Bid Contract Audit Report

Audited contract

Source code: ERC721Bid.sol
Address: Not deployed yet

Related contracts and assumptions

MANA contract

During the audit we assumed that ERC721Bid#manaToken pointed to the smart contract located at `0x0f5d2fb29fb7d3cfee444a200298f468908cc942`.

Source code: `MANAToken.sol`

Contracts ownership

We assumed that once deployed, `ERC721Bid` will be owned by Decentraland.

Audit Results

Critical severity

[DCL3-C01] Bids can be rendered invalid by anyone

An error in the `ERC721Bid#cancelBid` function can be exploited to render any bid invalid. An attacker could take advantage of this error to ensure nobody else can win a bid he's interested in.

If the bid being canceled isn't the last one submitted for a token, the last one will take the place of the removed one, but its index won't be updated. This results on `ERC721#onERC721Received` considering it invalid, so the owner of the token won't be able to accept it.

Update: This issue was independently discovered by Decentraland during the audit, and was fixed in this commit.

Comments and recommendations

[DCL3-O01] Unsupported Solidity version

This contract was developed using Solidity 0.4.x, which isn't supported anymore. While there are no known security issues for this version, we recommend migrating it to Solidity 0.5.x.

This migration would require rewriting the entire test suite if Truffle is to be used, but by using Buidler, with its buidler-truffle4 plugin, the migration can be really straightforward, as explained in this tutorial.

[DCL3-O02] Pre-release OpenZeppelin version being used

This project uses OpenZeppelin version 2.1.0-rc1. We don't recommend using pre-release versions. This version in particular is not only pre-release, but version 2.1.0 doesn't exist.

If [DCL3-O01] were to be addressed and the contracts migrated to Solidity 0.5.x, we recommend using the latest release of OpenZeppelin. If Solidity 0.4.x is used, we recommend using OpenZeppelin 2.0.0.

[DCL3-O03] `ERC721Bid#_requireComposableERC721`'s name is confusing

This name seems to imply that the function will fail if the token isn't a composable ERC721, but it doesn't.

[DCL3-O04] `ERC721#_placeBid` does not clear previous state

If the bidder has an active bid and places a new bid on the same token, this function will not clear the state of the previous bid in `bidIndexByBidId`.

[DCL3-O05] `ERC721Bid#MIN_BID_DURATION`'s name is confusing

The `MIN_` prefix seems to imply that it's an inclusive bound, which it isn't.

[DCL3-O06] Creating the same bid multiple times in a block results in unexpected behavior

If a user creates the exact same bid more than once, and those transactions end up in the same block, they will get the same id, resulting in multiple `BidCreated` emitted with the same data.

[DCL3-O07] There's no way to know that a bid was overwritten

If a user places multiple bids for the same token, the last one overwrites the previous ones, effectively canceling them. No event is emitted in this case. Depending on the intended usage of this contract, adding one may be useful.

[DCL3-O08] `ERC721#_bidderHasAnActiveBid`'s name is confusing

This function doesn't check bids' expiration dates. This results in inactive bids being reported as active. We recommend renaming it.

[DCL3-O09] `ERC721#onERC721Received`'s `msg.sender` validation is not trivial to understand

When using this function to accept a bid, it validates that the `msg.sender` is the bid's token registry in a fairly contrived way. It uses `_getBid(msg.sender, _tokenId, bidIndex)` to get the bid, so it can only get bids for that token registry. Then, by validating that `bid.id == bidId`, it further limits it to the bid being accepted.

We recommend documenting this behavior, or making it more explicit.

[DCL3-O10] Incorrect error message if `MAX_BID_DURATION` is exceeded

When placing a new bid, if its duration exceeds the `MAX_BID_DURATION` value, an error is emitted which indicates that it can't be larger than 6 months. This message is incorrect, as 6 months is slightly over 26 weeks, and the constant is set to 24 weeks.

We recommend setting the constant to `180 days` or `182 days`, which is more accurate.

[DCL3-O11] wei is used to incorrectly denote units of MANA

The name wei is used to express "units of MANA". This can be confused, as it's normally used to express units of ETH. We recommend choosing a name for such a concept, and using it across the different Decentraland projects. Our suggestion, as discussed with a member of the team, is naming a unit of MANA a *tibit*.