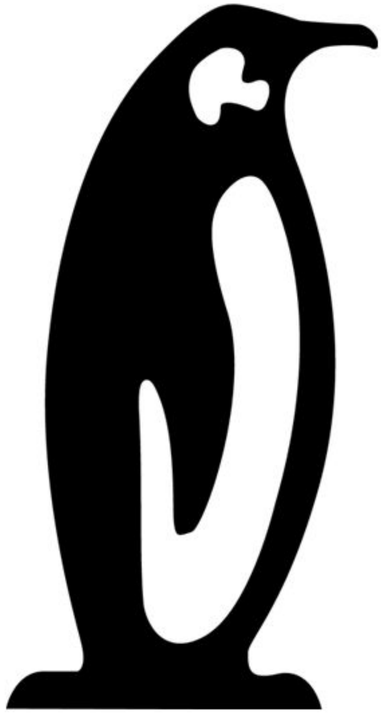


Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

Address Space Isolation

x86 Microconference

Brendan Jackman <jackmanb@google.com>

Context links:
2024 RFC | Demo branch (with tests + optimisations) | LSF/MM/BPF + recordings

LINUX PLUMBERS CONFERENCE | Vienna, Austria
Sept. 18-20, 2024

Agenda

- Hasty background refresher if needed? (5 mins)
- A look at some perf data (2 mins)
- Discuss how to get this thing merged

(Some) CPU Exploits refresher

Attacker domain
(guest/userspace)

Mistrain branch predictors

→

Victim domain
(host kernel)

Mis-speculate, access secret

↓

secret residual in microarchitecture

→

Attacker domain
(guest/userspace)

recover secret

→

Classic example: load from address depending on secret value, that address gets cached.

Check access timing to see which address is cached. “Flush+Reload”

(Some) CPU Exploits refresher

Attacker domain
(guest/userspace)

Mistrain branch predictors

Can't happen if secret isn't mapped

Mis-speculate, access secret

↓

esidual in
rchitecture

Chance to intervene on transition (clear mistraining)

Attacker domain
(guest/userspace)

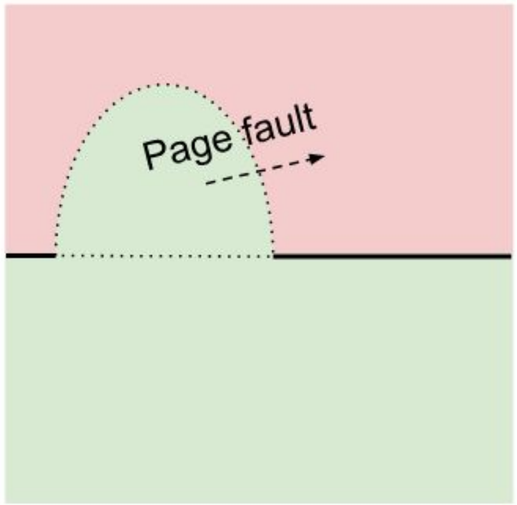
recover secret

Check access timing to see
ess is cached.
oad"

Classic example: load from
address de
value, that

Chance to intervene on transition (flush data buffers)

