Security Audit Results

Prepared on: 22 Feb 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 2 issues** with the WASM engine.

REMARK	MINOR	MAJOR	CRITICAL
0	0	0	2

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 first milestone consisted of auditing the WASM engine.

- Fuzzing of the WASM engine using a large corpus.
- Continuously fuzz the WASM engine until there are no furher crashes reported.

WASM/execution engine

Assert that for any WASM input, perform an equivalent execution for each given input code, do not crash, do not cause memory bugs, do not hang or take a long time to execute.

Audit Results – Casper WASM Engine

Casper Out of Memory bug

Our fuzzer produced a WASM that when uploaded to chain, caused the node software to utilise 32GB of memory causing the node software to crash. 60 Seconds after deploying the wasm, all nodes in network crash due to out of memory bug.

Out of memory bug confirmed by looking at /var/log/syslog

```
Jul 27 14:42:35 localhost kernel: [3817527.960921] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=user.slice,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-1174.scope,task=casper-node,pid=777250,uid=0
Jul 27 14:42:35 localhost kernel: [3817527.961013] Out of memory: Killed process 777250 (casper-node) total-vm:860550992kB, anon-rss:15791452kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:35132kB oom_score_adj:0
Jul 27 14:42:36 localhost kernel: [3817528.841931] oom_reaper: reaped process 777250 (casper-node), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
Jul 27 14:42:35 localhost systemd[1]: message repeated 4 times: [ session-1174.scope: A process of this unit has been killed by the OOM killer.]
```

WASM Deployed as following:

```
/root/casper-client-rs/target/release/casper-client put-deploy --chain-name casper-net-1 --node-address http://localhost:11101 --secret-key /root/casper-node/utils/nctl/assets/net-1/users/user-1/secret_key.pem --session-path /root/casper-oom.wasm --payment-amount 1500000000000 && tail -f /root/casper-node/utils/nctl/assets/net-1/nodes/node-1/logs/stdout.log
```

After about 60 seconds all nodes in network crash due to out of memory bug.
4. /var/log/syslog - shows the below message in regards to the casper-node using too much memory.

Effected Node version: casper-node 1.4.6-6e1f65e22-casper-mainnet (Fixed in 1.4.7)

Audit Results – Casper WASM Engine

WASM Slow input - DOS

Our fuzzer produced a WASM that took 20 minutes to upload which would cause a major DOS on the casper network.

Bug was reported to Medha via email and confirmed on Telegram on 28 Nov 2022.

Fix: Increased the control flow opcode cost to 440000 motes within the chainsepc cost tables. *https://github.com/casper-network/casper-node/blob/5622735454e1fd69c074b607c86fcff5b36b494b/execution_engine/src/shared/opcode_costs.rs#L30*

Effected Node version: casper-node 1.4.8