This paper is really interesting. It can be seen as addressing the “credible conditional recall” problem in an SGX enclave, because it implements a rapid “snapshot” and “reset” functionality. It’s implemented as a virtual machine layer for running guest processes within an SGX enclave.

bpb-us-w2.wpmucdn.com

[

](https://bpb-us-w2.wpmucdn.com/u.osu.edu/dist/0/113190/files/2023/07/Reusable-Enclave.pdf)

Reusable-Enclave.pdf

726.60 KB

[

image

955×571 81.4 KB

](https://collective.flashbots.net/uploads/default/original/2X/6/6ed48ff14f7b0a13c99bddc7a73790d62b9211be.png)

There’s a lot of other insights on hardening too, including when running Wasm code compiled to native@tolak

The implementation is summarized in this instructions page (other in forked version of other projects).

GitHub

GitHub - OSUSecLab/Reusable-Enclaves: Implementation of the Reusable Enclaves...

Implementation of the Reusable Enclaves paper. Contribute to OSUSecLab/Reusable-Enclaves development by creating an account on GitHub.

The paper is framed as about “serverless” functions, but my intended use would rather be to “clone” of a long-running program, getting a copy of its active stack/heap memory. But the paper seems to be about resetting to a nearly-initial state, so there isn’t as much to copy. I was hoping for something that worked like copy-on-write to get the best of both, so maybe that still remains open to try.