Memory protection in Android using KVM

Quentin Perret <qperret@google.com>
Why?
Exception levels on arm64, architecturally

- **EL0**: App, App, App, App
- **EL1**: Kernel, Kernel
- **EL2**: Hypervisor
- **EL3**: Firmware / Secure monitor

- **Normal world**: EL0, EL1, EL2, EL3
- **Secure world**: Trusted App, Trusted App, Trusted App *, Trusted App *
  - Trusted OS, Trusted OS *
  - Trusted Partition Manager / Hypervisor *

* From Arm v8.4A
Exception levels on arm64, in Android

![Diagram showing exception levels on arm64 in Android]

- **EL0**: App
- **EL1**: Android Kernel (GKI)
- **EL2**: Hypervisor
- **EL3**: Firmware / Secure monitor

### Normal world
- App
- App
- App

### Secure world
- Trusted App
- Trusted App
- Trusted App
- Trusted OS
- Trusted Partition Manager / Hypervisor

* From Arm v8.4A
Exception levels on arm64, by privilege

Normal world

EL0: App
EL1: Android Kernel (GKI)
EL2: Hypervisor

Secure world

sEL0: Trusted App
sEL1: Trusted OS
sEL2: Trusted Partition Manager / Hypervisor *
EL3: Firmware / Secure monitor

Increasing privilege
Exception levels on arm64, in Android today, by privilege

- **Normal world**
  - EL0: App, App
  - EL1: Android Kernel (GKI)
  - EL2: Hypervisor

- **Secure world**
  - sEL0: Trusted App, Trusted App
  - sEL1: Trusted OS
  - sEL2: Trusted Partition Manager / Hypervisor *
  - EL3: Firmware / Secure monitor

 Increasing privilege

- DRM, crypto, ...
- Third party OSes
- Opaque blobs
Exception levels on arm64, in Android today, by privilege

- Normal world
  - EL0: App, App
  - EL1: Android Kernel (GKI)
  - EL2: Hypervisor

- Secure world
  - sEL0: Trusted App, Trusted App
  - sEL1: Trusted OS
  - sEL2: Trusted Partition Manager / Hypervisor *
  - EL3: Firmware / Secure monitor

Increasing privilege:
- DRM, crypto, ...
- Third party OSes
- Opaque blobs
What?
What do we need?

We need a hypervisor that is:

1. open source
2. easy to ship and update
3. supports guest memory protection
4. trustworthy

KVM as part of GKI is a very good fit with the right extensions.
How ?
Virtual memory on arm64

EL0
Userspace

EL1
Kernel

MMU

Memory

VA
PA
PA
Stage 2 memory translation

EL0  Userspace  VA  IPA  PA  Memory
EL1  Kernel  VA  IPA
EL2  Hypervisor
KVM port on armv8.0A (nVHE)

- EL0: Host Userspace
- EL1: Host Kernel
- EL2: KVM "hyp" code
- Guest userspace
- Guest kernel

- ioctl(VCPU_RUN)
- kvm_call_hyp(fn, ...)
- eret

- Switches context
- Installs host-provided stage 2 page-table
KVM port on armv8.0A (nVHE)

- **EL0**: Host Userspace, Guest userspace
- **EL1**: Host Kernel, Guest kernel
- **EL2**: KVM “hyp” code

- Switches context
- Uninstalls stage 2 page-table
Proposal

- **Install a stage 2 page-table over the host**
  - Elevate stage 2 page-table management at EL2
  - Requires basic mm at EL2

- **Elevate hyp stage 1 page-table management at EL2**

- **Clean split between host and hyp code**
  - Separate ELF section for hyp code: `.hyp.text`
  - “Proper” HVC interface between host and hyp

- **Guest memory and state unaccessible** from host kernel and VMM
  - Message passing to setup shared memory regions (PSA-FFA + Virtio)
  - Minimal guest bootloader, which verifies guest payload signature

- **Formal verification** of the code running at EL2
Get the code

The repo is public:

https://android-kvm.googlesource.com/linux/

And there are patches on the list:

https://lore.kernel.org/kvmarm/20200730132519.48787-1-dbrazdil@google.com/
https://lore.kernel.org/kvmarm/20200722164424.42225-1-dbrazdil@google.com/
https://lore.kernel.org/kvmarm/20200721094445.82184-1-dbrazdil@google.com/
https://lore.kernel.org/kvmarm/20200625131420.71444-1-dbrazdil@google.com/
https://lore.kernel.org/kvmarm/20200820103446.959000-1-ascull@google.com/
https://lore.kernel.org/kvmarm/20200730151823.1414808-1-ascull@google.com/
https://lore.kernel.org/kvmarm/20200713210505.2959828-1-ascull@google.com/
https://lore.kernel.org/kvmarm/20200327143941.195626-1-ascull@google.com/
https://lore.kernel.org/kvmarm/20200505154520.194120-1-tabba@google.com/
https://lore.kernel.org/kvmarm/20200818132818.16065-1-will@kernel.org/

...
Thanks.

android-kvm@google.com