s390 (IBM Z) Ultravisor and Protected VMs

Summary

Protected virtual machines (PVM) are KVM VMs that do not allow KVM to access VM state like guest memory or guest registers. Instead, the PVMs are mostly managed by a new entity called Ultravisor (UV). The UV provides an API that can be used by PVMs and KVM to request management actions.

Each guest starts in non-protected mode and then may make a request to transition into protected mode. On transition, KVM registers the guest and its VCPUs with the Ultravisor and prepares everything for running it.

The Ultravisor will secure and decrypt the guest's boot memory (i.e. kernel/initrd). It will safeguard state changes like VCPU starts/stops and injected interrupts while the guest is running.

As access to the guest's state, such as the SIE state description, is normally needed to be able to run a VM, some changes have been made in the behavior of the SIE instruction. A new format 4 state description has been introduced, where some fields have different meanings for a PVM. SIE exits are minimized as much as possible to improve speed and reduce exposed guest state.

Interrupt injection

Interrupt injection is safeguarded by the Ultravisor. As KVM doesn't have access to the VCPUs' lowcores, injection is handled via the format 4 state description.

Machine check, external, IO and restart interruptions each can be injected on SIE entry via a bit in the interrupt injection control field (offset 0x54). If the guest cpu is not enabled for the interrupt at the time of injection, a validity interception is recognized. The format 4 state description contains fields in the interception data block where data associated with the interrupt can be transported.

Program and Service Call exceptions have another layer of safeguarding; they can only be injected for instructions that have been intercepted into KVM. The exceptions need to be a valid outcome of an instruction emulation by KVM, e.g. we can never inject a addressing exception as they are reported by SIE since KVM has no access to the guest memory.

Mask notification interceptions

KVM cannot intercept lctl(g) and lpsw(e) anymore in order to be notified when a PVM enables a certain class of interrupt. As a replacement, two new interception codes have been introduced: One indicating that the contents of CRs 0, 6, or 14 have been changed, indicating different interruption subclasses; and one indicating that PSW bit 13 has been changed, indicating that a machine check intervention was requested and those are now enabled.

Instruction emulation

With the format 4 state description for PVMs, the SIE instruction already interprets more instructions than it does with format 2. It is not able to interpret every instruction, but needs to hand some tasks to KVM; therefore, the SIE and the ultravisor safeguard emulation inputs and outputs.

The control structures associated with SIE provide the Secure Instruction Data Area (SIDA), the Interception Parameters (IP) and the Secure Interception General Register Save Area. Guest GRs and most of the instruction data, such as I/O data structures, are filtered. Instruction data is copied to and from the SIDA when needed. Guest GRs are put into / retrieved from the Secure Interception General Register Save Area.

Only GR values needed to emulate an instruction will be copied into this save area and the real register numbers will be hidden.

The Interception Parameters state description field still contains the bytes of the instruction text, but with pre-set register values instead of the actual ones. I.e. each instruction always uses the same instruction text, in order not to leak guest instruction text. This also implies that the register content that a guest had in r<n> may be in r<m> from the hypervisor's point of view.

The Secure Instruction Data Area contains instruction storage data. Instruction data, i.e. data being referenced by an instruction like the SCCB for sclp, is moved via the SIDA. When an instruction is intercepted, the SIE will only allow data and program interrupts for this instruction to be moved to the guest via the two data areas discussed before. Other data is either ignored or results in validity interceptions.

Instruction emulation interceptions

There are two types of SIE secure instruction intercepts: the normal and the notification type. Normal secure instruction intercepts will make the guest pending for instruction completion of the intercepted instruction type, i.e. on SIE entry it is attempted to complete emulation of the instruction with the data provided by KVM. That might be a program exception or instruction completion.

The notification type intercepts inform KVM about guest environment changes due to guest instruction interpretation. Such an interception is recognized, for example, for the store prefix instruction to provide the new lowcore location. On SIE reentry, any

KVM data in the data areas is ignored and execution continues as if the guest instruction had completed. For that reason KVM is not allowed to inject a program interrupt.

Links

KVM Forum 2019 presentation