*SECURITY AUDIT OF*

# BYTHENCHIP NFT SMART CONTRACT

## Public Report

*May 03, 2024*

## 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 was prepared by Verichains Lab on May 03, 2024. We would like to thank the Bythen 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 BythenChip NFT Smart Contract. 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 application, along with some recommendations.

# TABLE OF CONTENTS

# 1. MANAGEMENT SUMMARY

## 1.1. About BythenChip NFT Smart Contract

Bythen is building an AI product protocol that enables the creation of AI-powered, 3D avatar agents with autonomous capabilities, allowing humans to achieve more with less effort.

The BythenChip contract is an ERC721-compliant smart contract that allows user to buy nft through minter signature.

## 1.2. Audit scope

This audit focused on identifying security flaws in code and the design of the smart contracts of BythenChip NFT Smart Contract.

It was conducted on commit `b2916acac475f23fe2cd9d0eb4b1e2bfcfc263f9` from git repository link: *https://github.com/tanookiai/smart-contracts/*.

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

| SHA256 Sum | File |
|---|---|
| d0ed9a2f615733388c9c90c520c12de8cc5acfce5bfe92c0e07624a4ed0d24c4 | ERC721LazyMintWith712Signature.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 were 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

Bythen acknowledges that the security services provided by Verichains, are conducted to the best of their professional abilities but cannot guarantee 100% coverage of all security vulnerabilities. Bythen understands and accepts that despite rigorous auditing, certain vulnerabilities may remain undetected. Therefore, Bythen agrees that Verichains shall not be held responsible or liable, and shall not be charged for any hacking incidents that occur due to security vulnerabilities not identified during the audit process.

## 1.5. Acceptance Minute

This final report served by Verichains to the Bythen will be considered an Acceptance Minute. Within 7 days, if no any further responses or reports is received from the Bythen, the final report will be considered fully accepted by the Bythen without the signature.

# 2. AUDIT RESULT

## 2.1. Overview

The BythenChip NFT Smart Contract was written in `Solidity` language, with the required version to be `^0.8.20`. The source code was written based on thirdweb's library.

### 2.1.1. BythenChip contract

The BythenChip contract is ERC721-compliant and builds upon the ERC721A implementation from thirdweb's library. It inherits functionality from PermissionsEnumerable, PrimarySale, and ReentrancyGuard (also from thirdweb).

- Deployer Privileges: Upon deployment, the deployer is granted the `MinterRole` and `DefaultAdminRole`. The `DefaultAdminRole` allows adding or removing any roles from other accounts. The `MinterRole` provides control over the contract's minting and burning logic.
- Minting Restrictions: To mint NFTs, users require approval via a signature from a holder of the `MinterRole`.
- Direct Minting/Burning: The `MinterRole` can also directly mint or burn NFTs for other users.

## 2.2. Findings

During the audit process, the audit team found no vulnerability in the given version of BythenChip NFT Smart Contract.

### 2.2.1. Centralized mechanisms INFORMATIVE

The contract currently relies on a centralized mechanism where the `MinterRole` and `DefaultAdmiRole` control the minting and burning of ERC721 tokens. This centralization creates a potential vulnerability, as a compromise of these accounts could allow hackers to exploit the system.

**RECOMMENDATION**

Consider transitioning to a decentralized mechanism or utilizing smart contract-based accounts with enhanced security measures to mitigate the risks associated with the protocol's centralized roles.

# 3. VERSION HISTORY

| Version | Date | Status/Change | Created by |
|---|---|---|---|
| **1.0** | May 03, 2024 | Public Report | Verichains Lab |

*Table 2. Report versions history*