Prepared on: 5 April, 2023

Contract: AUD468

# Security Audit Results

**Prepared by:**

Charles Holtzkampf

Sentnlio Ltd

**Prepared for:**

EOS Network Foundation

# Table of Contents

## Executive Summary

Sentnl were hired by the EOS Network foundation to perform extensive fuzzing on the EOS EVM and verify that the EVM engine implementations is bug-free.

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

We **found 0 issues** with the repositories tested.

| 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 audit consisted of two parts:**

- Differential fuzzing of the EVM.
- Differential fuzzing of the cryptographic precompiles.

During the fuzzing we looked for:

- memory bugs
- undefined behavior bugs
- resource exhaustion bugs
- instances of disproportional slowness
- stack overflow bugs
- consensus bugs, e.g. deviations from the logic mandated by the canonical specifications

within your EVM implemented by https://github.com/eosnetworkfoundation/eos-evm/tree/main/contract (commit -39b3098181) and its dependencies. Silkworm - .https://github.com/eosnetworkfoundation/silkworm/tree/0ed5362304a1cca16bf0a0cbd6cc7693bcfa1668)

**The code from** https://github.com/eosnetworkfoundation/mandel/blob/main/libraries/chain/webassembly/crypto.cpp ( v3.1.0-rc3) was used to implement the precompiles, along with the mandel-fc (commit - c5f5d1876ca1c722769990f49827f34b39e4a6bc) library which implemented the core functionality.

## EVM differential fuzzer

## Results

Construction and deployment of a differential fuzzer which verifies the correct operation of both the EVM core and the precompiles with a large degree of certainty. The differential fuzzer was delieverd here - https://github.com/guidovranken/eos-evm-audit