(bool,string)(transferERC20(_erc20,beneficiary,_toPay),Pay:: Transfer Failed) (NitroCollection_flat.sol#1663-1666)
                - IERC20(_erc20).transferFrom(msg.sender,_recipient,_toPay) (NitroCollection_flat.sol#1694)
        - safeMint(msg.sender) (NitroCollection_flat.sol#1667)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (NitroCollection_flat.sol#1288-1298)
        Event emitted after the call(s):
        - Paid(_erc20,_toPay,msg.sender,beneficiary,block.timestamp,tier) (NitroCollection_flat.sol#1668-1675)
        - Transfer(address(0),to,tokenId) (NitroCollection_flat.sol#1182)
                - safeMint(msg.sender) (NitroCollection_flat.sol#1667)
Reentrancy in NitroCollection.payPending(address,uint256) (NitroCollection_flat.sol#1697-1714):
        External calls:
        - require(bool,string)(transferERC20(_erc20,teller,_toPay),PayPending:: Transfer Failed) (NitroCollection_flat.sol#1702-1705)
                - IERC20(_erc20).transferFrom(msg.sender,_recipient,_toPay) (NitroCollection_flat.sol#1694)
        Event emitted after the call(s):
        - PendingPaid(_erc20,_toPay,msg.sender,teller,block.timestamp,tier) (NitroCollection_flat.sol#1706-1713)
Reentrancy in NitroCollection.superMint(address) (NitroCollection_flat.sol#1683-1687):
        External calls:
        - safeMint(_to) (NitroCollection_flat.sol#1685)
                - IERC721Receiver(to).onERC721Received(_msgSender(),from,tokenId,_data) (NitroCollection_flat.sol#1288-1298)
        Event emitted after the call(s):
        - OperatorMinted(msg.sender,_to,block.timestamp) (NitroCollection_flat.sol#1686)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

