Trail of Bits
228 Park Ave S #80688
New York, NY 10003

Jim Miller
Engineering Director
james.miller@trailofbits.com
www.trailofbits.com

March 25, 2025

Scroll

Scroll engaged Trail of Bits to review the security of changes related to the Euclid Upgrade. These changes include OpenVM guest programs for the chunk, batch, and bundle verification circuits, changes to the Scroll smart contracts related to supporting a special state transition due to the change in the computation of the state root, and a Go command-line tool to validate states during the Euclid transition.

A team of two consultants conducted the review from February 24 to March 5, 2025, for a total of three engineer-weeks of effort. Our testing efforts focused on assessing the soundness and correctness of the zero-knowledge circuits, whether the Euclid finalization function works as specified, and whether the Go command-line tool can validate that the state of two nodes is identical at a certain block height. With full access to source code and documentation, we performed static and dynamic testing of the codebase, using automated and manual processes.

Our audit uncovered two high-severity issues, two low-severity issues, and four informational issues.

We identified two high-severity soundness issues affecting the batch and bundle circuits that would allow obtaining a bundle proof for which a malicious prover could bypass the verification of the inner proofs for the chunk and batch circuits.

We identified two impactful but difficult-to-exploit issues related to the GitHub actions used in the codebases that could undermine the security of the released images to DockerHub. Note that GitHub actions security was not in the audit's scope; we identified these issues using a static analysis tool for GitHub actions.

We did not identify security issues in the smart contract related to the one-off Euclid finalization functionality or in the Go command-line utility. However, we found that the migration

checker was tested only with test data and does not have associated unit tests for its subfunctions.

Sincerely,
Jim Miller