Smart Contract Audit (NFT COLLECTION)

By Umair Mirza (Discord: dreamygeek#8033)

I was asked to review the NFT Collection smart contract developed by **John Nguyen** (Github). This report is a part of the Crystalize Bootcamp assignment. The contract has been audited using Slither Static Analysis Framework for Smart contracts.

Below are the findings of the smart contract audit:

1. Findings

ID	Severity	Subject
2.1	Low	Reentrancy in NFTContract.mint(uint256)
2.2	Informational	solc-0.8.13 is not recommended for deployment
2.3	Informational	Low level call in NFTContract.withdraw()

2. Details

2.1. Reentrancy in NFTContract.mint(uint256)

- **Severity:** Low
- **Description:** State variables written after the call(s):
  - _safeMint(msg.sender,tokenID) (contracts/NFTContract.sol#52)
  - _setTokenURI(tokenID,TOKEN_URI) (contracts/NFTContract.sol#53
- **Recommendation:** State variables should be updated before the mint function call.

2.2. solc-0.8.13 is not recommended for deployment

- **Severity:** Informational
- **Description:** Pragma version^0.8.13 (contracts/NFTContract.sol#2) necessitates a version too recent to be trusted.
- **Recommendation:** Consider deploying with 0.6.12/0.7.6/0.8.7

2.3. Low level call in NFTContract.withdraw()

- **Severity:** Informational
- **Description:** The use of low-level calls is error-prone. Low-level calls do not check for code existence or call success. (contracts/NFTContract.sol#66)
- **Recommendation:** Avoid low-level calls. Check the call success. If the call is meant for a contract, check for code existence.