Summary

• Observations
  ◦ STR-5 (Virtio RPMB) – using QEMU for backend development
  ◦ STR-26 (Virtio I2C) – Using C language for backend development?

• Proposal/Ask
  ◦ Adopt Rust-VMM as default platform for virtio backends
  ◦ Improve Rust-VMM for adoption in ARM64 production environment
Virtio - KVM vs Qualcomm Hypervisor

Virtio Backend has NO access to guest OS memory.

No ready-to-use backend drivers.
Virtio Usage

Primary VM

- VMM-lite (virtio device)
  CrosVM based

- Virtio BackEnd Kernel Module

- VM Loader (remote-proc)

Secondary VM

(SVM private space)

- Virtio FrontEnd Kernel Module

(SVM shared space)

- Block1 Ring

- Block2 Ring
  - Bounce buffers

- Kernel-space VM loader (remote_proc/PIL)
- VMM required to host only virtio device backends
Backend Selection

• Choices Evaluated
  ◦ LKVM – not production ready
  ◦ Qemu – Complexity
  ◦ ACRN
  ◦ Rust-VMM
  ◦ CrosVM

• Why we went with CrosVM?
  ◦ Promise of RUST language to avoid memory-related bugs
  ◦ Adoption of CrosVM in Android

• Future Plans
  ◦ Evaluate Rust-VMM and adopt in scenarios where CrosVM may not be feasible
Rust Experience So far

• **Takeaways**
  ◦ Modified CrosVM undergoing product adoption
  ◦ Has been relatively “easy” to make required changes

• **Observations**
  ◦ Initial learning curve - ~1 month
  ◦ Extensive examples/documentation on Internet helped make required changes
  ◦ Android specific build mechanism for Rust
  ◦ Use of traits – came in handy to override the default implementation of some functions (roughly accomplished with function pointers in C)
  ◦ “auto” generated code – when variables go out of scope (closing file descriptors for example)
  ◦ Good reliability – no language related issues found (so far!)
  ◦ IDE – vim integration did not work (for me)

• **Unexplored**
  ◦ Debugging via GDB
  ◦ Profiling
Why rust-vmm?

- Share common virtualization code between CrosVM & Firecracker
- Create custom VMMs
- Modularity & testing
Rust-vmm in Production

- Firecracker
- Cloud Hypervisor
- Alibaba Cloud Sandbox
- Enarx
- libkrun
- Nydus: Dragonfly Container Image Service
- ...
Rust-VMM

- Vm-memory
- Virtio-devices
- Qualcomm Hypervisor bindings
- KVM-bindings

Vmm Glue

Block
Net
Rust-vmm - todo

• Aarch64 support
• Promote required crates to “production-ready” level
• Qualcomm hypervisor bindings
BACKUP
Status 2020 - New Development

- vfio-ioctl
- vhost-user-backend
- vm-allocator

- vhost
- vm-device
- vm-virtio
- vmm-reference

- event-manager
- linux-loader
- virtio-bindings
- kvm-bindings
- vm-superio
- vm-memory
- vmm-sys-util
- vfio-bindings
- kvm-ioctl

Milestone 1
Empty Crate

Milestone 2
Crate in Dev

Publish crates.io
CSS Component

- Security bugs since Firefox started: 69
- Rust would have prevented: 51

73.9%

Implications of Rewriting a Browser Component in Rust
By Diane Hostelt, 2019-02-28

https://www.youtube.com/watch?v=A3AdN7U24iU
Chrome: 70% of all security bugs are memory safety issues

Microsoft: 70 percent of all security bugs are memory safety issues

We closely study the root cause trends of vulnerabilities & search for patterns