Address.verifyCallResult(bool,bytes,string) (NitroCollection_flat.sol#576-596) uses assembly
        - INLINE ASM (NitroCollection_flat.sol#588-591)
ERC721._checkOnERC721Received(address,address,uint256,bytes) (NitroCollection_flat.sol#1281-1302) uses assembly
        - INLINE ASM (NitroCollection_flat.sol#1294-1296)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage
```

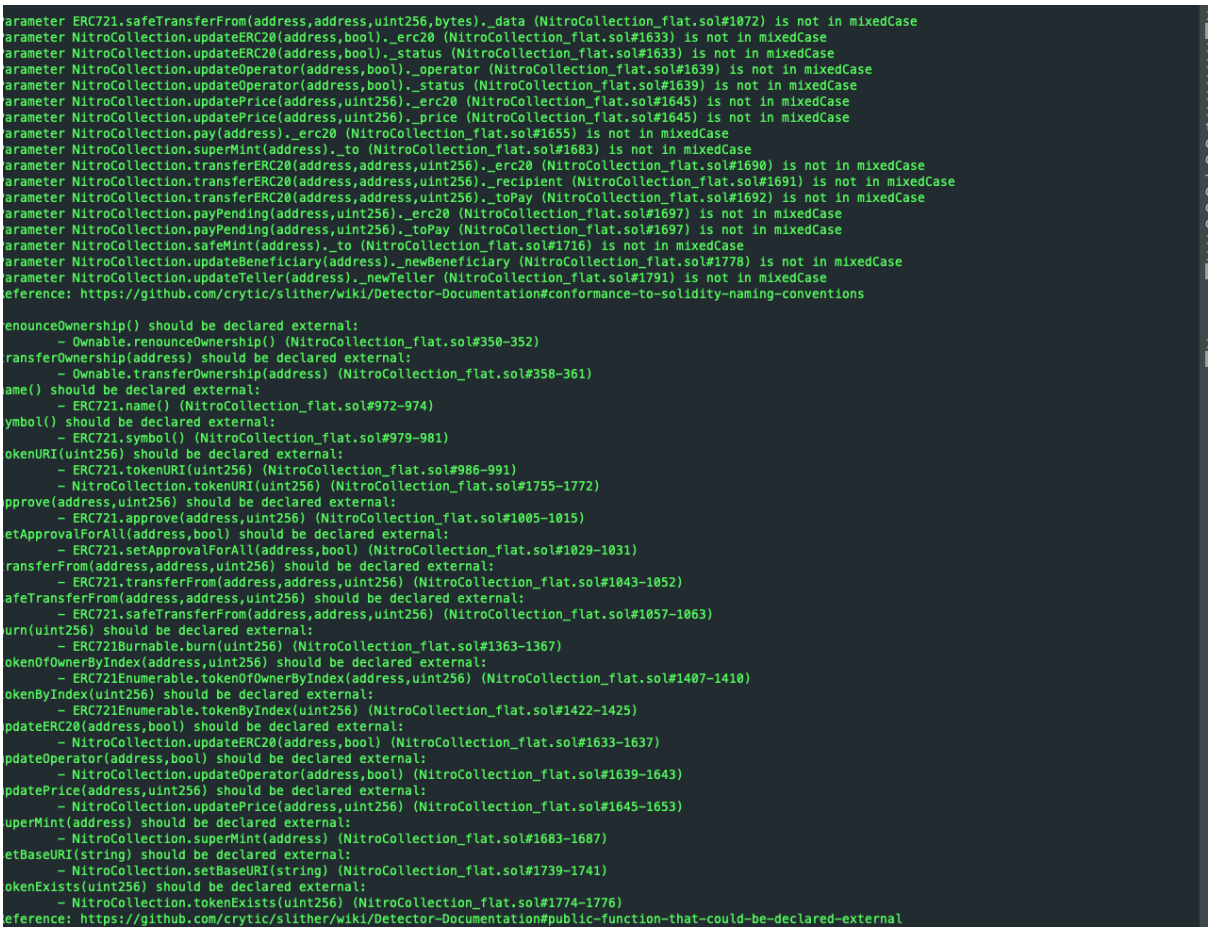
```
Different versions of Solidity is used:
        - Version used: ['0.8.9', '^0.8.0', '^0.8.1']
        - 0.8.9 (NitroCollection_flat.sol#7)
        - ^0.8.0 (NitroCollection_flat.sol#53)
        - ^0.8.0 (NitroCollection_flat.sol#119)
        - ^0.8.0 (NitroCollection_flat.sol#204)
        - ^0.8.0 (NitroCollection_flat.sol#274)
        - ^0.8.0 (NitroCollection_flat.sol#301)
        - ^0.8.1 (NitroCollection_flat.sol#379)
        - ^0.8.0 (NitroCollection_flat.sol#604)
        - ^0.8.0 (NitroCollection_flat.sol#634)
        - ^0.8.0 (NitroCollection_flat.sol#662)
        - ^0.8.0 (NitroCollection_flat.sol#693)
        - ^0.8.0 (NitroCollection_flat.sol#838)
        - ^0.8.0 (NitroCollection_flat.sol#869)
        - ^0.8.0 (NitroCollection_flat.sol#898)
        - ^0.8.0 (NitroCollection_flat.sol#1347)
        - ^0.8.0 (NitroCollection_flat.sol#1375)
        - 0.8.9 (NitroCollection_flat.sol#1538)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#different-pragma-directives-are-used

