L2Labs

ZKSwap Circuit

Security Assessment

February 6th, 2021

By:
Georgios Delkos @ CertiK
georgios.delkos@certik.org

Camden Smallwood @ CertiK
camden.smallwood@certik.org

Patrick Ventuzelo @ FuzzLabs
ventuzelo.patrick@gmail.com

Disclaimer

CertiK reports are not, nor should be considered, an “endorsement” or “disapproval” of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any “product” or “asset” created by any team or project that contracts CertiK to perform a security review.

CertiK Reports do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors, business, business model or legal compliance.

CertiK Reports should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.

CertiK Reports represent an extensive auditing process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk. CertiK’s position is that each company and individual are responsible for their own due diligence and continuous security. CertiK’s goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies, and in no way claims any guarantee of security or functionality of the technology we agree to analyze.

What is a CertiK report?

- A document describing in detail an in depth analysis of a particular piece(s) of source code provided to CertiK by a Client.
- An organized collection of testing results, analysis and inferences made about the structure, implementation and overall best practices of a particular piece of source code.
- Representation that a Client of CertiK has completed a round of auditing with the intention to increase the quality of the company/product's IT infrastructure and or source code.

Overview

Project Summary

Project Name	L2labs: ZkSwap - Circuit
Description	Circuit portion of the zkSwap repository
Platform	Proprietary; Rust
Codebase	GitHub Repository
Commits	1. 1fb223956953eed471a7bb8736cebd26849f703c

Audit Summary

Delivery Date	Feb. 6, 2021
Method of Audit	Static Analysis, Manual Review, Fuzzing
Consultants Engaged	3
Timeline	Dec. 20, 2020 - Jan. 17, 2021

Executive Summary

L2Labs requested for CertiK to perform an audit on the Circuit portion of their new zkSwap protocol, which is based on ZK-Snarks. The auditing team conducted the audit in the timeframe between December 20, 2020, and January 17, 2021, with 3 engineers. The auditing process evaluated code implementation against provided specifications, examining language-specific issues, and performed fuzzing against identified endpoints.

The audit's main outcome is that the implementation of the zkSwap Circuit is well constructed and aligns with the specifications put forth by L2Labs, with some minor exceptions that are related to the original implementation in the zkSync repository, which zkSwap is forked from.

The audit team examined the code in 3 phases. The first phase was onboarding the auditors to the codebase, where the team deep-dived the system's specifications to understand the intended functionality and design decisions. The team started with the system's original specifications and then moved to the team's specifications regarding the newly added functionality. System design, structures, and pseudo code were all taken under consideration to move to the next phase.

In the second phase, the team identified the implementation's structure, endpoints, and critical functionality. Every related structure of the system was identified and noted to be further examined in the third phase. Additionally, the team identified the modifications that the team has made to the original system.

In the third and final phase, the team focused on evaluating the specifications' implementation to the code itself. By taking the specifications and pseudocode, the team performed a line by line check to ensure that it matches the design.

Structural artifacts and composition of the various operation primitives were examined for correctness against the specifications, expanding on the functionality, ensuring that the codebase performs the various operations and respects the desired outcome. The code examined respects the given specifications with some minor exceptions that do not affect the overall sentiment but would be essential for the code's maintainability and readability.