



Laying a Secure Foundation for Mobile Devices

Stephen Smalley
Trusted Systems Research
National Security Agency



Trusted Systems Research

- Conduct and sponsor research to provide information assurance for national security systems.
- Enabling safe operation in risky or compromised environments.
- Research into cryptographic algorithms and protocols, system analysis and design methods, trust mechanisms, and systems behavior.
- Creators of SE Linux, Xen Security Modules, Linux Kernel Integrity Monitor, and SE for Android.



Our Motivation

- Increasing demand to use mobile devices.
 - NSA Mobility Program
- Desire to use commodity solutions.
 - NSA Commercial Solutions for Classified (CSfC)
- Risks posed by currently available solutions.
 - Exploitation over wireless, radio, NFC, ...
 - Data Leakage
 - Application privilege escalation



Why It Matters for Everyone

- Explosion in mobile malware.
 - Rapid growth, increasing sophistication.
- Increasing market drivers for mobile device attacks.
 - Payment, banking, remote control.
 - BYOD trend for corporate/enterprise use.
 - Increasing use of mobile platforms in non-traditional venues, including safety-critical.
- It isn't just a problem for government use.



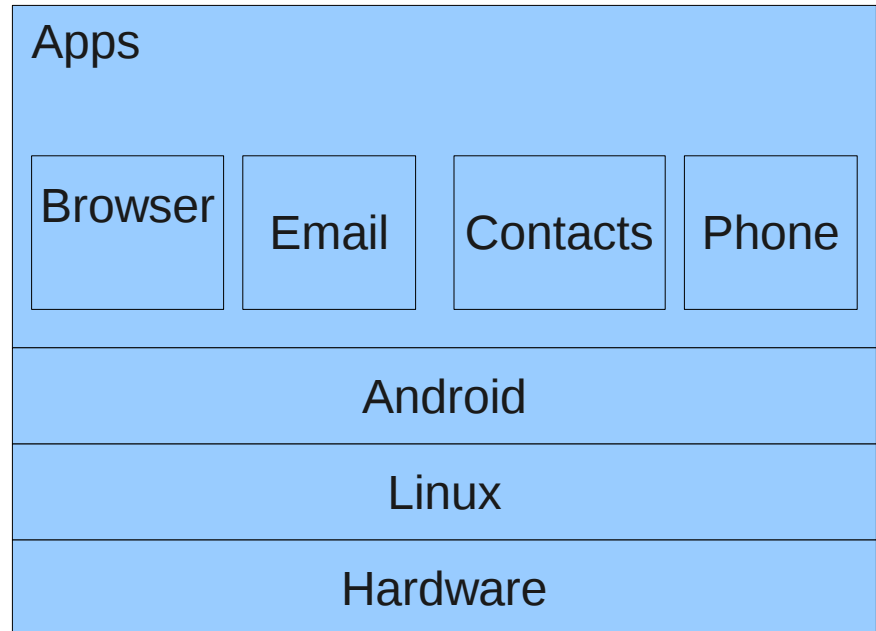
A Step in the Right Direction

- NSA Security Enhancements (SE) for Android project
 - formerly known as Security-Enhanced (SE) Android
- Identify and address critical gaps in the security of Android.
- Why Android?
 - Open source platform: suitable for a reference implementation accessible to anyone.
 - Broad market adoption: opportunity to improve the security of a widely used mobile platform.



Android Security Concerns

- Weak separation.
- Prone to privilege escalation.
- Lack of support for enforcing organizational security goals.

