



Contribution ID: 253

Type: **not specified**

Address Space Isolation is ready

Last year at LPC I presented the latest status of Address Space Isolation.

The key feedback was: we aren't really interested if it only works for the KVM use-case. x86 folks would still need to develop & maintain the bespoke mitigations. We only want it if it also protects against native attackers.

Since then, I've developed a version that does that. It performed very badly! At LSF/MM/BPF this year, I mostly focussed on the design for the page cache, but in a hasty "bonus section" I also proposed a design to fix the major performance issue (slides). Well, now I've implemented it and shared the prototype. This solves the problem of ASI killing performance of workloads that access file memory.

While that prototype is indeed just an early prototype, I'm claiming that it's strong enough evidence to show that the big and scary unknowns for ASI's viability are "solved". That means we're finally ready to evolve from slideware and RFC-ware into "real" code.

At the time of writing, there's no [PATCH] series on the list yet, but I'm actively preparing it. This is just a first series of many, since ASI is too big to review all at once. Optimistically, if that series is already merged by the time of LPC, we'll need time to discuss the strategy for evolving towards a complete implementation. Pessimistically, if it's still on v15, we'll need some time to discuss blockers and work through what's needed to get it ready for merge.

Primary author: JACKMAN, Brendan (Google)

Presenter: JACKMAN, Brendan (Google)

Session Classification: x86 MC

Track Classification: x86 MC