SMART CONTRACT AUDIT

By Le Hoang Vinh

23rd August 2022

This report summaries the audit results of the NFT contract given from Anh's Github repo (UsuaOSilver/fst-nft (github.com)).

1. Introduction

This audit has been made in accordance with the Crystalize.dev blockchain bootcamp, providing the audit performed by Vinh as the assignment requests. The audit goal is an overall review of given smart contract structure, detection of any critical / major / moderate / minor bugs and provision of corresponding general recommendations.

I have audited the fst-nft repository. Specifically, the following file were audited:

- contracts\Nft.sol;

The full details of issues descriptions and suggestions made are presented in the following section.

2. Detailed results

2.1. Different versions of Solidity are used

- **Severity:** Minor
- **Description:**

- Version used: ['0.8.16', '^0.8.0', '^0.8.1']

- 0.8.16 (contracts/Nft.sol#2)

- ^0.8.0 (node_modules/@openzeppelin/contracts/access/Ownable.sol#4)
(node_modules/@openzeppelin/contracts/token/ERC721/ERC721.sol#4)
(node_modules/@openzeppelin/contracts/token/ERC721/IERC721.sol#4)
(node_modules/@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol#4)
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol#4)
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Enumerable.sol#4)
(node_modules/@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol#4)

(node_modules/@openzeppelin/contracts/utils/Context.sol#4)

(node_modules/@openzeppelin/contracts/utils/Counters.sol#4)

(node_modules/@openzeppelin/contracts/utils/Strings.sol#4)
(node_modules/@openzeppelin/contracts/utils/introspection/ERC165.sol#4)
(node_modules/@openzeppelin/contracts/utils/introspection/IERC165.sol#4)

- ^0.8.1 (node_modules/@openzeppelin/contracts/utils/Address.sol#4)

- **Recommendation:** Change the version of every contracts to 0.8.0/0.8.1 for consistency.

2.2. Literals with too many digits:

- **Severity:** Minor
- **Description:** Literals with many digits are difficult to read and review.

- PRICE = 10000000000000000 (contracts/Nft.sol#22)

- **Recommendation:** Use ether suffix to shorten literals.

2.3. Conformance to Solidity naming convention:

- **Severity:** Minor
- **Description:**
    - Variable FstNFT.TOTAL_SUPPLY (contracts/Nft.sol#19) is not in mixedCase
    - Constant FstNFT.maxNftPurchase (contracts/Nft.sol#25) is not in UPPER_CASE_WITH_UNDERSCORES
    - Variable FstNFT.URI (contracts/Nft.sol#27) is not in mixedCase

- **Recommendation:** Follow solidity naming convention.

2.4. State variable should be declared constant:

- **Severity:** Minor
- **Description:** Constant state variables should be declared constant to save gas.
  FstNFT.TOTAL_SUPPLY (contracts/Nft.sol#19) should be constant

Recommendation: Add the constant attributes to state variables that never change.

2.5. Public function that could be declared external

- **Severity:** Minor
- **Description: public** functions that are never called by the contract should be declared **external** to save gas.
  mintNft(uint256) should be declared external (contracts/Nft.sol#37-68)
  refund() should be declared external (contracts/Nft.sol#72-75)
- **Recommendation:** Use the **external** attribute for functions never called from the contract.

3. Summary

The use of given smart contract is simple, with the size of the code being relatively small. Overall, the code demonstrates effective use of abstraction, separation of concern, and modularity. However, there are a few problems/vulnerabilities that need to be fixed at different security levels before any additional work or deployment.