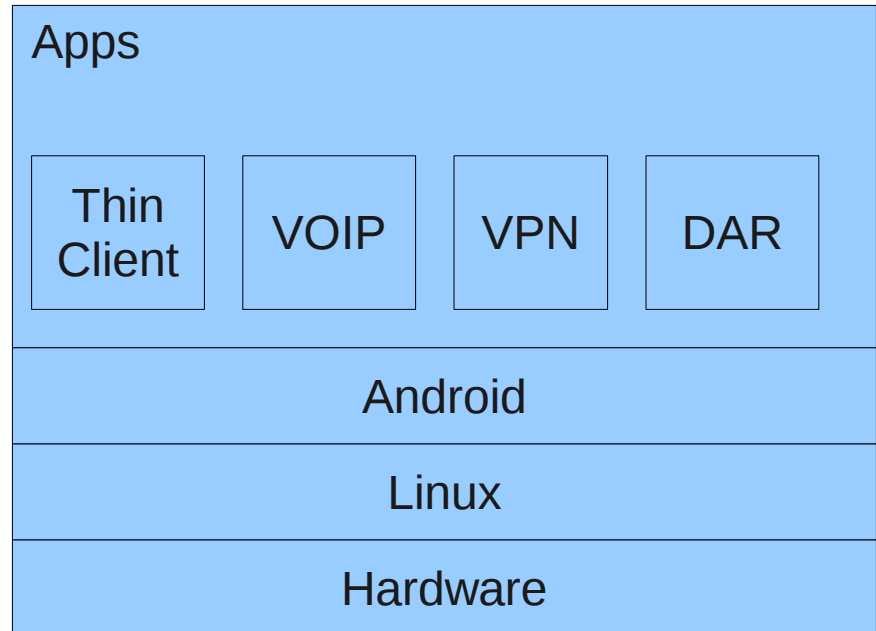




Secure Solutions on Android

Security Concerns

- Exposure of secrets.
- Protection of app mechanisms and configurations.
- No guaranteed invocation.





Building on a Solid Foundation

- Critical role of operating system protection mechanisms in supporting higher level security goals.
 - *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments*, 21st NISSC, Oct 1998.
 - Flexible Mandatory Access Control (MAC) as a key mechanism
- SE Linux as a well-established foundation for mitigating threats posed by flawed and malicious applications.



SE for Android

- Kernel Mandatory Access Control (MAC).
 - SELinux-based.
 - Root exploits are no longer fatal.
 - Apps can be strongly separated.
- Middleware Mandatory Access Control (MMAC).
 - Taking Android permissions out of the hands of users and apps.



SELinux: What is it?

- Mandatory Access Control (MAC) for Linux.
 - Enforces a system-wide security policy.
 - Over all processes, objects, and operations.
 - Based on security labels.
- Can confine flawed and malicious applications.
 - Even ones that run as “root” / uid 0.
- Can prevent privilege escalation.



How can SELinux help Android?

- Confine privileged daemons.
 - Protect from misuse.
 - Limit the damage that can be done via them.
- Sandbox and isolate apps.
 - Strongly separate apps from one another.
 - Prevent privilege escalation by apps.
- Provide centralized, analyzable policy.



Middleware MAC (MMAC)

- Many attacks occur entirely at middleware layer.
 - Cannot be addressed via kernel layer MAC.
- SELinux userspace object manager model not readily applicable.
 - Binder IPC, multi-stage call chains.
 - checkPermission API.
 - Implications for SELinux policy.
- Required a separate middleware MAC layer.



MMAC mechanisms

- Install-time MAC
 - Enforced by PackageManagerService.
 - Based on app certificate, package name.
 - Can disable even pre-installed apps.
 - Linkage to SELinux policy via seinfo tag.
- Permission revocation
- Intent MAC, Content Provider MAC



Effective Against

Root Exploits

- Motochopper
- MempoDroid
- GingerBreak
- Exploit
- Zimperlich
- RageAgainstTheCage
- KillingInTheNameOf

