Primitive

Primitive Bits, Inc.
New York, NY
primitive.xyz

January 16, 2024

Arbiter Security Review

Engagement II

Waylon Jepsen	Lead
Colin Roberts	Lead
Could be you	Researcher

1 Executive Summary

Over the course of X days in total, Protocolname engaged with Spearbit to review Protocolname protocol.

We found a total of X issues with Protocolname.

Repository	**Commit**
Projectname	commithash

Summary

Type of Project	TYPE
Timeline	Feb 29, 2022 - Feb 31, 2022
Methods	Manual Review
Documentation	High
Testing Coverage	High

Total Issues

Critical Risk	1
High Risk	1
Medium Risk	0
Low Risk	0
Gas Optimizations and Informational	0

Contents

2 Primitive

Primitive is a team of deeply technical passionate indciduals, building the future of finance. You can find more information about us at primitive.finance.

3 Introduction

The focus of the security review was on the following:

1. Measure the saftey critical properties of the dispute games.
2. Measure the economic security of the bond mechanisms.
3. Statefully fuzz the system over different conditions.

Disclaimer: This security review does not guarantee against a hack. It is a snapshot in time of brink according to the specific commit by a three person team. Any modifications to the code will require a new security review.

3.1 Agent Based Modeling

Arbiter uses agent based modeling with the rust evm to provide security and risk analysis insights that are traditionally more difficult to audit. Our agent architecture for the dispute game is as follows:

- Oracle Agent: Responsible for syncing the dispute game state by loading the latest claim and then solving the correct move for that claim by making an api call to durin. The oracle agent will then send honest moves to the honest agent.
- Honest Agent: The Honest Agent is responsible for receiving the honest moves from the oracle agent and then acting on them in the dispute game.
- Dishonest Agent: The Dishonest Agent is responsible for acting attempting to resolve an incorrect move in the dispute game, We will perturb the dishonest agent to look for insecurities in the protocol.

3.2 Simulation Components

The system is composed of several agents and contracts. Below is a table summarizing these components:

- **Agents:**

  - Oracle Agent: Responsible for syncing the dispute game state by loading the latest claim and then solving the correct move for that claim by making an api call to durin. The oracle agent will then send honest moves to the honest agent.
  - Honest Agent: Responsible for receiving the honest moves from the oracle agent and then acting on them in the dispute game.
  - Dishonest Agent: Responsible for attempting to resolve an incorrect move in the dispute game. We will perturb the dishonest agent to look for insecurities in the protocol.
- **Contracts:**
  - Dispute Game: Holds all moves currently in the dispute game.
  - Dispute Game Factory: Has pointers to all created dispute games.
- **Oracle:**
  - Durin: A single oracle used in the system.

3.3 Risk Modeling

We will perturb over the infinite space of dishost actors for the dispute game. We will also perturb various L1 preposals. This will enable us to model the risk of the protocol in a more robust way.

4 Findings

4.1 Critical Risk

4.2 High Risk

4.2.1 Issue title (Only first word should be capitalized; titles should never end with punctuation)

Severity: High

Context: `Contract.sol#L160-L165`

Description:

```
contract Test {
    ...
    // Code blocks must be indented with 4 spaces.
}
```

Recommendation:

```diff
+ use diff syntax to describe what should be changed
- ...
```

Project: Fixed in PR #1.

Spearbit: Resolved.

4.3 Medium Risk

4.4 Low Risk

4.5 Gas Optimizations

5 Additional Comments

\clearpage

6 Appendix