verichains

SECURITY AUDIT OF

OP3N NEIGHBORHEADZ NFT SMART CONTRACTS

Public Report

Apr 09, 2022

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology > Forward

ABBREVIATIONS

Name	Description
Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
Ether (ETH)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
ERC20	ERC20 (BEP20 in Binance Smart Chain or *x*RP20 in other chains) tokens are blockchain-based assets that have value and can be sent and received. The primary difference with the primary coin is that instead of running on their own blockchain, ERC20 tokens are issued on a network that supports smart contracts such as Ethereum or Binance Smart Chain.

EXECUTIVE SUMMARY

This Security Audit Report prepared by Verichains Lab on Apr 09, 2022. We would like to thank the OP3N for trusting Verichains Lab in auditing smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the OP3N neighborheadz NFT Smart Contracts. The scope of the audit is limited to the source code files provided to Verichains. Verichains Lab completed the assessment using manual, static, and dynamic analysis techniques.

During the audit process, the audit team had identified no vulnerable issues in the smart contracts code.

TABLE OF CONTENTS

1. MANAGEMENT SUMMARY **5**

1.1. About OP3N neighborheadz NFT Smart Contracts **5**

1.2. Audit scope **5**

1.3. Audit methodology **5**

1.4. Disclaimer **6**

2. AUDIT RESULT **7**

2.1. Overview **7**

2.2. Findings **7**

2.3. Additional notes and recommendations **7**

2.3.1. NeighborheadzNFT.sol - Unused _ReentrancyGuard INFORMATIVE 7

3. VERSION HISTORY **10**

1. MANAGEMENT SUMMARY

1.1. About OP3N neighborheadz NFT Smart Contracts

OP3N is THE platform for creators & fans. With OP3N, musicians, filmmakers, gamers, and artists can host content and metaverse experiences through NFT Memberships.

1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the OP3N neighborheadz NFT Smart Contracts.

It was conducted on commit 3876c81e782d62c3e27e0fb2f3eff2be48f7447c from git repository *https://github.com/EST-Media/op3n-neighborheadz-nft*.

The latest version of the following files were made available in the course of the review:

SHA256 Sum	File
1bddbfbd8d2c0e633b65eb4d8c5c5cdeff2e716410bf5c4c817a3534e5a3b2d2	**NeighborheadzNFT.sol**

1.3. Audit methodology

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and RK87, our in-house smart contract security analysis tool.
- 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:

- Integer Overflow and Underflow
- Timestamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- DoS with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference
- Reentrancy

- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories as listed in table below, depending on their severity level:

SEVERITY LEVEL	DESCRIPTION
CRITICAL	A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.
HIGH	A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.
MEDIUM	A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.
LOW	An issue that does not have a significant impact, can be considered as less important.

Table 1. Severity levels

1.4. Disclaimer

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

2. AUDIT RESULT

2.1. Overview

The OP3N neighborheadz NFT Smart Contracts was written in Solidity language, with the required version to be ^0.8.12. The source code was written based on OpenZeppelin's library.

The OP3N neighborheadz NFT Smart Contracts is the ERC721 NFT contract which extends AccessControl, ERC721 and ERC721Royalty contract. AccessControl allows the contract to implement role-based access control mechanisms which adds token owner (contract deployer) DEFAULT_ADMIN_ROLE role, he then can set any roles for anyone at any time. The contract also sets the contract deployer to be owner and the owner can transfer his ownership to a new account at any time. By extending ERC721Royalty, the contract supports EIP-2981: NFT Royalty Standard, providing a standardized way to retrieve royalty payment information for non-fungible tokens (NFTs) to enable universal support for royalty payments across all NFT marketplaces and ecosystem participants.

Users need to have both valid proof and sig and must pay 0.08 native tokens (ETH, BNB, ...) to mint a NFT, each user can mint up to 2 NFT. The proof is used to verify that the caller is a part of the Merkle tree defined by _preSaleRoot (whitelisted by DEFAULT_ADMIN_ROLE). The sig will be generated by _verifiers (backend servers). Total amount of minted NFTs must not be greater than the _totalSupply set by DEFAULT_ADMIN_ROLE. There are also some VIP NFTs which can only be minted by DEFAULT_ADMIN_ROLE. They reserved the token ID from 1 to _startIndex (_startIndex is set by DEFAULT_ADMIN_ROLE). DEFAULT_ADMIN_ROLE can also giveaway NFTs to anyone but the number of giveaway NFTs must not be greater than the _maxGiveaway set by DEFAULT_ADMIN_ROLE. _startIndex, _maxGiveaway and _totalSupply can only be set by DEFAULT_ADMIN_ROLE when activate the contract, they can't be changed if there are any NFT minted or _startIndex is set to greater than 0 (VIP NFT's token IDs reserved).

