Contribution ID: **263** Type: **not specified**

VFIO: Very Frightening I/O? Taming Wild Guests and their PCIe Config-Space Abuse

Cloud workloads with strict performance needs (AI, HPC, large-scale data processing) frequently use PCIe device passthrough (e.g., via VFIO in Linux/KVM) to reduce latency and improve bandwidth. While effective for performance, this approach also exposes low-level device configuration interfaces directly to guest workloads, which may be malicious or running untrusted software.

In our experiments across multiple device types and vendors, we observed that legal but unexpected writes to the PCIe configuration space of a passthrough device can trigger PCIe errors that lead to host-level failures, posing a risk to system reliability, availability, and serviceability (RAS). While it is well known that PCIe errors can cause host-level failures, our observations are noteworthy for two main reasons:

1. Tenants or guest workloads can trigger host-level failures simply by having direct access to a PCIe device interface.
2. The PCIe configuration space offsets that caused these failures often correspond to unassigned registers, i.e., registers that do not belong to any exposed capability. Our discussions with device vendors indicated that these unassigned registers belong to hidden, vendor-specific register sets.

To begin addressing this issue, we prototyped a VFIO kernel patch that blocks guest accesses to unassigned regions in the PCIe configuration space. The patch allows two coarse-grained policies that can be enforced through module parameters: 1) Reads and writes to unassigned regions can be independently blocked (e.g., block only writes, block only reads, or block both). 2) Specific devices can be whitelisted to allow accesses to unassigned regions if required.

Looking forward, we propose a more fine-grained model analogous to how the IOMMU validates IOVA mappings. In this design, VFIO would maintain a table of "valid"config-space regions per device, supplied by the device driver (most realistic) or potentially extracted from firmware/device descriptors (longer-term). VFIO would consult this table on each guest config access, permitting safe operations while blocking or emulating unsafe ones.

In this talk, we aim to discuss:
- The implications of exposing PCIe configuration space in passthrough setups.
- Whether VFIO (or related subsystems) should adopt a stricter access model.
- Potential directions for long-term solutions, involving kernel, firmware, and hardware changes.

Our goal is to engage with the community to shape a path forward that preserves both the performance benefits of passthrough and the robustness required for large-scale, multi-tenant deployments.

Primary authors: RAJAPAKSHA, Chathura (Boston University); KOTESHWARA, Sandhya (IBM Research); DAS, Bandan (Red Hat); MOHAN, Apoorve (IBM Research); FRANKE, Hubertus (IBM Research); JOSHI, Ajay (Boston University); EGELE, Manuel (Boston University)

Presenters: RAJAPAKSHA, Chathura (Boston University); KOTESHWARA, Sandhya (IBM Research); DAS, Bandan (Red Hat); MOHAN, Apoorve (IBM Research)

Session Classification: VFIO/IOMMU/PCI MC

Track Classification: VFIO/IOMMU/PCI MC