Security Audit
Decentraland Land Auction

Saturday, 01-Dec-18

Agustin Aguilar

Introduction

The Decentraland Team requested the review of the project on the repository https://github.com/decentraland/land-auction/, the commit referenced for this audit is 0xd304396.

The audited contracts are:

- LANDAuctionStorage.sol: Defines the storage layout for LANDAuction.sol.
- LANDAuction.sol: Handles the reverse auction of the remaining parcels of the Decentraland world, burns or forwards the tokens used to buy the parcels.
- KyberConverter.sol: Provides abstraction and helper functions to interact with KyberNetwork

The contracts are well written, no high severity issues were found.

Issues

Medium severity

Three medium severity vulnerabilities were found on the project, they seem to be compromises made by the Decentraland team to save gas or keep the contract code simple, but they could hide vulnerabilities, and they should be carefully examined by the team before production release.

1 - Possible denial-of-service attack against large bids.

The land auction contract allows a bidder to buy multiple parcels in a single transaction; if any of the parcels is already owned, the whole transaction fails.

An attacker could front-run and buy a single parcel to provoke a failure of the whole transaction. The original bidder could not realize about the collision and end up losing the bid.

The same situation could happen by accident if most users try to batch-bid at the same time.

Proposal:
Add functionality to batch-bid and ignore parcels already owned.

Note: Some bidders may only want to batch-bid upon the warranty of buying all parcels or none, a flag could be added, to enable the bidder to request the failure if one the parcels is no longer available.

2 - Lack of specification in the scope of the land auction

A user could bid and buy any land parcel, the only requirement on the LANDAuction.sol contract is that all the parcels are between 150 and -150 x & y coordinates.

It is recommended to add additional validations to avoid selling districts, plazas, and roads.

At the time of writing this audit, 01/12/2018, some districts have parcels non-owned, if these parcels are not assigned yet, they will be on sale during the auction, (see 3).

Proposal:
a - Add additional validations on the land auction contract.
b - Properly document the scope of the land auction.

Note: Additional comments on the scope of the token sale could improve the auditability of the contracts, the desired behavior may be to sell all the non-owned parcels, but this should be specified.

3 - Lack of ownership validations in the auction contract

The land auction contract calls the method *assignMultipleParcels* without any validation, without knowing the specific behavior of the land registry contract, the auction could end up selling parcels that are already owned.

The current land registry contract (block 6791965) throws when trying to assign a parcel already owned; nonetheless, the land registry is upgradeable and it's outside of the scope of this report, so those validations may not be effective or present during the final auction.

Proposal:
Check the ownership of the parcel directly from the land auction contract prior to assign it to the new buyer.

Low severity

Low severity bugs/vulnerabilities were found on the contract code, they don't compromise the security and solidity of the land auction contract; but they should be addressed to improve setup, compatibility, and user experience.

4 - Lack of support for tokens without a return value

To be compatible with LANDAuction.sol contract, an allowed token should return true upon a successful transaction; this is not met in some implementations, like OMG (OmniseGO), BNB (Binance coin) and other not fully ERC20 compliant tokens.

Proposal:
a - Implement a *_safeTransfer* inner method for handling ERC20 token transfers, the implementation of this helper method should check the balances before and after the transfer, and ignore the result of the transfer if *RETURNDATASIZE* is 0.
b - Restrict the allowed tokens to only fully compliance ERC20 tokens.

Note 2: The Decentraland team addressed this issue at commit 863a3c2 creating the SafeERC20 library.

5 - Unchecked MANA token transfers

The current mana token implementation (0x0f5d2fb29fb7d3cfee444a200298f468908cc942) uses *assert()* to check allowance and balance during the transfer operation; this would cause a loss of all gas if by accident a user tries to bid without funds or without allowing the transfer of MANA.

Proposal:
Check balance and allowance before transferring MANA tokens; this solution could be implemented alongside 4A.

Note 2: The Decentraland team addressed this issue at commit 863a3c2 creating the SafeERC20 library.

6 - Lack of validation in defined token decimals

When allowing a new third party token, the number of decimals must be manually defined, and this value is never validated.

Proposal:
Add validation of the input using the *decimals*() method specified on the ERC20 standard.

7 - Incomplete abstraction in dex implementation

The handling of decimals is being done directly by the LandAuction.sol contract but is only required when interacting with KyberNetwork. Migrating to use another dex would require to modify the core auction contract.

