Smart Contract Security Audit Report

A7A5 Digitall Bill Audit

1. Contents

2. General Information

This report contains information about the results of the security audit of the A7A5 (hereafter referred to as "Customer") smart contracts, conducted by Decurity in the period from 21/01/2025 to 23/01/2025.

2.1. Introduction

Tasks solved during the work are:

- Review the protocol design and the usage of 3rd party dependencies,
- Audit the contracts implementation,
- Develop the recommendations and suggestions to improve the security of the contracts.

2.2. Scope of Work

The audit scope included the contracts in the following repository: https://github.com/a7a5-defi/a7a5. Initial review was done for the commit 3bfbe2.

The following contracts have been tested:

- contracts/NFT.sol

2.3. Threat Model

The assessment presumes actions of an intruder who might have capabilities of any role (an external user, token owner, token service owner, a contract). The centralization risks have not been considered upon the request of the Customer.

The main possible threat actors are:

- User,
- Protocol owner,
- Liquidity Token owner/contract.

The table below contains sample attacks that malicious attackers might carry out.

Table. Theoretically possible attacks

Attack	Actor
Contract code or data hijacking *Deploying a malicious contract or submitting malicious data*	Contract owner Token owner
Financial fraud *A malicious manipulation of the business logic and balances, such as a re-entrancy attack or a flash loan attack*	Anyone
Attacks on implementation *Exploiting the weaknesses in the compiler or the runtime of the smart contracts*	Anyone

2.4. Weakness Scoring

An expert evaluation scores the findings in this report, an impact of each vulnerability is calculated based on its ease of exploitation (based on the industry practice and our experience) and severity (for the considered threats).

2.5. Disclaimer

Due to the intrinsic nature of the software and vulnerabilities and the changing threat landscape, it cannot be generally guaranteed that a certain security property of a program holds.

Therefore, this report is provided “as is” and is not a guarantee that the analyzed system does not contain any other security weaknesses or vulnerabilities. Furthermore, this report is not an endorsement of the Customer’s project, nor is it an investment advice.

That being said, Decurity exercises best effort to perform their contractual obligations and follow the industry methodologies to discover as many weaknesses as possible and maximize the audit coverage using the limited resources.

3. Summary

As a result of this work, we have discovered low level security issue. These issues have been addressed and thoroughly re-tested as part of our process.

3.1. Suggestions

The table below contains the discovered issues, their risk level, and their status as of February 3, 2025.

Table. Discovered weaknesses

Issue	Contract	Risk Level	Status
Lack of zero checks	contracts/NFT.sol	Low	Fixed
URI should revert for incorrect tokenId	contracts/NFT.sol	Low	Fixed
URI may not always be distinct	contracts/NFT.sol	Low	Fixed

4. General Recommendations

This section contains general recommendations on how to improve overall security level.

The Findings section contains technical recommendations for each discovered issue.

4.1. Security Process Improvement

The following is a brief long-term action plan to mitigate further weaknesses and bring the product security to a higher level:

- Keep the whitepaper and documentation updated to make it consistent with the implementation and the intended use cases of the system,
- Perform regular audits for all the new contracts and updates,
- Ensure the secure off-chain storage and processing of the credentials (e.g. the privileged private keys),
- Launch a public bug bounty campaign for the contracts.

5. Findings

5.1. Lack of zero checks

Risk Level: Low

Status: Fixed in the commit f734b1.

Contracts:

- contracts/NFT.sol

Description:

The code contains instance where zero address checks and validation is missing.

If the zero address is assigned to critical roles (e.g., `owner`, `compliance`, `accountant`), the contract may become unmanageable, as these roles have privileged access to key functions.

```
constructor(
        string memory name_,
        string memory symbol_,
        address owner_,
        string memory baseUrl_
    ) ERC721(name_, symbol_) {
        // @audit missing zero checks
        _owner = owner_;
        _baseUrl = baseUrl_;
    }
```

Remediation:

Consider validating that address is not equals to zero.

5.2. URI should revert for incorrect tokenId

Risk Level: Low

Status: Fixed in the commit f734b1.

Contracts:

- contracts/NFT.sol

Description:

The `tokenURI` function does not validate if `_tokenId` is a valid token before constructing and returning the URI. As per the EIP-721 standard, the `tokenURI` function should revert if `_tokenId` is not a valid NFT. In the current implementation, the function will concatenate `baseURI` with an empty string (`cid`) if the `_tokenId` does not exist.

Remediation:

Add a check to ensure `_tokenId` is valid before constructing the URI. If `_tokenId` is not valid, revert the transaction.

5.3. URI may not always be distinct

Risk Level: Low

Status: Fixed in the commit f734b1.

Contracts:

- contracts/NFT.sol

Description:

The `tokenURI` function constructs the URI by concatenating a base URI with a `cid` retrieved from `tokenCID`. The `mint` function does not validate whether the provided `cid` already exists for another token. This can lead to multiple tokens sharing the same `cid`, which may cause confusion or issues with token differentiation.

Remediation:

Add a validation step in the `mint` function to ensure that the provided `cid` is not already associated with another token. If it exists, revert the transaction.

6. Appendix

6.1. About us

The Decurity team consists of experienced hackers who have been doing application security assessments and penetration testing for over a decade.

During the recent years, we've gained expertise in the blockchain field and have conducted numerous audits for both centralized and decentralized projects: exchanges, protocols, and blockchain nodes.

Our efforts have helped to protect hundreds of millions of dollars and make web3 a safer place.