Polygon Pessimistic Proofs

Security Assessment (Summary Report)

Aug 26, 2024

Prepared for:
Christopher Von Hessert
Polygon Labs

Prepared by: **Joe Doyle**

About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 100+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence with regard to blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Ethereum 2.0, MakerDAO, Matic, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

Trail of Bits, Inc.
497 Carroll St., Space 71, Seventh Floor
Brooklyn, NY 11215
https://www.trailofbits.com
info@trailofbits.com

Notices and Remarks

Copyright and Distribution

Test Coverage Disclaimer

All activities undertaken by Trail of Bits in association with this project were performed in accordance with a statement of work and agreed upon project plan.

Security assessment projects are time-boxed and often reliant on information that may be provided by a client, its affiliates, or its partners. As a result, the findings documented in this report should not be considered a comprehensive list of security issues, flaws, or defects in the target system or codebase.

Trail of Bits uses automated testing techniques to rapidly test the controls and security properties of software. These techniques augment our manual security review work, but each has its limitations: for example, a tool may not generate a random edge case that violates a property or may not fully complete its analysis during the allotted time. Their use is also limited by the time and resource constraints of a project.

Table of Contents

Project Summary

Contact Information

The following project manager was associated with this project:

Jeff Braswell, Project Manager
jeff.braswell@trailofbits.com

The following engineering director was associated with this project:

Jim Miller, Engineering Director, Cryptography
james.miller@trailofbits.com

The following consultant was associated with this project:

Joe Doyle, Consultant
joe.doyle@trailofbits.com

Project Timeline

The significant events and milestones of the project are listed below.

Date	Event
August 16, 2024	Pre-project kickoff call
August 26, 2024	Delivery of report draft
August 26, 2024	Report readout meeting
TBD	Delivery of summary report

Project Targets

The engagement involved a review and testing of the following target.

pessimistic-proof

Repository	https://github.com/agglayer/agglayer
Version	`d9d33885b6a396a69f57054d7b549edc6ca1ef51`
Type	Rust
Platform	Native, SP1

Executive Summary

Engagement Overview

Polygon engaged Trail of Bits to review the security of their pessimistic-proof crate, commit `d9d33885b6a396a69f57054d7b549edc6ca1ef51`. This crate implements validation logic to ensure certain minimally-correct behavior of inter-blockchain bridge systems even in the presence of unreliable blockchains.

A team of one consultant conducted the review from August 19 to August 23, for a total of one engineer-week of effort. With full access to source code and documentation, we performed static and dynamic testing of the target, using automated and manual processes.

Observations and Impact

The primary focus of the engagement was to address the following questions:

- Are the authenticated data structures – sparse and append-only Merkle trees – implemented correctly?
- Does the logic ensure that transfers and balances are correctly validated?
- Do signatures and data commitments include all required data?

During the review period, the Polygon team independently found a high-severity vulnerability that would have allowed receiving a given bridge exit multiple times to different networks. We reported one medium-severity issue and two low-severity issues related to missing fields in data structure commitments, which would allow a high-privilege attacker to corrupt pessimistic-proof data structures. In addition, we reported several informational-severity issues related to hazardous implementation choices and APIs. While none of the informational-severity issues appear to be exploitable in the current implementation, several of them impose undocumented requirements on the overall implementation which, if violated in future versions, could lead to state corruption or proof forgery.

Overall, the `pessimistic-proof` crate appears to be a well-designed and focused library, but it shows signs of insufficient testing, in particular deficiencies when handling edge cases and maliciously-crafted inputs. Safely deploying software which regulates the movement of high-value assets requires a high level of assurance. That level of assurance requires an adversarial and thorough test suite, and the current testing of the library does not meet those criteria. The Polygon team should invest in developing and refining a thorough and aggressive test-suite that covers all execution paths through validation before deploying this library.

Recommendations

We recommend expanding both unit and integration testing of the pessimistic proofs implementation, especially focused on negative tests. The Polygon team should include tests that create transactions that trigger every possible rejection. For example, a test that tries to import bridge exits to the wrong destination network would have discovered the high-severity vulnerability immediately. Similarly, the low-severity issues found would be caught by including tests that attempted to tamper with the signature or the Merkle tree size.

We also recommend that the Polygon team create more extensive documentation for their data structures, to ensure that they do not get reused without correct precautions. For example, the sparse Merkle tree implementation in this crate would enable an attacker to create fraudulent lookup proofs that would be accepted by an implementation that verifies proofs with an incorrect total tree height, or that accepts a variable-length proof.

We also recommend adding dylint to the continuous integration process. Two of the informational issues were highlighted by running the general patterns from.