Android Virtualization Framework
Protected computing for the next generation use cases

David Brazdil (he/him)
Technical Lead
@dbrazdil

Serban Constantinescu (he/him)
Product Manager
@serbanc
Android defense mechanisms

Google Play Store
Malware and security scanning for Android applications.

Android Application Sandbox
Isolates Android applications and their resources from each other.

Android Kernel
Enforces the application sandbox and process isolation and security policies.

Isolated Execution Environment
Isolates security critical payloads even in the event of a compromised Android Kernel.

Fragmented, constrained APIs, limited security updates and no mutual distrust.

Too big of an attack surface for privacy-sensitive use cases.
A standard deployment across the ecosystem

More use cases require an Isolated Execution Environment

- Personal identifiable information
  - Biometrics information and algorithms.

- Intellectual Property
  - DRM, Machine Learning and IP protection.

- Digital assets
  - Digital keys, crypto and more.

- Healthcare
  - Medical and digital fitness data.

- Ambient information
  - Confidential or personal information that should never leave the device.

- Defense in depth
  - Kernel protection and malware detection.

These use cases need isolation even in the event of a kernel vulnerability (think Dirty Pipe).

The TrustZone TEE is too privileged and fragmented to use. Deploying there would further increase the vulnerable TCB.
The Execution Environment needed
to enable Protected Computing on Android

- **Isolated**
  from the kernel’s attack surface and other Protected Virtual Machines.

- **Updatable**
  using the same containers and update technologies as Android.

- **Least privilege**
  mutually distrusted and isolated even in the event of an exploit.
Android Virtualization Framework
android13-5.x branches:
we’ve been very busy!

Key pKVM hypervisor features available today:

- VM state and management isolated from the host
- Guest memory isolation
  - Including some IOMMU support (S2MPU)
- Services exposed as hypercalls to the guest payload
  - Memory sharing and virtio using bounce buffers
  - MMIO guard
  - TRNG proxied to secure world
- pVM firmware loading
- Non-protected guests for debug visibility

Actively working on upstreaming all of these features!
See our talks at KVM Forum
Architecture

Android
- Java API
- virtualization service
- crosvm
- libbinder

Microdroid
- Native API
- userspace tools
- microdroid_manager
- libbinder

Linux Kernel

Hypervisor (pKVM)

EL0
- Protected VM

EL1

EL2
Programming Model

- Using APIs, you create a protected VM and run a native shared library in your APK there
- The library implements a Binder service
- Android app connects over Binder to send commands and get results

Protected VM's memory is protected from host VM (and vice versa)
**Secret Provisioning Using DICE**

- Each pVM has its own secret key, not available to Android.
- The per-pVM secret is not a random number, nor kept in a secure key store.
- It is a function of:
  1. Measurements of the software that defines the behavior of the pVM*
  2. Unique Device Secret (UDS)
- Provisioned during the pVM boot.

*From bootloader, hypervisor, up to the application*
Documentation

https://source.android.com/devices/virtualization

android-kvm+partner@google.com
Questions

- How do you plan to use the Android Virtualization Framework?
- What use cases do you deploy at EL2/TZ today?
Thank you!
Today, more cyber attacks than ever are happening on a broader, global scale. The targets of these attacks are ... but also individuals.
Attacks are moving to more privileged layers like the TEE.
Android Virtualization Framework

- Hypervisor, Protected Virtual Machines, and Framework APIs that enable Protected Computing in Android.

- Upstream Protected KVM or vendor specific.
- Isolated from Android, other VMs and DMA devices.

Integrated in Android as a first-class primitive; standard and developer friendly.
**Key Components (1/3)**

**Hypervisor**
- Must isolate VM memory from others, even from the host; enforced with stage-2 page tables and IOMMUs
- Reference implementation: KVM/arm in protected mode (pKVM)

**Generic Kernel Image (GKI)**
- pKVM distributed as part of GKI, enabled when kernel booted in EL2
- Exposes /dev/kvm as the control interface
- Host GKI remains in charge of scheduling
- Guests run the same GKI kernel booted in EL1
Key Components (2/3)

**virtualization service**
- System service managing lifecycle of VMs
- Actual creation of VM is delegated to crosvm
- Accessed via Java API (optional library)

**crosvm**
- Virtual machine monitor written in Rust
- Hypervisor and PV device backends
- Resource management (memory, vCPUs)

**pvmfw**
- First code that runs in a protected VM
- Verifies the payload, derives per-VM secret
Key Components (3/3)

Microdroid
- Lightweight headless Android for pVM

microdroid_manager
- Manages application inside the VM
- Securely mounts APK/APEXes from host
- Provides access to per-VM secret

libbinder
- Extended to work over vsoc
- Primary means of inter-VM communication

Native API
- A subset of NDK provided to application
  - libc/m/dl, no libandroid.so
Packaging

- boot.img
  - GKI
  - Protected KVM
- pvmfw.img (new partition)
  - pvmfw
- system_ext.img
- com.android.virt.apex

- microdroid_system.img
  - Native API
  - userspace tools
- microdroid_boot.img
  - GKI (guest)
  - Java API
  - virtualization service
  - crosvm
Lifecycle of a pVM

Once created, a pVM instance can be repeatedly started and stopped, as long as the software running inside the pVM remains the same. Future changes will allow to update forward without invalidating the instance.
Secret Provisioning Using DICE

- Each stage in boot sequence derives a secret for the next stage

```
<table>
<thead>
<tr>
<th>Instance disk</th>
<th>Stage N</th>
<th>Key Derivation Function</th>
<th>salt</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>save/</td>
<td>compare</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Unique Device Secret
- Android Bootloader
- pvmfw secret
- pvmfw
- OS secret
- Microdroid
- per-VM secret
- Application