KVM to sandbox firmwares from the Kernel
or: How I learned to stop worrying and love EFI

Florent Revest
February 4th 2018
• INSA Toulouse student
• Embedded soft. consultant
• Creator of AsteroidOS
• Ex-Intern at ARM
Table Of Contents

1. KVM
2. Internal KVM
3. EFI Runtime Services
4. EFI Sandboxing
KVM
General Overview of KVM

• Host kernel module
• Assisted by the CPU
• Supports x86, ARM, PPC...
• Used by QEMU, kvmtool...

KVM to sandbox firmwares from the Kernel - Florent Revest
KVM APIs

- Provides an ioctl API
- For VM & VCPU creation
- Userspace addr mapping

KVM to sandbox firmwares from the Kernel - Florent Revest
Internal KVM
THE IDEA BEHIND INTERNAL KVM

- Cross platform code isolation
- For security and stability
- Of foreign or critical code
- Generic enough to be used by various subsystems

KVM to sandbox firmwares from the Kernel - Florent Revest
Problems Encountered

- Lack of functions
- Many copy_from_user()
- Preemption control
- Userspace addr mapping
- No hypercalls routing
kvm = kvm_create_internal_vm(0);
kvm_vm_create_vcpu(kvm, 0);
vcpu = kvm_get_vcpu_by_id(kvm, 0);
kvm_vcpu_preferred_target(&init);
kvm_arch_vcpu_ioctl_vcpu_init(vcpu, &init);
EFI Runtime Services
General Overview

- Part of the UEFI standard
- Handlers for EFI variables, RTC, etc...

mjg59 | More ways for firmware to screw you
https://mjg59.dreamwidth.org/11235.html
UEFI defines two types of code - boot services and runtime services. While runtime services code and data must be preserved by the OS, in theory boot services ...

[PDF] UEFI is not your enemy - LinuxTag
www.linuxtag.org/2014/.../Leif_Lindholm_-_UEFI_is_not_your_enemy.e1454.pdf
shrugged off as BIOS bugs now referred to as "UEFI secure boot bollox on a slippery ..... GRUB, Linux UEFI runtime services support, kernel UEFI stub support ...
Problems With EFI Runtime Services

- Proprietary code running with the privileges of the kernel
- Can modify system registers
- "Jump and pray for the best"
EFI Sandboxing
• Start KVM before EFI Drivers
• Simple phys. & virt. memory
• Args via VCPU regs & buffers
• Return with hypercalls
Results

• No systems register mess
• Clean exceptions handling
• Tested on SoftIron OverDrive
• Tested on an ARMv8.3 model
• RFC posted on LKML, reviewed
Limitations

- EL3 handlers dependencies
- DMA misconfiguration
Thank you! Questions?

Oh and by the way, I am searching for my next summer internship!