Address.functionCall(address,bytes) (NitroCollection_flat.sol#460-462) is never used and should be removed
Address.functionCall(address,bytes,string) (NitroCollection_flat.sol#470-476) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (NitroCollection_flat.sol#489-495) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (NitroCollection_flat.sol#503-514) is never used and should be removed
Address.functionDelegateCall(address,bytes) (NitroCollection_flat.sol#549-551) is never used and should be removed
Address.functionDelegateCall(address,bytes,string) (NitroCollection_flat.sol#559-568) is never used and should be removed
Address.functionStaticCall(address,bytes) (NitroCollection_flat.sol#522-524) is never used and should be removed
Address.functionStaticCall(address,bytes,string) (NitroCollection_flat.sol#532-541) is never used and should be removed
Address.sendValue(address,uint256) (NitroCollection_flat.sol#435-440) is never used and should be removed
Address.verifyCallResult(bool,bytes,string) (NitroCollection_flat.sol#576-596) is never used and should be removed
Context._msgData() (NitroCollection_flat.sol#291-293) is never used and should be removed
Counters.decrement(Counters.Counter) (NitroCollection_flat.sol#35-41) is never used and should be removed
Counters.reset(Counters.Counter) (NitroCollection_flat.sol#43-45) is never used and should be removed
ERC721._baseURI() (NitroCollection_flat.sol#998-1000) is never used and should be removed
NitroCollection._setTokenURI(uint256,string) (NitroCollection_flat.sol#1743-1749) is never used and should be removed
Strings.toHexString(uint256) (NitroCollection_flat.sol#240-251) is never used and should be removed
Strings.toHexString(uint256,uint256) (NitroCollection_flat.sol#256-266) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

Pragma version0.8.9 (NitroCollection_flat.sol#7) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#53) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#119) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#204) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#274) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#301) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.1 (NitroCollection_flat.sol#379) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#604) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#634) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#662) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#693) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#838) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#869) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#898) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#1347) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version^0.8.0 (NitroCollection_flat.sol#1375) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
Pragma version0.8.9 (NitroCollection_flat.sol#1538) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.9 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (NitroCollection_flat.sol#435-440):
        - (success) = recipient.call{value: amount}() (NitroCollection_flat.sol#438)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (NitroCollection_flat.sol#503-514):
        - (success,returndata) = target.call{value: value}(data) (NitroCollection_flat.sol#512)
Low level call in Address.functionStaticCall(address,bytes,string) (NitroCollection_flat.sol#532-541):
        - (success,returndata) = target.staticcall(data) (NitroCollection_flat.sol#539)
Low level call in Address.functionDelegateCall(address,bytes,string) (NitroCollection_flat.sol#559-568):
        - (success,returndata) = target.delegatecall(data) (NitroCollection_flat.sol#566)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Parameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (NitroCollection_flat.sol#1072) is not in mixedCase
Parameter NitroCollection.updateERC20(address,bool)._erc20 (NitroCollection_flat.sol#1633) is not in mixedCase
Parameter NitroCollection.updateERC20(address,bool)._status (NitroCollection_flat.sol#1633) is not in mixedCase
Parameter NitroCollection.updateOperator(address,bool)._operator (NitroCollection_flat.sol#1639) is not in mixedCase
Parameter NitroCollection.updateOperator(address,bool)._status (NitroCollection_flat.sol#1639) is not in mixedCase
Parameter NitroCollection.updatePrice(address,uint256)._erc20 (NitroCollection_flat.sol#1645) is not in mixedCase
```



```
arameter ERC721.safeTransferFrom(address,address,uint256,bytes)._data (NitroCollection_flat.sol#1072) is not in mixedCase
arameter NitroCollection.updateERC20(address,bool)._erc20 (NitroCollection_flat.sol#1633) is not in mixedCase
arameter NitroCollection.updateERC20(address,bool)._status (NitroCollection_flat.sol#1633) is not in mixedCase
arameter NitroCollection.updateOperator(address,bool)._operator (NitroCollection_flat.sol#1639) is not in mixedCase
arameter NitroCollection.updateOperator(address,bool)._status (NitroCollection_flat.sol#1639) is not in mixedCase
arameter NitroCollection.updatePrice(address,uint256)._erc20 (NitroCollection_flat.sol#1645) is not in mixedCase
arameter NitroCollection.updatePrice(address,uint256)._price (NitroCollection_flat.sol#1645) is not in mixedCase
arameter NitroCollection.pay(address)._erc20 (NitroCollection_flat.sol#1655) is not in mixedCase
arameter NitroCollection.superMint(address)._to (NitroCollection_flat.sol#1683) is not in mixedCase
arameter NitroCollection.transferERC20(address,address,uint256)._erc20 (NitroCollection_flat.sol#1690) is not in mixedCase
arameter NitroCollection.transferERC20(address,address,uint256)._recipient (NitroCollection_flat.sol#1691) is not in mixedCase
arameter NitroCollection.transferERC20(address,address,uint256)._toPay (NitroCollection_flat.sol#1692) is not in mixedCase
arameter NitroCollection.payPending(address,uint256)._erc20 (NitroCollection_flat.sol#1697) is not in mixedCase
arameter NitroCollection.payPending(address,uint256)._toPay (NitroCollection_flat.sol#1697) is not in mixedCase
arameter NitroCollection.safeMint(address)._to (NitroCollection_flat.sol#1716) is not in mixedCase
arameter NitroCollection.updateBeneficiary(address)._newBeneficiary (NitroCollection_flat.sol#1778) is not in mixedCase
arameter NitroCollection.updateTeller(address)._newTeller (NitroCollection_flat.sol#1791) is not in mixedCase
eference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

