Conversions: when/how to update trusted firmware?

For 2026-03-19 guest_memfd bi-weekly upstream call

Contact ackerleytng@google.com if you have questions/suggestions!

Current Status: v4 prepared, have some questions

- `SET_MEMORY_ATTRIBUTES2` guest_memfd ioctl: set shared/private state
- Userspace can specify flags to tell KVM what it expects of the memory content after conversion
  - `ZERO`: Userspace can expect that when it next reads the converted memory, it will read zeros
    - `EOPNOTSUPP` for shared to private conversions
  - `PRESERVE`: If host writes `0xbeef`, guest will read `0xbeef` after conversion (and encryption/decryption, if any). If guest writes `0xbeef`, host will read `0xbeef`.
    - Applies in both directions
  - `UNSPECIFIED` (default): No guarantees. Userspace cannot assume anything about the memory contents after conversion.

[1] RFC v3: https://lore.kernel.org/all/20260313-gmem-inplace-conversion-v3-0-5fc12a70ec89@google.com/T/

When to update trusted firmware?

- When handling conversion
  - If `to_shared`, `kvm_arch_gmem_invalidate()`
  - If `to_private`, don’t update trusted firmware
- During faults (`kvm_gmem_get_pfn()`)
  - If `fault->private`, `kvm_gmem_prepare_folio()`
  - If `fault->shared`, don’t update trusted firmware

TDX

- Doesn’t need any separate updates to trusted firmware
- Unmapping == make it not private
- Mapping == make it guest-only, host-only or shared

SNP

- Conversion to shared requires `rmp_make_shared()`
  - If to_shared, `rmp_make_shared()` from guest_memfd conversion ioctl
  - If to_private, do nothing
- Conversion to private requires `rmp_make_private()`
  - If `fault->private` calls `kvm_gmem_prepare_folio()` to make private
  - If `fault->shared`, do nothing
- Is this sufficient, or should guest_memfd not do nothing in the above two cases?
- Michael: prepare should take into account private/shared-ness
  - Prepare based on the attributes within guest_memfd, since it is more directly matched with the userspace's requested state
  - Prepare always == make private, guest_memfd call prepare if needed.

pKVM

- `kvm_arch_gmem_invalidate()` should check for a pending request from a guest before permitting conversion to shared
  - ~~This prevents breaking confidentiality~~
    - No loss in confidentiality even if we didn't check for a pending request since if EL2 thinks the page is private, going ahead with the conversion still won't allow memory to be read. The check for pending request just ensures the host doesn't get to think of it as shared, map it and then end up crashing itself by accessing a page EL2 thinks is private.
  - Fuad: Quentin: No need to go to EL2 in guest_memfd. By the time the conversion request is made, EL2 would already think of the page as shared.
  - This doesn't matter for TDX/SNP because encryption/built-in zeroing retains confidentiality?
- Quentin: shared to private: need some way to ask userspace (some hypercall - arch specific)
  - EXIT_MEMORY_FAULT
- Ordering - must `invalidate()` before applying zeroing so that for to_shared, the conversion process itself doesn't cause a host crash
  - Fuad: Slight asymmetry:
    - To_shared request from guest: switch to shared on request
    - To private request from guest: switch after conversion is done
  - TDX doesn't zero anyway so the invalidation (implicit in unmapping) just has to happen before the ioctl completes.
  - Michael: SNP requires zeroing in gmem

ARM CCA

- Does the above framework (invalidate and prepare) work for ARM CCA?