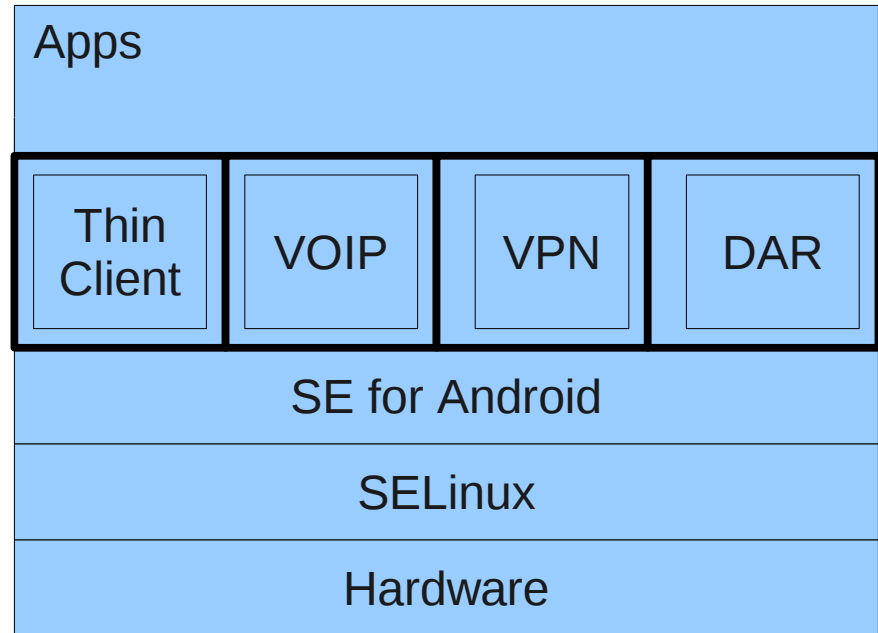
Vulnerable Apps

- Skype
- Lookout Mobile Security
- Opera Mobile



SE for Android: Security Benefits

- ✓ Strong separation of apps.
- ✓ Prevents privilege escalation by apps.
- ✓ Enforces organizational security goals.
- ✓ Protects app mechanisms & configurations.





SE for Android Timeline

Jan 6 2012
SE for
Android
released

Feb 2012
Google
begins
merge

Oct 2012
Google enables
SELinux in
userspace

Apr 2013
Galaxy S4
begins
shipping

Jan 9 2012
Google
invites
submission
to AOSP

Mar 2012
Samsung
collaboration
begins

Feb 2013
Samsung
announces
KNOX with
SE for Android

Jul 2013
Android
4.3
released



Mainline Android Adoption

- 4.1: changes below dalvik, disabled by default.
- 4.2: changes up to middleware, disabled by default.
- 4.3: (released 24 July 2013)
 - SELinux support enabled by default
 - required by the Android 4.3 Compatibility Definition Document (CDD)
 - permissive mode, liberal policy

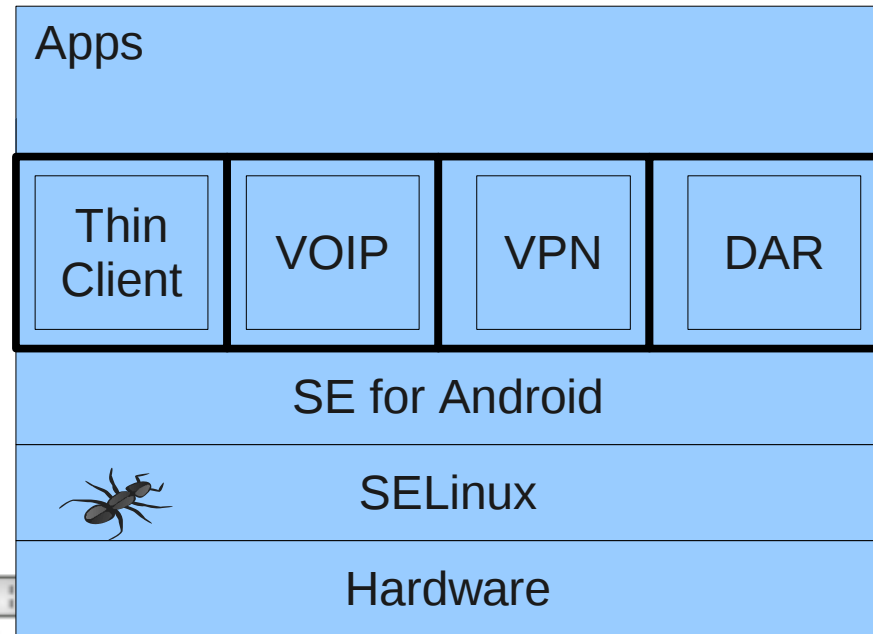


Commercial Adoption

- Samsung Galaxy S4
 - Started shipping 26 April 2013.
 - Reportedly > 20 million units sold already.
 - Shipped with Android 4.2.2 + SE for Android changes.
 - Permissive mode by default.

SE for Android: Residual Risks

- › Kernel vulnerability.
- › Platform component vulnerability.
- › Loading an unauthorized OS / configuration.