Proposal:
Move all handling of decimals to KyberConverter.sol

Notes

- Not using solidity constants

It's recommended to use Solidity constants to represent time unit like days: *24 * 60 * 60* on the line 65 could be replaced with *1 day* to improve readability.

Note: Fixed at 863a3c2

- Malleable transaction ID

The ID of the bid is an auto-increment uint; this scheme makes the traceability of bids only possible after the confirmation of the transaction;

Proposal:
Consider replacing it with a deterministic id.

- Dispersed input validation

Input validation on bid calls is performed on *_validateBidParameters*, but validation of parcel bounds are performed before calling *assignMultipleParcels*.

Proposal:
Consider moving validation of bounds of x & y to *_validateBidParameters*

Note: Fixed at 863a3c2

- Mixing of uint and uint256

LANDAuction.sol contract uses, in most cases, uint256 to declare integers, except for 132, 221, 369 and 553. It will be preferable to migrate all integers to the same syntax.

Note: Fixed at 863a3c2

- Improperly documented slope formula

The documentation of the _getFunc method specifies the base formula as:
*(x2 * y1) - (x1 * y2) / x2 - x1*
but is implemented on the method as
*(x2 * y1) - (x1 * y2) / **(x2 - x1)***

Note: The implemented formula seems to be correct and the comment seems to be a typo, it should be fixed to avoid confusion.

Note: Fixed at 863a3c2

- Tests are written using a different implementation for MANA token

All tests use the ERC20 implementation provided by OpenZepellin, but this implementation differs from the deployed MANA token.

Proposal:
Use the same code as deployed for *manaToken* on the LANDAuction tests.

Note:
Using the real implementation could have helped to detect number 5.

- Incorrect parameter type in _validateBidParameters docs

The documentation specifies two uint256[] inputs, but the method takes two int256[].

Note: It seems to be a typo in the documentation.

Note: Fixed at 863a3c2

- *_processFunds* has undocumented behavior

_processFunds requires balance in *_token* and requires balance in *manaToken,* or reverts the transaction.

When buying a land using a 3rt party token, a 5% margin of the original token is always kept, so the processFunds method does not fail; this behavior is not detailed on the processFunds method.

Proposal:
Document the behavior of _processFunds, and the requirement to have funds in manaToken and _token.

Note: Partially fixed at 863a3c2, process funds only works if the contract has balance in manaToken and _token, this behavior is not documented properly.

- Copying struct to memory without being required

When reading the current price using *_getPrice* for each past curve, the method copies all the struct to memory, but only(N - 1) times the *limit* field is read, this could increase the gas cost of the function if multiple curves are configured.

Proposal:
Change **memory** keyword to **storage**

Note: The extra gas cost becomes negligible if few curves are going to be defined.

Note 2: Fixed at 863a3c2

- Lack of sanity checks ~~of~~ constant conditions

_calculateRequiredManaAmount expects *PERCENTAGE_OF_TOKEN_TO_KEEP* to be less than 100 and the _processFunds method will throw if *PERCENTAGE_OF_TOKEN_TO_KEEP* is set to 0.

Proposal:
Add a sanity check during the contract construction.

Note: Fixed at 863a3c2

- Payable KyberConverter convert function

KyberConverter convert method is declared as payable, but it does not implement any handling of ETH.

Proposal:
a - Assert msg.value == 0 during convert
b - Remove payable from the interface
c - Add support for converting from ETH to Tokens

Note: Fixed at 863a3c2

- Lack of integration tests with KyberNetwork

LANDAuction.sol contract is closely tied to the Kyber Network platform to allow non-mana bids. This integration is only tested using KyberMock.sol, considering the lack of total abstraction of the dex interactions, some edge cases of Kyber Network could not be tested lacking integration tests.

Proposal:
a - Add integration tests with KyberNetwork

Final thoughts

The auction contracts are well written, documented, and have unit testing with full coverage; no critical vulnerabilities or bugs were found on LANDAuction.sol or KyberConverter.sol. Is important to meet the following conditions before production release:

1 - All parcels that should not be included on the auction should have an owner assigned.

2 - The LANDRegistry contract **should** fail in assignMultipleParcels if any of the parcels are already owned.

3 - The front-end should warn the bidder about the high chance of failing if trying to bid multiple parcels at the same time.

Update 2018/12/10

The Decentraland team addressed some of the most critical issues before the release, the biggest change is the addition of a SafeERC20 library, this library acts as a normalizer of ERC20 behavior.

The new changes are well written and ready for production.

December 2018, Madrid, Spain - by Agustin Esteban Aguilar