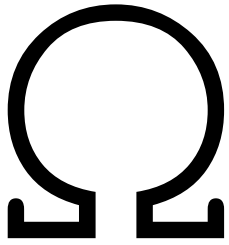
EnterDAO Landworks

Audit of Referral Mechanism

Final Audit Report

September 28, 2022



Team Omega

teamomega.eth.limo

Summary

EnterDAO has asked Team Omega to audit the revenue share mechanism that was added to the Landworks system that we audited previously.

We found **no high or medium severity issues** - these are issues that can lead to a loss of funds, and are essential to fix, or that we believe are necessary to address. We classified **6** issues as “low”, and **5** issues as “info” - we believe the code will improve if these issues were to be addressed.

None of these issues present a security risk.

EnterDAO subsequently addressed some of the issues we found - of the issues we mentioned, 3 were resolved.

Severity	Number of issues	Number of resolved issues
High	0	
Medium	0	
Low	3	0
Info	5	3

Scope of the Audit

Scope of the Audit

The scope of the audit is the revenue-share mechanism that was added to the Landworks code. The scope is given by all changes in the solidity code between the commits:
`https://github.com/EnterDAO/LandWorks-protocol/compare/145e22e23d058ddc98c7b4796ab37557f23d7572...c546ec90eca443ff4c2510a13cdd44b03e26b7b1`

In particular, the scope of the audit regards the files:

```
contracts/facets/ReferralFacet.sol [123 cloc]
contracts/facets/RentFacet.sol [65 cloc]
contracts/libraries/LibReferral.sol [30 cloc]
contracts/libraries/marketplace/LibRent.sol [224 cloc]
```

Together with minor related changes in `MarketPlaceFacet.sol`, `MetaverseConsumableAdapterFacet.sol`, `DecentralandFacet.sol`, `LibFee.sol` and the associated tests.

Some of the issues were subsequently addressed in commit

```
29fc142f243ec33239f8a54a60b933eaa22503fd
```

We reviewed these changes and noted our observations below.

Methods Used

Code Review

We manually inspected the source code to identify potential security flaws.

The contracts were compiled, deployed, and tested in a test environment.

Automatic analysis

We have used static analysis tools to detect common potential vulnerabilities. No vulnerabilities were detected using the static analysis tools.

Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

Severity definitions

High	Vulnerabilities that can lead to loss of assets or data manipulations.
Medium	Vulnerabilities that are essential to fix, but that do not lead to assets loss or data manipulations
Low	Issues that do not represent direct exploit, such as poor implementations, deviations from best practice, high gas costs, etc
Info	Matters of opinion

Findings

General

G1. Use Solidity Custom Errors instead of strings [low] [not resolved]

The use of custom errors, that are available in Solidity since version 0.8.4, instead of error strings, will reduce gas costs. Cf. https://blog.soliditylang.org/2021/04/21/custom-errors/
Recommendation: Use Custom Errors instead of error strings.
Severity: Low
Resolution: The issue was not addressed, as it would require a refactor that was considered too large.

G2. Secondary fee percentage is not settable by list referrer or by rent referrer [low] [not resolved]

In the documentation string in the `rent()` function in `LibRent.sol` we read that:

```
/// The lister itself might take percentage of the list referral based on
`secondaryPercentage`,
/// adding an additional amount to the rent reward.
```

and

```
/// The renter itself might take percentage of the rent referral based on
`secondaryPercentage`,
/// which will serve as discount to the initial rent amount.
```

There is no place in the contracts where either the list referrer or the rent referrer can “take a percentage” of the fee to give to the lister or the renter by setting `secondaryPercentage`. Based on these doc strings and the way the fees are structured,, we assume this is an omission.
Recommendation: To be consistent with the doc strings, add setters for lister and renter to set the `secondaryPercentage`. If instead the error is in the documentation, we recommend to flatten the fee structure and simplify the fee calculations.
Resolution: The issue was not addressed - the team reported this was the current business requirement.

G3. Rent referrer and list referrer fee structures are mixed up [low] [not resolved]

The values of `mainPercentage` and `secondaryPercentage` for a rent referrer determine a discount for the renter, while for list referrers these values determine a payback of some of the fees to the consumer/owner of the original asset. The values for list referrers and rent referrers are managed in a single mapping, `referrerPercentages`.
This means that if an address functions as both a list referrer and a rent referrer, it will apply the same percentages to these two discounts. This complicates choosing the best values for the referral fees, as any pair of values that is chosen for (or by) a list referrer will automatically become available to renters, who are free to choose any whitelisted referrer address when calling `rent`.
Recommendation: Use different mappings for `listReferrerPercentages` and `rentReferrerPercentages`.
Severity: Low
Resolution: The issue was not resolved - the team reported this is the current business requirement.

G4. Use latest Open Zeppelin version [info] [not resolved]

The contracts use version 4.4.0 of the `@openzeppelin/contracts` package. A newer point release is available (4.4.1) and the latest minor version is 4.7.3. Although to our knowledge none of these later versions address any issues that are relevant for the current code base, we recommend upgrading in any case.
Recommendation: Update the `@openzeppelin/contracts` package.
Severity: Info
Resolution: The issue was not addressed.

G5. Use latest Solidity version [info] [not resolved]

The contracts use Solidity version 0.8.10. The latest version is 0.8.17, which contains some important bug fixes (although none of these fixes seem directly relevant for the current code).
Recommendation: Update the code to use Solidity 0.8.17.
Severity: Info
Resolution: The issue was not addressed.

G6. Use a uniform value for the precision of fractions [info] [resolved]

The contracts use two different values for the precision of percentages. The `feePercentage` uses the precision given by `FEE_PRECISION`, which is set to 100_000. Instead, the various referral fees are represented with a precision of 10_000. This is confusing.
Recommendation: Define a single constant (e.g. `FEE_PRECISION`) and use that throughout the system to represent the precision of percentages.
Resolution: The issue was addressed as recommended.

G7. Constant ETHEREUM_PAYMENT_TOKEN is defined twice [info] [resolved]

The constant `ETHEREUM_PAYMENT_TOKEN` is defined in LibTransfer.sol and in LibRent.sol. A single definition suffices.
Recommendation: Define the constant only once.
Severity: Info
Resolution: The issue was addressed as recommended.

G8. Unnecessary loading of variable on updateAdapterAdministrativeState [info] [resolved]

The function `updateAdapterAdministrativeState` loads the `metaverseConsumableAdapterStorage` into `mcas` variable on line 232, but this is then never used and should be removed.
Recommendation: Remove the `mcas` variable on line 232.
Severity: Info
Resolution: The issue was addressed as recommended.