Google

Deploying Confidential VMs via Linux

Marc Orr <marcorr@google.com>, Sep 21, 2021

Executive Summary

- Great progress enabling Linux-based confidential VMs (CVM)
  - SEV: available on Google Compute Engine (GCE) today
  - SEV-ES: not currently available on GCE, but supported by Linux
- Deploying code highlighted gaps in both functionality and processes
- Merging code is not enough to run reliably on a public cloud

Challenges in deploying CVMs via Linux

- Guest image support
- Testing CVMs
- Host-side kexec and kdump support
- Summary

Guest kernel: Images

Two challenges (*same solution?*):

1. Fixing bugs for currently supported Linux kernel features
2. Getting new CVM functionality into guest images

Guest kernels lag Linux tip (by a lot!)

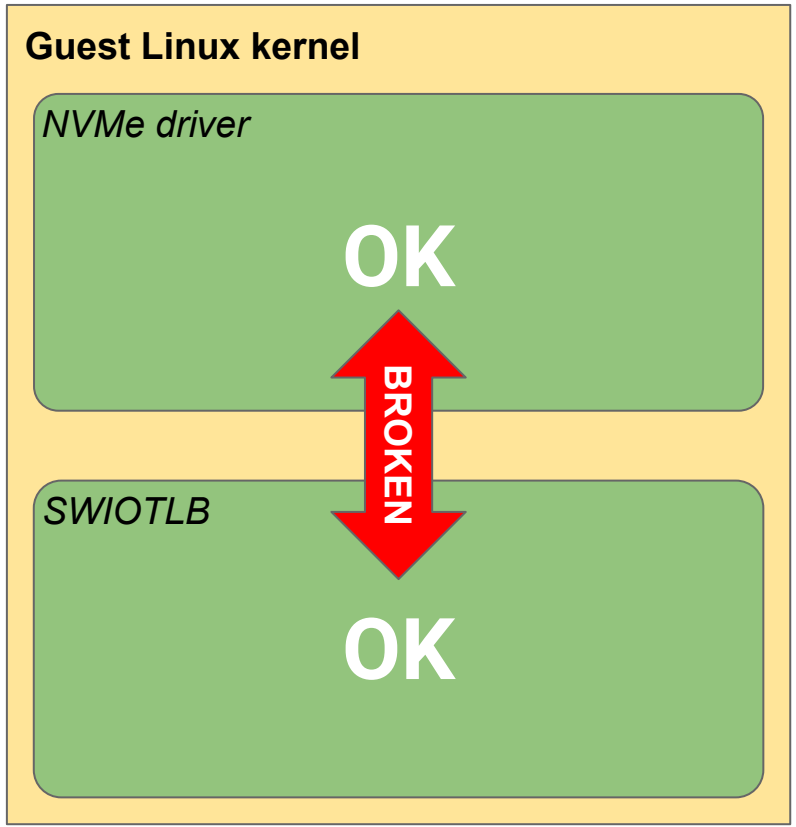
- Current Linux tip: 5.15

GCE Guest Linux Distro	Linux kernel version
CentOS 8	4.18.0
COS 97	5.10.61
RHEL 8.4	4.18.0
SLES 15.2	5.3.18
Ubuntu 18.04.5 LTS	5.4.0
Ubuntu 20.04.2 LTS	5.8.0

Problem #1: Guest-side bugs

- CVM IO goes through bounce buffer
  - Bounce private (encrypted) pages through SWIOTLB's shared (plaintext)
- Two guest bugs exposed by routing NVMe DMA through SWIOTLB
- 1/2 fixes accepted into LTS 5.4

Bugs exposed using components in new ways are not necessarily candidates for LTS kernels



Problem #2: New guest kernel functionality

- SEV-ES, SEV-SNP, TDX all require new guest kernel code
- Patches will not show up in guest images based on old kernels

Thoughts?

How to get CVM bug fixes into guest images used by customers?

How to get distros to release guest images with support for new CVM offerings?

Discussion ideas:

- Can we partner with distros to establish CVM images?
  - Forked from existing guest images
- Is it possible to support a bleeding edge distro so customers can use new features like CVM?
  - Is this too much toil for customers?
- Can we partner with distros to test CVMs, to prevent regressions?

Challenges in deploying CVMs via Linux

- Guest image support
- Testing CVMs
- Host-side kexec and kdump support
- Summary

Testing

- Deploying new functionality without tests is a non-starter
- Writing tests for new code helps to deploy it and maintain it

Example KVM selftest for SEV (not written)

KVM: SVM: Call SEV Guest Decommission if ASID binding fails

Send SEV_CMD_DECOMMISSION command to PSP firmware if ASID binding fails.

