

Smart Contract Audit

Linguo

By,
Shebin John
16th June 2020

Index

1. [Introduction](#)
2. [High Threats](#)
3. [Medium Threats](#)
4. [Low Threats](#)
5. [Optimization & Readability](#)
6. [Typo's & Comments](#)
7. [Suggestions](#)

Note: Some threat levels or headings might be empty if there is no vulnerability/updates/suggestions found.



Introduction

The contract audited here is/are:

<https://github.com/kleros/kleros-interaction/blob/master/contracts/standard/arbitration/Linguo.sol>
<https://github.com/kleros/kleros-interaction/blob/master/contracts/standard/arbitration/LinguoToken.sol>

and was shared by Kleros for auditing purposes while working with them.

Based on the audit, we were able to find no High threats, no Medium threat, and 2 Low threat with some other changes in the optimization, readability, typos, and comments section.



High Threats

1. None

Medium Threats

1. None

Low Threats

1. [submitEvidence\(\)](#) can be used by anyone to submit 100s of fake evidence which increases the work of the Jurors. It can be used by someone who wants to hurt Kleros. It can be used by the Challenger also, as a reverse psychology method to make the jurors frustrated and think that the translator might have done this, as his work might be below average. Simple solution, limiting the evidence proposals of each task to only the challenger.
2. In [line 522](#) if the requester does not accept the token transfer, it can lead to DoS situation.



Optimization & Readability

1. [batchRoundWithdraw\(\)](#) can be optimized if we find the iteration limit before the for loop, rather than in the for loop. The below code might help I believe:

```
uint iterateUntil = tasks.rounds.length;
```

```
if (_count != 0) { iterateUntil = tasks.rounds.length > (_cursor + _count) ? (_cursor + _count):  
tasks.rounds.length; }
```

Typo's & Comments

1. None

Suggestions

1. None