Smart Contract Audit

Binary Arbitrable Proxy

By,
Shebin John

21st July 2020

Index

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/binary-arbitrable-proxy-contract/blob/c4704ba/contracts/BinaryArbitrableProxy.sol

Related Contracts:
https://github.com/kleros/binary-arbitrable-proxy-contract/blob/b736597/contracts/AutoAppealableArbitrator.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 1 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. This won't compromise the Blockchain working but can mess up the UI. The disputeID in createDispute emits the event first and then assigns the disputeID. Making it to provide the disputeID always as 0.

Optimization & Readability

1. None

Typo's & Comments

1. None

Suggestions

1. Assigning the owner or the governor inside a constructor using msg.sender
2. Dispute instead of Dispute Struct
3. Instead of line 65 and 67, creating a mapping from disputeID to DisputeStruct and including the round info in the struct itself, will reduce a lot of code like the use of externalIDToLocalID, simplify the createDispute() and so on without any compromise in security, saving gas as well due to fewer steps to execute and data redundancy. This is considering the arbitrator is not changed, which in this contract, is initialized at the constructor and is not changed as well.