enounceOwnership() should be declared external:
        - Ownable.renounceOwnership() (NitroCollection_flat.sol#350-352)
ransferOwnership(address) should be declared external:
        - Ownable.transferOwnership(address) (NitroCollection_flat.sol#358-361)
ame() should be declared external:
        - ERC721.name() (NitroCollection_flat.sol#972-974)
ymbol() should be declared external:
        - ERC721.symbol() (NitroCollection_flat.sol#979-981)
okenURI(uint256) should be declared external:
        - ERC721.tokenURI(uint256) (NitroCollection_flat.sol#986-991)
        - NitroCollection.tokenURI(uint256) (NitroCollection_flat.sol#1755-1772)
pprove(address,uint256) should be declared external:
        - ERC721.approve(address,uint256) (NitroCollection_flat.sol#1005-1015)
etApprovalForAll(address,bool) should be declared external:
        - ERC721.setApprovalForAll(address,bool) (NitroCollection_flat.sol#1029-1031)
ransferFrom(address,address,uint256) should be declared external:
        - ERC721.transferFrom(address,address,uint256) (NitroCollection_flat.sol#1043-1052)
afeTransferFrom(address,address,uint256) should be declared external:
        - ERC721.safeTransferFrom(address,address,uint256) (NitroCollection_flat.sol#1057-1063)
urn(uint256) should be declared external:
        - ERC721Burnable.burn(uint256) (NitroCollection_flat.sol#1363-1367)
okenOfOwnerByIndex(address,uint256) should be declared external:
        - ERC721Enumerable.tokenOfOwnerByIndex(address,uint256) (NitroCollection_flat.sol#1407-1410)
okenByIndex(uint256) should be declared external:
        - ERC721Enumerable.tokenByIndex(uint256) (NitroCollection_flat.sol#1422-1425)
pdateERC20(address,bool) should be declared external:
        - NitroCollection.updateERC20(address,bool) (NitroCollection_flat.sol#1633-1637)
pdateOperator(address,bool) should be declared external:
        - NitroCollection.updateOperator(address,bool) (NitroCollection_flat.sol#1639-1643)
pdatePrice(address,uint256) should be declared external:
        - NitroCollection.updatePrice(address,uint256) (NitroCollection_flat.sol#1645-1653)
uperMint(address) should be declared external:
        - NitroCollection.superMint(address) (NitroCollection_flat.sol#1683-1687)
etBaseURI(string) should be declared external:
        - NitroCollection.setBaseURI(string) (NitroCollection_flat.sol#1739-1741)
okenExists(uint256) should be declared external:
        - NitroCollection.tokenExists(uint256) (NitroCollection_flat.sol#1774-1776)
eference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Results:

No major issues were found. Some false positive errors were reported by the tool. All the other issues have been categorised above according to their level of severity.

Closing Summary

Contracts are well written and most of the low severity and informational issues are acknowledged by the team.

The contracts are good to get deployed on public smart chains.

Disclaimer

Quillhash audit is not a security warranty, investment advice, or endorsement of the **Nitro Network platform**. This audit does not provide a security or correctness guarantee of the audited smart contracts. The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them. Securing smart contracts is a multistep process. One audit cannot be considered enough. We recommend that the **Nitro Network** Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.