Addressing the Risks

- Requires mechanisms outside the scope of what any operating system mechanism can provide.
 - Cannot be addressed via SE for Android.
 - Also true for SE Linux (or any other secure OS).
- Two key enablers emerged in commodity PC hardware:
 - Virtualization
 - Trusted Computing



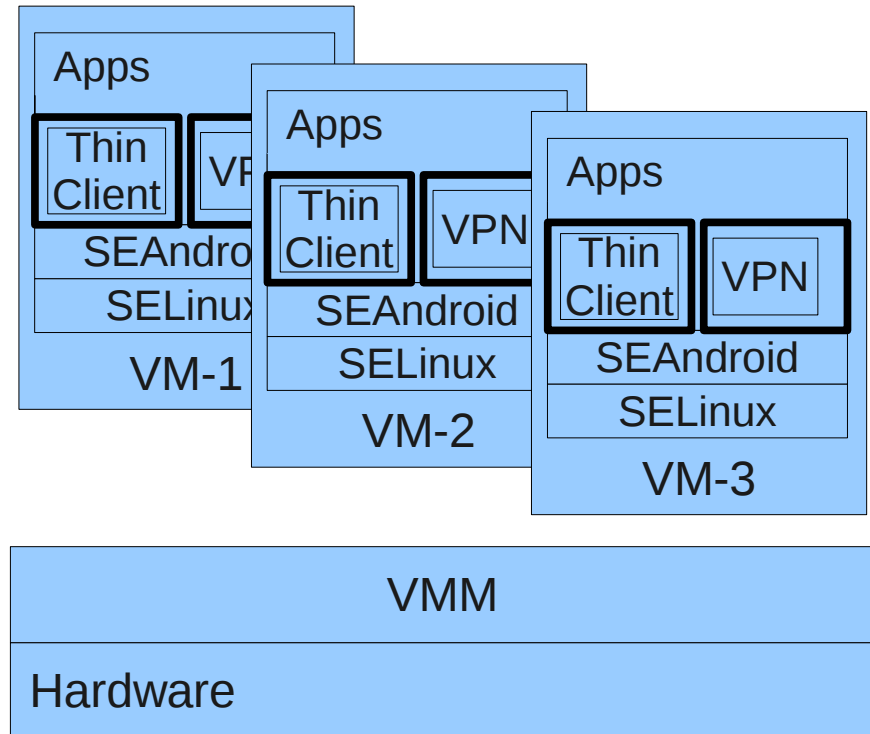
Secure Virtual Platform (SVP)

- NSA research program dating back to circa 2002.
- Explored the use of emerging hardware support for virtualization and trusted computing to address these same kinds of concerns for SE Linux.
- Investigated application of virtualization and trusted computing to construct an overall secure system architecture.

Basic Virtualization

Security Benefits

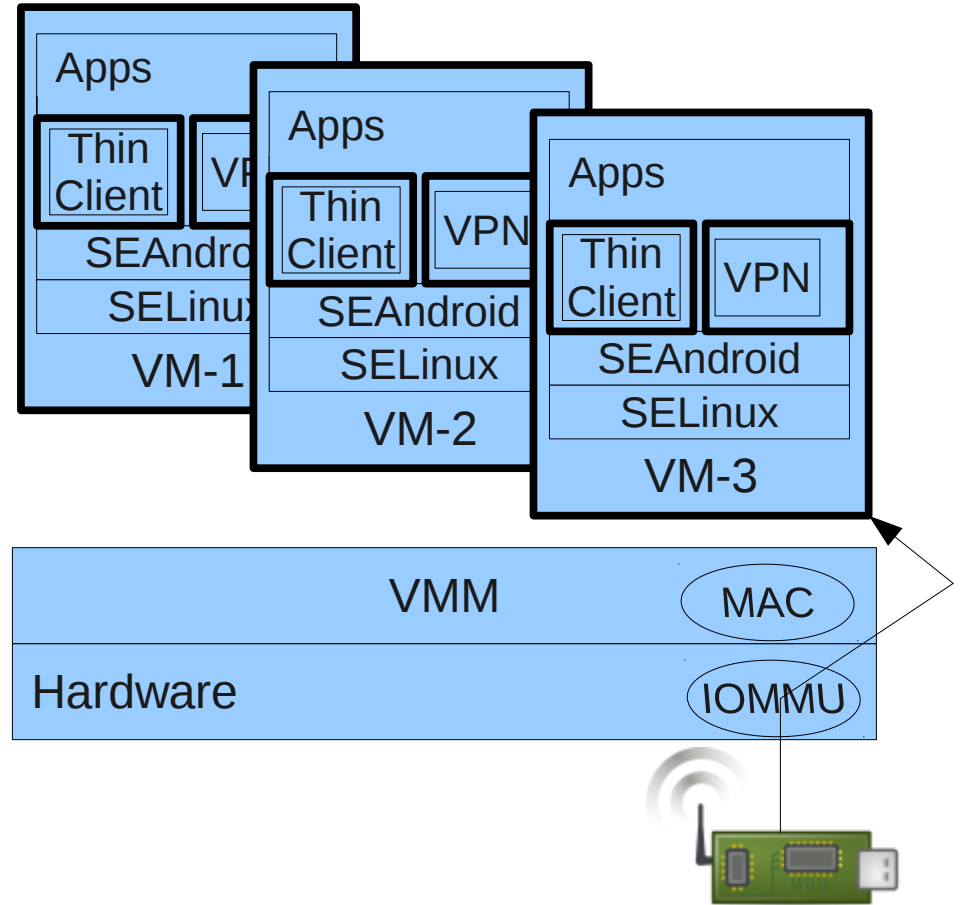
- ✓ Guest kernel vulnerability contained to single VM.
- ✓ Isolated environments via separate VMs.