- If a failure happens after a successful LAUNCH_START command, a decommission command should be executed.
- Otherwise, guest context will be unfreed inside the AMD SP.
- After the firmware will not have memory to allocate more SEV guest context, LAUNCH_START command will begin to fail with SEV_RET_RESOURCE_LIMIT error.

```
/* 1. Exhaust all SEV ASIDs */
for (i = 0; i < MAX_ASID; i++) {
     vm_fds[i] = sev_vm_create();
     ASSERT(vm_fds[i] != -1);
}

/* 2. Exhaust all PSP contexts */
for (; i < MAX_CONTEXT; i++) {
     vm_fds[i] = sev_vm_create();
     ASSERT(vm_fds[i] == -1);
}

/* 3. Delete all VMs */
for (i = 0; i < MAX_ASID; i++)
     kvm_vm_free(vm_fds[i]);

/* 4. Check that SEV VMs can be created */
ASSERT(sev_vm_create() != -1);
```

Thoughts?

How to prioritize KVM-Unit-tests and KVM selftests for SEV?

How to effectively test guest support?

What else should we be testing?

Are there other open-source test frameworks we should be leveraging and extending for CVMs?

Discussion ideas:

- KVM-Unit-tests for SEV-ES posted to KVM mailing list
  - https://patchwork.kernel.org/project/kvm/list/?series=538063
- selftests for SEV starting to appear on KVM mailing list
  - https://patchwork.kernel.org/project/kvm/list/?series=546789
- How to test other software components (e.g., UEFI, guest OS)?

Challenges in deploying CVMs via Linux

- Guest image support
- Testing CVMs
- Host-side kexec and kdump support
- Summary

Host kernel: kexec and kdump

- kexec and kdump are baked into public cloud infrastructure
- We cannot deploy features that break kexec or kdump
- More generally, we cannot deploy features that break pre-existing host-side functionality

kexec OK on hosts configured to run SEV-ES VMs

- kexec became flaky when PSP was updated to manage SEV-ES guests

[BUG] crypto: ccp: random crashes after kexec on AMD with PSP since commit 97f9ac3d
On several AMD systems, we see random crashes after kexec, during the boot of the new system (typically 1 out of 5 boots ends up with a crash).

According to git bisect, the regression was introduced by commit 97f9ac3d ("crypto: ccp - Add support for SEV-ES to the PSP driver"), included since 5.8-rc1. 5.14-rc3 is still affected.

Removing the 'ccp' module before kexec makes the problem disappear.

It is worth noting that there was prior work about getting kexec to work with PSP/SEV (commit f8903b3e, "crypto: ccp - fix the SEV probe in kexec boot path").

I can help test patches if needed. If this gets fixed, it would be fantastic if the fix was backported to 5.10.

Here are some crash logs. As you can see, the kernel seems to crash at various places.

- kexec fixed for SEV-ES

[PATCH] crypto: ccp: shutdown SEV firmware on kexec
The commit 97f9ac3db6612 ("crypto: ccp - Add support for SEV-ES to the PSP driver") added support to allocate Trusted Memory Region (TMR) used during the SEV-ES firmware initialization. The TMR gets locked during the firmware initialization and unlocked during the shutdown. While the TMR is locked, access to it is disallowed.

Currently, the CCP driver does not shutdown the firmware during the kexec reboot, leaving the TMR memory locked.

Register a callback to shutdown the SEV firmware on the kexec boot.

Fixes: 97f9ac3db6612 ("crypto: ccp - Add support for SEV-ES to the PSP driver")
Reported-by: Lucas Nussbaum <lucas.nussbaum@inria.fr>
Tested-by: Lucas Nussbaum <lucas.nussbaum@inria.fr>

Thoughts?

Opening bid: kexec support should be mandatory for host-side CVM enablement (e.g., SNP, TDX)

What other host-side kernel features do we need to test when hosts are configured to run CVMs?

Discussion ideas:

- Can we persist the RMP across kexec?
- What about making kexec'ing into kernel with no knowledge of the SNP -- is it possible?
  - Is it useful?

Challenges in deploying CVMs via Linux

- Guest image support
- Testing CVM
- Host-side kexec and kdump support
- Summary

Summary

- Running CVMs on a public cloud is non-trivial
- There exists a gap between merging code into Linux and deploying it to a public cloud
- We need better guest distro support
  - To adopt new CVM features
  - To accept guest kernel bug fixes for CVMs
- Writing tests for new code helps to deploy the code
- kexec and kdump must work on the host kernel