Saigon Marketplace Audit

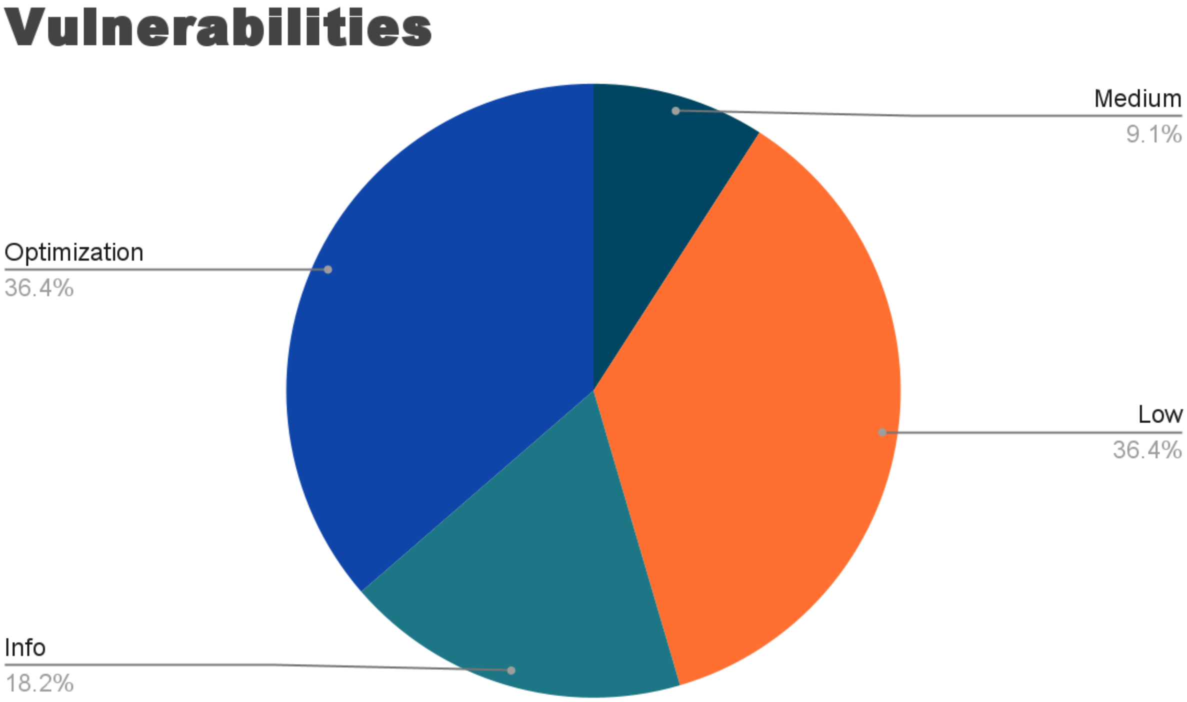
By John Nguyen (jooohn.eth)

General Info

Resources:
Github repo which consists of the project's core smart-contracts, tests, user interface and documentation.

Project author:
Anh Nguyen (UsuaOSilver)

Audit author:
John Nguyen (jooohn.eth)



Summary

An NFT marketplace that allows users to buy, sell, and create their own auction houses with the special Variable Rate GDA (VRGDA) price strategy. Payment is accepted with any ERC20 tokens depending on the auction house owner. ERC721 and ERC1155 token standards are supported. Royalties will be set by the creator of the NFT collection to honor creatives and artists.

The main branch of Saigon Marketplace was reviewed.
Covered:

- SaigonMarket.sol and SaigonNftFactory.sol - main contracts that are used to interact with the project.
- SaigonMarket.test - contract's unit tests and mocks.

The project was reviewed manually and with the help of tools.

Scope:

Github Repo
Commit

The commit reviewed was fce2728c9da7428502a8d48339e16f9434fcb681. The review covered the repository at the specific commit and focused on the contracts directory.

Code Evaluation Matrix

Category	Mark	Description
Access Control	Okay	No access control was used. Access control not needed at this stage but highly recommended.

Libraries	Okay	Only Openzeppelin’s libraries were used. Using a lot of external libraries leads to a lot of external dependency.
Documentation	Good	All comments were provided where needed.
Monitoring	Good	Events exist for all important functions that modify state variables.
Testing	Good	All tests passed with a good percentage of code coverage.
Decentralization	Good	No external party access provided.

Findings Explanation

Findings are broken down into sections by their respective impact:

- Critical, High, Medium, Low impact
  - These are findings that range from attacks that may cause loss of funds, impact control/ownership of the contracts, or cause any unintended consequences/actions that are outside the scope of the requirements.
- Gas Savings
  - Findings that can improve the gas efficiency of the contracts
- Informational
  - Findings including recommendations and best practices

No Critical and High Findings

Medium Findings

1. **Reentrancy risk.**
   - Location: SaigonNFTFactory.mint
   - Description: state variables modified after external call

```
_safeMint(msg.sender, newTokenId);
_setTokenURI(newTokenId, _tokenURI);
```

- Recommendation: modify state variables before calling an external contract.

```
_setTokenURI(newTokenId, _tokenURI);
_safeMint(msg.sender, newTokenId);
```

Low Findings

2. **Reentrancy risk**
    - Location: SaigonNFTFactory.mint
    - Description: event emitted after external call.
    - Recommendation: emit events before making external calls

```
 emit Minted(_tokenURI);
_setTokenURI(newTokenId, _tokenURI);
```

3. **Reentrancy risk**
    - Location: SaigonMarket. createListing, createMarketSale, reselNFT
    - Description: events emitted after external call.
    - Recommendation: emit events before making external calls

```
emit Event(parameters);
_externalCall(parameters);
```

Informational Findings

4. Naming convention not followed.

- Description: function parameters do not follow Solidity conventions.
- Recommendation: rename variables according to Solidity naming conventions.

5. Unused global variable.

- Location: SaigonMarket

```
  bytes4 private constant INTERFACE_ID_ERC1155 = 0xd9b67a26;
```

- Recommendation: remove unused variables.

Gas Saving Findings

6. Function visibility

- Description: function visibility could be limited to external to save gas.
- Recommandation: modify function visibility to external

```
    function createMarketSale(address _nftAddress, uint256 _listingId) external
payable isListed(_listingId) nonReentrant {}

    function fetchMarketListings(address _nftAddress) external view returns {}
    function fetchMyNFTs(address _nftAddress) external view returns {}
    function fetchOwnedListings(address _nftAddress) external view returns {}
```

Final Remarks

After reviewing the core smart contracts, no critical and high vulnerabilities were found, mostly low-level or informational issues occurred, one medium level vulnerability and a few gas saving recommendations. Unit tests were reviewed - no anomalies found, the tests were accurate.