Smart Contract Audit Conclusion

by Silver Consulting

24th August 2022

This is the review of Le Vinh's Hardhat-NFT-contract repo. No critical bug found, but there exist a few minor issues.

Findings

ID	Severity	Subject
CVF-1	Minor	Improper approach
CVF-2	Minor	Different pragma directives are used
CVF-3	Minor	Improper Solidity version
CVF-4	Minor	Improper approach
CVF-5	Minor	Bad naming
CVF-6	Minor	Bad naming
CVF-7	Minor	Function declaration
CVF-8	Minor	Function declaration
CVF-9	Minor	Function declaration

Contents

1. Document properties

Version

Version	Date	Author	Description
0.1	Aug. 24, 2022	Anh Nguyen	Initial Draft

Contact

Anh Nguyen

anhnhatnguyen@protonmail.com

2. Introduction

The following document provide the results of the audit performed by Silver Consulting. The audit goal is a general review of the smart contracts structure, critical/major bugs detection and issuing the general recommendations.

We have audited the Hardhat-NFT-contract. Concretely, the following file was audited:

- Contracts/Cutie.sol

3. Detailed Results

3.1 CVF-1 Improper approach

- **Severity** Minor
- **Category** Suboptimal

Description Benign reentrancy vulnerability

Recommendation Apply the `check-effects-interactions` pattern.

Listing 1: Improper approach

External calls:

59 _safeMint(msg.sender,tokenId)

State variables written after the call(s):

60 _setTokenURI(tokenId,TOKEN_URI)

3.2 CVF-2 Different pragma directives are used

- **Severity** Minor
- **Category** Suboptimal

Description Different versions of Solidity are used throughout the OpenZepellin dependencies and contracts/Cutie.sol

Recommendation Use one Solidity version.

Listing 2: Different pragma directives are used

Version used: ['^0.8.0', '^0.8.1', '^0.8.13']

3.3 CVF-3 Improper Solidity version

- **Severity** Minor
- **Category** Suboptimal

Description Pragma version^0.8.13 (contracts/Cutie.sol#2) necessitates a version too recent to be trusted.

Recommendation Consider deploying with 0.6.12/0.7.6/0.8.7

Listing 3: Improper Solidity version

2 pragma solidity ^0.8.13;

3.4 CVF-4 Improper approach

- **Severity** Minor
- **Category** Suboptimal

Description Low level call in Cutie.withdraw(). The use of low-level calls is error-prone. Low-level calls do not check for code existence or call success.

Recommendation Avoid low-level calls. Check the call success. If the call is meant for a contract, check for code existence.

Listing 4: Improper approach

```
70  (status) = (msg.sender).call{value: balance}()
```

3.5 CVF-5 Bad naming

- **Severity** Minor
- **Category** Bad naming

Description Solidity defines a naming convention that should be followed.

Recommendation Constant Cutie.mintPrice should be MINT_PRICE

Listing 5: Bad naming

```
27  Cutie.mintPrice
```

3.6 CVF-6 Bad naming

- **Severity** Minor
- **Category** Bad naming

Description Solidity defines a naming convention that should be followed.

Recommendation Constant Cutie.maxPerTransaction should be MAX_PER_TRANSACTION

Listing 6: Bad naming

```
28  Cutie.maxPerTransaction
```

3.7 CVF-7 Improper Approach

- **Severity** Minor
- **Category** Suboptimal

Description public functions that are never called by the contract should be declared external to save gas.

Recommendation safeMint(uint256) should be declared external

Listing 7: Improper Approach

```
44 Cutie.safeMint(uint256)
```

3.8 CVF-8 Improper Approach

- **Severity** Minor
- **Category** Suboptimal

Description public functions that are never called by the contract should be declared external to save gas.

Recommendation withdraw() should be declared external

Listing 8: Improper Approach

```
67 Cutie.withdraw()
```

3.9 CVF-9 Improper Approach

- **Severity** Minor
- **Category** Suboptimal

Description public functions that are never called by the contract should be declared external to save gas.

Recommendation getSupply() should be declared external

Listing 9: Improper Approach

```
77 Cutie.getSupply()
```