2.2. Findings

During the audit process, the audit team had identified no vulnerable issues in the smart contracts code, only some notes and recommendations.

2.3. Additional notes and recommendations

2.3.1. NeighborheadzNFT.sol - Unused _ReentrancyGuard INFORMATIVE

The contract extends ReentrancyGuard, but it is not used anywhere, consider removing it.

UPDATES

- *Apr 09, 2022*: This issue has been acknowledged and fixed by the OP3N team in commit ceb002622ca0bd738e9d317b22c5e360d8cff549.

APPENDIX

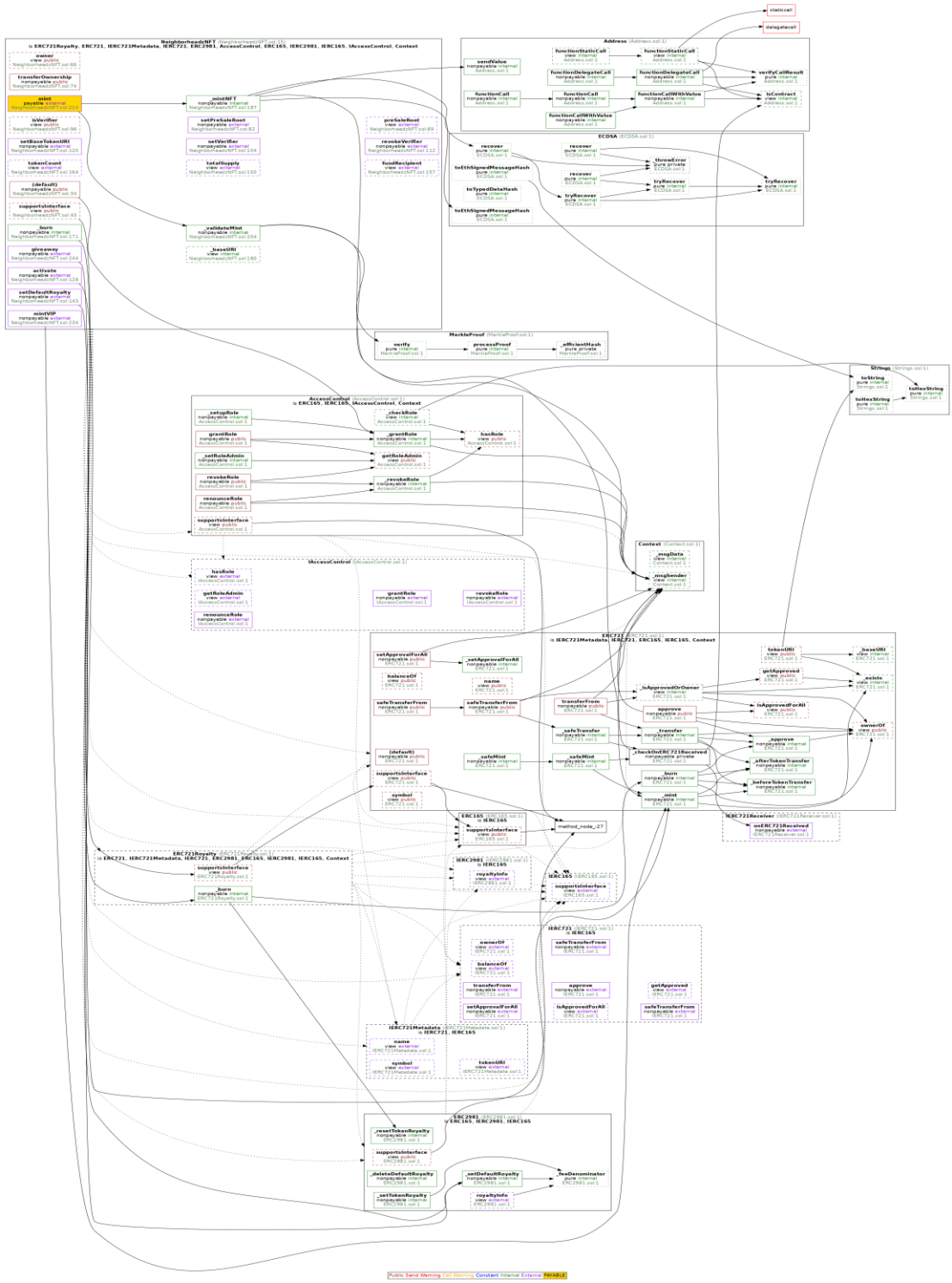

Image 1. OP3N neighborheadz NFT Smart Contracts call graph

3. VERSION HISTORY

Version	Date	Status/Change	Created by
1.0	*Apr 08, 2022*	Public Report	Verichains Lab
1.1	*Apr 09, 2022*	Public Report	Verichains Lab

Table 2. Report versions history