Rentals Contract Audit Report

Date: 09/30/2022

Auditor: Xinghui Chen
Email: ChenXinghui@protonmail.com

Summary

Scope

Common Contracts
https://github.com/decentraland/common-contracts/tree/93a3471df9e53b11bf2f36fc8139f8ae8f99b34e

Rentals
https://github.com/decentraland/rentals-contract/tree/bfe7522314ae24d27aa927d74dee4718b90e55a7

Every contract inside /contracts except for the ones in the /mocks folder.

Findings

	Critical	High	Middle	Low	Gas	Recommendation	Misc
NativeMetaTransaction.sol	-	-	-	-	1	-	-
AssetNonceVerifiable.sol	-	-	-	-	-	-	-
ContractNonceVerifiable.sol	-	-	-	-	-	-	-
SignerNonceVerifiable.sol	-	-	-	-	1	-	-
IERC721Rentable.sol	-	-	-	-	-	-	1
Rentals.sol	-	-	-	-	5	5	1

Details

NativeMetaTransaction.sol

1. In function **executeMetaTransaction**, param `_functionData` and `_signature` are declared as memory, it's better using calldata as it's more gas friendly(`_signature` param in `_verify` function can be declared as calldata too).

SignerNonceVerifiable.sol

1. As **_bumpSignerNonce** is only called by **bumpSignerNonce**, it's better removing `_bumpSignerNonce` and move its logic into `bumpSignerNonce`.

IERC721Rentable.sol

1. Line 17 and 22, `memory` can be changed to `calldata` as these params are actually `calldata` in `EstateRegistry.sol`.

Rentals.sol

Comment mistake

1. Line 202, **Get if and asset..** should be **Get if an asset..**.

Gas optimization

1. In function **acceptListing**, param `_listing` is declared as memory, it's better using calldata as it's more gas friendly(`_listing` param in `_verifyListingSigner` function can be declared as calldata too). Same for other external functions:
    - function **acceptOffer** , param `_offer`.
    - function **claim**, all params.
    - function **setUpdateOperator**, all params.
    - function **setManyLandUpdateOperator**, param `_landTokenIds` and `_operators`.
    - function **onERC721Received**, param `_data`.

2. As the existence of line 275, line 270 and 271 can be removed.

3. line 402 use `.length` in the loop condition, it's better saving it first. e.g.,

```
uint256 tokenIdLength = _landTokenIds.length;
for (uint256 i; i < tokenIdLength; ++i) {..}
or
for (uint256 i; i < tokenIdLength; ) {
    ..
    unchecked {
      ++i;
    }
}
```

   Same for line 314, 359.

4. line 321 `Rental memory rental =` can be changed to `Rental storage rental =` as the logic below only access its one field one times. As Rental struct has 3 members, if use memory here, will cost more than 3 SLOAD(cold), if use storage, will only cost 1 SLOAD(cold). Same for line 362, 393.

Out of block gas limit

1. Line 553 calls `verifyFingerprint` in `EstateRegistry.sol`, which may cause `out of gas` if the estate contains too many lands. In my test(use line 1302 in `Rentals.spec.ts`), 3000 lands cost around 9_000_000 gas(x∈[-8,7), y∈[-100,100)), 10000 lands will fail(x∈[-25,25), y∈[-100,100)). I think it's ok as few people will trigger it.

Recommendation

1. Line 428 use `msg.sender` directly. As the contract supports meta-transactions, it's better using `_msgSender()` everywhere in case of unintended behavior.
2. Line 7 can be removed as `Rentals` doesn't inherit `OwnableUpgradeable` directly.
3. Term `asset` is arbitrary: It means a nft in the comment, means nft contract in the code(line 330, 549).
4. `pragma solidity ^0.8.7` could be changed to `pragma solidity 0.8.7`(all files), for more detail please visit here.
5. In function **_rent**, when `isExtend` is true or `isReRent` is true, should not try to transfer asset any more(line 600-602). Instead, shall assert that current contract must already own the asset. So maybe can change as below:

```
if (isExtend || isReRent) {
    require(_ownerOf(_rentParams.contractAddress, _rentParams.tokenId) ==
address(this), "Not own(accept[Listing|Offer])");
} else {
    // Verify that the asset is not already rented. (line 566)
    require(!isRented, "Rentals#_rent: CURRENTLY_RENTED");
    if (_msgSender() == _rentParams.contractAddress) {
        require(_ownerOf(_rentParams.contractAddress, _rentParams.tokenId)
== address(this), "Not own(onERC721Received)");
    } else {
        asset.safeTransferFrom(_rentParams.lessor, address(this),
_rentParams.tokenId);
    }
}
```

Then remove line 600-602 and function `_verifyUnsafeTransfer` (not need deal unsafe transfer situation specially any more).

PS

1. The contract depends on `OpenZeppelin 4.5.0,` please be aware that it has some known issues, all these won't affect current contract currently.