Qeios

Peer Review

Review of: "SARMF-Bench: A Reproducible Smart Contract Vulnerability Benchmark Dataset"

Yuusuke Harada[1]

1. Hiroshima University, Japan

Overall Impression

This paper addresses a long-standing and critical issue in the evaluation of smart contract static analysis tools: the lack of reproducibility. The proposed approach—combining a strictly version-locked analysis environment with structured JSON outputs—provides an excellent foundation for future researchers to conduct fair and consistent evaluations. I highly commend the author's dedication to open science best practices. To further refine the manuscript and enhance its impact, I offer a few constructive suggestions below.

Strengths

Exceptional Commitment to Reproducibility: Providing a strictly pinned execution environment (e.g., Slither v0.11.5, solc-select) brilliantly addresses the "version drift" problem that frequently plagues longitudinal tool evaluations.

Exemplary Open Science Practices: Archiving the dataset and artifacts across multiple repositories with persistent DOIs (Zenodo, Harvard Dataverse, OSF, etc.) is a model approach to data availability and sharing.

Readiness for Automation: Delivering both the ground truth and findings in a machine-readable JSON format makes this benchmark highly practical for seamless integration into CI/CD pipelines and automated evaluation harnesses.

doi.org/10.32388/08B4MH

Constructive Feedback & Suggestions

1. Dataset Size and Paper Positioning The design philosophy of using intentionally minimal contracts to isolate specific vulnerability patterns is highly logical and analytically sound. However, because the dataset currently consists of only five contracts (mapping to five SWC classes), readers might perceive a gap between the expectations set by the term "benchmark dataset" and the actual scale of the artifacts, especially when compared to larger prior works like SmartBugs.

Suggestion: If publishing at the current scale, it might be beneficial to explicitly frame the paper's primary contribution as a **reproducible evaluation protocol or Proof-of-Concept (PoC)** for future large-scale benchmarking. Alternatively, adding a small number of contract variations would significantly boost the paper's robustness as a standalone dataset.

2. Demonstrating Cross-Tool Interoperability A major strength of this work is the standardized JSON output designed to facilitate automated comparisons. Currently, the baseline results rely solely on Slither.

Suggestion: Since the format aims to improve interoperability, adding a brief demonstration with a second tool (such as Mythril) to show exactly how outputs from different tools can be uniformly parsed and compared would make the framework's utility much more apparent. If this is out of scope for the current paper, explicitly detailing this as immediate future work in the Discussion section would be helpful.

3. Clarification on Table 1 (SC05 Severity) In Table 1, Slither's detection impact for SC05 (DoS via Unbounded Loop: SWC-113) is reported as "Informational / Medium." Given that Denial-of-Service vulnerabilities can have severe operational and financial consequences for smart contracts, readers might find this classification counterintuitive.

Suggestion: Adding a brief 1-2 sentence explanation in the Discussion section as to *why* Slither categorizes this specific pattern as "Informational" would deepen the reader's understanding of the tool's inherent characteristics and limitations.

Conclusion

The proposed methodology of strict version control combined with structured outputs is highly valuable to the smart contract security research community. By addressing the points regarding the dataset's scale framing and perhaps adding a brief multi-tool comparison (or clarifying it as future work), I am confident this will become an even stronger and more influential paper.

Declarations

Potential competing interests: No potential competing interests to declare.