Security Audit Results

Prepared on: 16 May 2023

Contract: 688

Prepared by:

Charles Holtzkampf

Sentnl

Prepared for:

Casper Network

Table of Contents

Executive Summary

Sentnl were hired by Casper Network to perform extensive fuzzing on the Execution engine (WASM execution and smart contract processing) and Node (the network-facing component of the project)

Additionally we have also provided Casper Network with an custom fuzzer and a extensive fuzzing corpus to allow for continuous fuzzing by the their internal team.

We **found 0 further issues** with the Node engine. Further to our fuzzing we have also provided access to our proprietary fuzzer and the entire corpora used during our fuzzing process.

REMARK	MINOR	MAJOR	CRITICAL
0	0	0	0

Severity Description

REMARK

Remarks are instances in the code that are worthy of attention, but in no way represent a security flaw in the code. These issues might cause problems with the user experience, confusion with new developers working on the project, or other inconveniences.

Things that would fall under remarks would include:

- Instances where best practices are not followed
- Spelling and grammar mistakes
- Inconsistencies in the code styling and structure

MINOR

Issues of Minor severity can cause problems in the code, but would not cause the code to crash unexpectedly or for funds to be lost. It might cause results that would be unexpected by users, or minor disruptions in operations. Minor problems are prone to become major problems if not addressed appropriately.

Things that would fall under minor would include:

- Logic flaws (excluding those that cause crashes or loss of funds)
- Code duplication
- Ambiguous code

MAJOR

Issues of major security can cause the code to crash unexpectedly, or lead to deadlock situations.

Things that would fall under major would include:

- Logic flaws that cause crashes
- Timeout exceptions

CRITICAL

Critical issues cause a loss of funds or severely impact contract usage.

Things that would fall under critical would include:

- Missing checks for authorization
- Logic flaws that cause loss of funds
- Logic flaws that impact economics of system
- All known exploits

Methodology

The second milestone consisted of auditing the Casper node software.

- Fuzzing of the Node software RPC and P2P ports using a large corpus.
- Continuously fuzz the RPC and P2P endpoints until there are no furher crashes reported.

Fuzzer software - Casper Audit

Fuzzer software

Our fuzzer contains our entire corpora used to fuzz the Casper software.
The execution engine sits here - https://github.com/ankh2054/casper-audit/tree/master/execution-engine which contains a build script which will build the software

https://github.com/ankh2054/casper-audit

Once build you can continuously run the fuzzer against any future versions of the Casper software.