Pushing guest_memfd conversions along

For 2026-02-19 guest_memfd bi-weekly upstream call

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

Private MMIO wrt guest_memfd conversion ioctl

Context

- Alexey is working on TEE-IO [1], where the fd of a VFIO MMIO is a dmabuf fd
- The KVM fault path reads attributes from guest_memfd
    - (should not, need to teach KVM to work with non-guest_memfd provider of private memory)

[1] https://lore.kernel.org/all/07836b1d-d0d8-40f2-8f7b-7805beca31d0@amd.com/

What I want to take away from this discussion

- Do people think it will affect conversion uAPI (the ioctl being sent to guest_memfd)?

Questions

- Any ideas for how conversion would look like for private MMIO?
- Is conversion necessary? Is conversion allowed for device addresses?
- Would it look like sending a conversion ioctl to the dmabuf fd?

Memory preservation during conversion

Background

- guest_memfd conversions series RFC v2 [1] posted
- Currently, memory might be preserved depending on whether the CoCo vendor does zeroing

- KVM’s ABI cannot let behavior be undefined, or be based on (CoCo) vendor
- All decisions that affect guest data must be made by userspace.

[1] https://lore.kernel.org/all/CAEvNRgFMNywpDRr+WeNsVj=MnsbhZp9H3j0QRDo_eOP+kGCNJw@mail.gmail.com/

Alternatives

- Kconfig
  - Bad: not userspace determined
- KVM module param
  - Bad: VMs on the same host may want different settings
- Guest_memfd creation time flag
  - Bad: different conversion requests might want different settings
- Ioctl flag
  - Let’s discuss!

Implementation: PRESERVE_CONTENTS

- Call arch function if PRESERVE_CONTENTS

- SW_PROTECTED_VM will do nothing (no clearing), apply software zeroing otherwise.
- TDX’s preserve function would return -EOPNOTSUPP
- pKVM: require PRESERVE_CONTENTS, or let it be a lost optimization? (is it just a lost optimization?)

Next steps

Next steps

- Ackerley: send RFC v3