ASI refresher



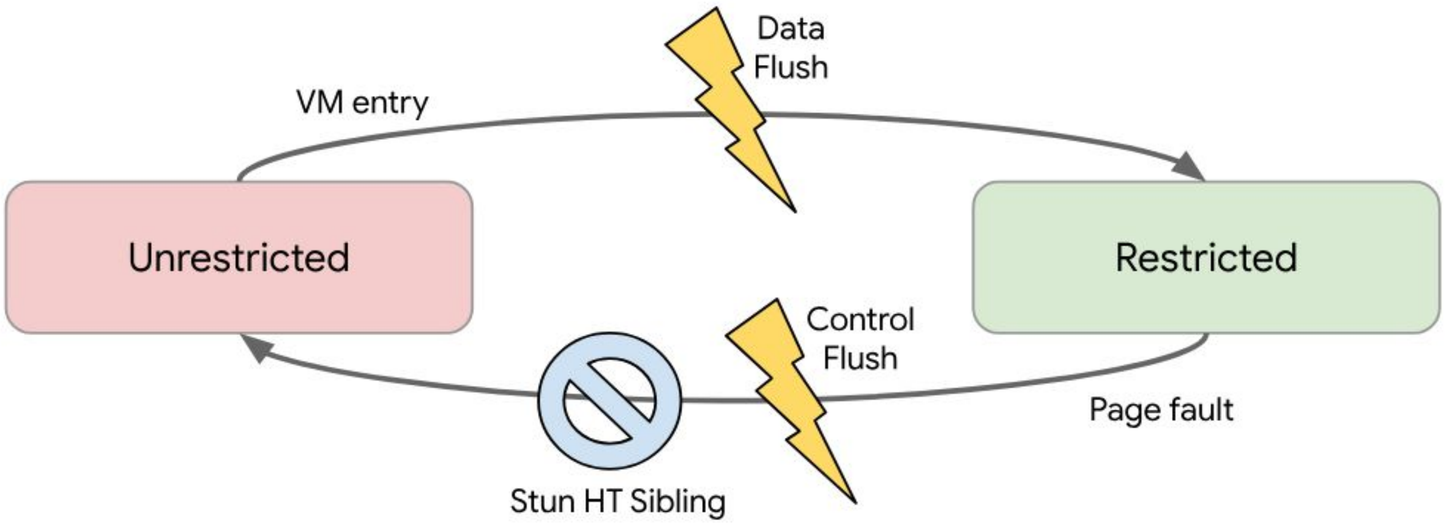
Kernel

Userspace/Guest

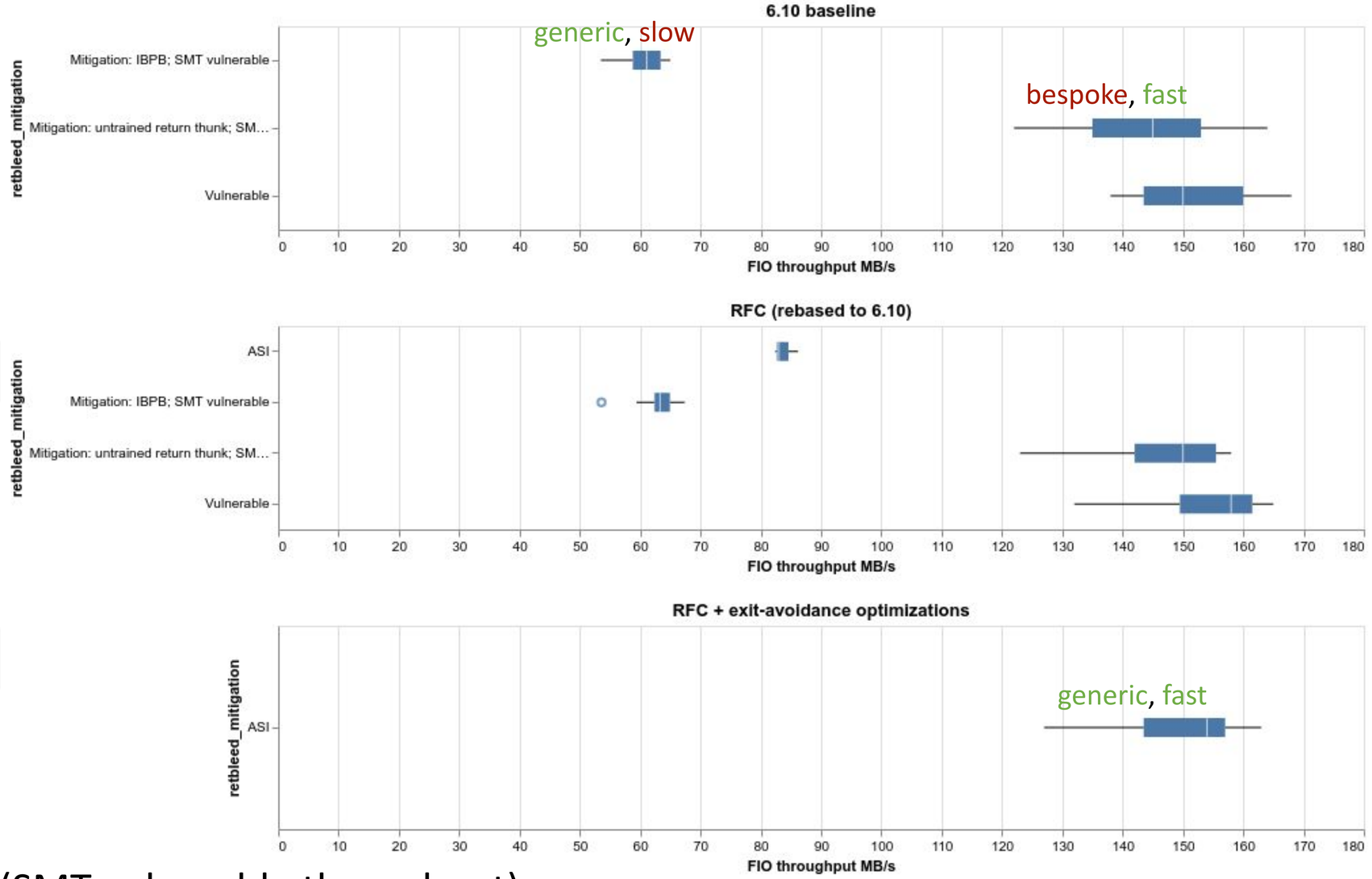
Unrestricted address space (secrets mapped)

Restricted address space (secrets unmapped)

ASI refresher



Performance (Zen2)



(SMT vulnerable throughout)

Performance = comparable to bespoke mitigations

Security properties = comparable to sledgehammer mitigations

I presented something similar at LSF/MM/BPF. Claimed low-confidence in the data. But now it's a different benchmark, different platform, more evidence for same conclusion. Also matches experience in Google's kernel.

Austria
20, 2024

Topics for discussion

- Thoughts on RFC…?
- mm folks seemed up for it at LSF/MM/BPF (but Mel wasn’t there)
- “Denylist” approach: start with only protecting GFP_USER directmap
  - Probably prevents all existing attacks, but obviously not watertight
  - But lets us work in-tree on an ASI that’s actually viable for production
  - Build up security from there, with a meaningful performance baseline
- Roadmap for bare-metal sandboxing
- How should users configure it?
- We have tests (KUnit, e2e exploits that stop working)
  - Testing mitigations is hard though

Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024