Adding Third-Party Hypervisor to Android Virtualization Framework

Elliot Berman <eberman@quicinc.com> (Senior Software Engineer)
Prakruthi Deepak Heragu <pheragu@quicinc.com> (Senior Software Engineer)

Qualcomm Innovation Center, Inc.
What is AVF?

See David Brazdil’s talk from LPC 2022: lpc.events/event/16/contributions/1330/
What is Gunyah?

• Hypervisor solution implemented by Qualcomm Technologies, Inc.

• EL2 Hypervisor is small microkernel

• “Resource Manager” VM implements policy for EL2 & runs isolated from other VMs

• Key design differences from other hypervisors:
  • Designed for VM isolation
    • De-privileged VMs, including the primary VM
    • Memory, IO space isolation
    • Interrupt assignment
    • EL2-based scheduling
  • Meet automotive use cases (MISRA)
  • Performance optimized for mobile/auto/IoT use cases

Hypervisor Access Control

• Multi-OS Support
• Smaller Attack surface
Hypervisor Requirements
As of Android 14! Subject to change in future Android Releases

• Dynamically allocate virtual machines
  • AVF does not require guests to be Linux-based

• Handle MMIO from vCPU to support virtio

• Shutdown support (PSCI for ARM® ecosystems)

• Essential arch-specific devices (e.g., interrupt controller, timers)

• Support Protected VMs
  • Memory Isolation
  • Authenticated and have anti-rollback (Android Verified Boot)
  • Attestation that they were loaded by a trusted boot chain
Protected VM Memory Isolation

View of Host memory before launching the VM

View of Host memory after launching the VM

Host Memory

Host Memory

Guest Stage-2

Guest Private Memory

Guest/Host Shared Memory
New additions to support Gunyah in AVF

### CrosVM
- New Gunyah hypervisor crate
- Use /dev/gunyah ioctls
- Detection/selection of Gunyah as hypervisor
- **16 changes to crosvm**
  - 9 are “generic” changes to prepare for Gunyah-specific changes

### Linux
- kernel.org first for all AVF work
- /dev/gunyah device
- Driver abstracts all communication to Gunyah
- Virtual machine memory management
- **52 changes to android14-6.1**
  - We kept up with reviewer feedback from LKML
Qualcomm adopting AVF
Enhancements in Gunyah

- Host-only device emulation
- Stolen time accounting
- Support to run pvmfw before VM runs
Debugging a Protected VM

• VM instances can become debuggable if debug policy exists

• Debug policy implemented as a devicetree overlay
  • Enables adbd on Microdroid
  • Enables console on the Guest VM
  • Extensible for more properties in future

• Debug policy devicetree copied into pvmfw and host devicetree
  • Changes in Android Bootloader to load the debug policy
What’s Next for AVF + Gunyah?

• Guest memory upstreaming
  • Tracking memory that has been isolated from host
  • Demand paging

• kdump support (de-isolating memory)

• VM communication with firmware

• Challenges
  • Device Sanitization during VM or VMM crash scenario
References

• LVC21F-224 Gunyah Open Source Hypervisor
  • https://www.youtube.com/watch?v=EM4lTICBGDk

• Source Code for Gunyah
  • https://github.com/quic/gunyah-hypervisor

• CrosVM changes
  • https://chromium.googlesource.com/crosvm/crosvm/+refs/heads/main/hypervisor/src/gunyah/

• Kernel.org changes
  • https://lore.kernel.org/lkml/20230613172054.3959700-1-quic_eberman@quicinc.com/
Thank you