Secure Virtualization

Security Benefits

- ✓ Platform component vulnerability contained to single VM.
- ✓ VM interactions and privileges controlled by MAC policy.

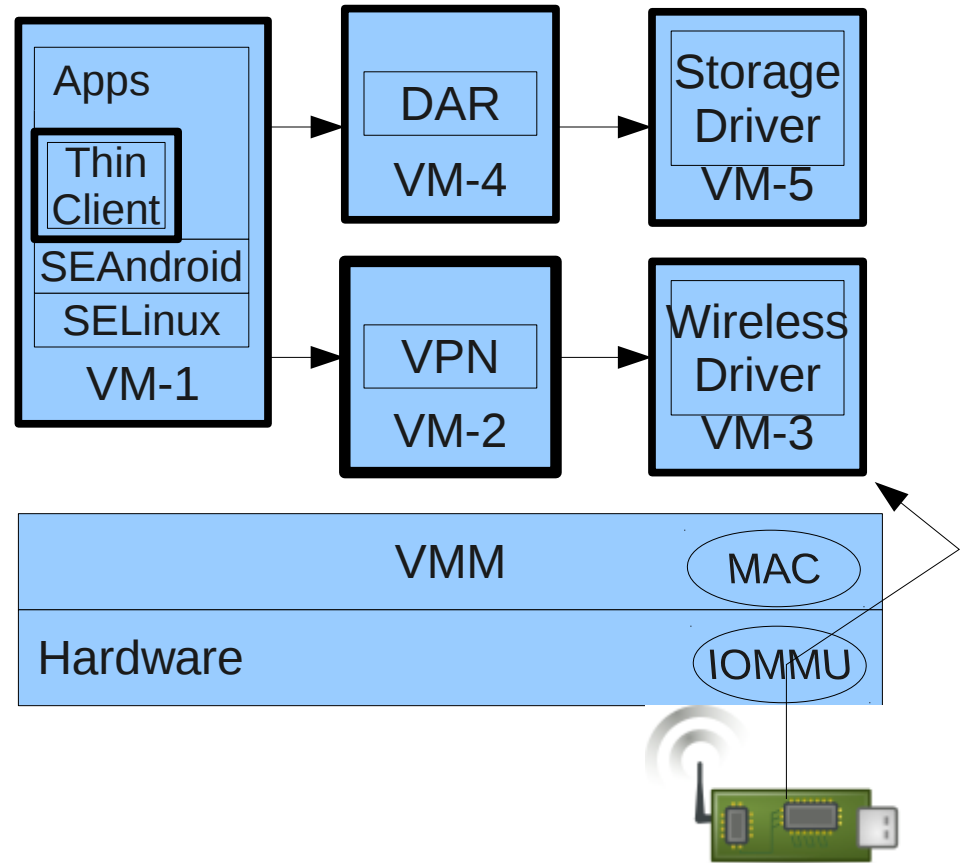




Virtualization for Security

Security Benefits

- ✓ Driver isolation.
- ✓ Protection of security services.
- ✓ Assured invocation of security services.





Virtualization instead of SE for Android?

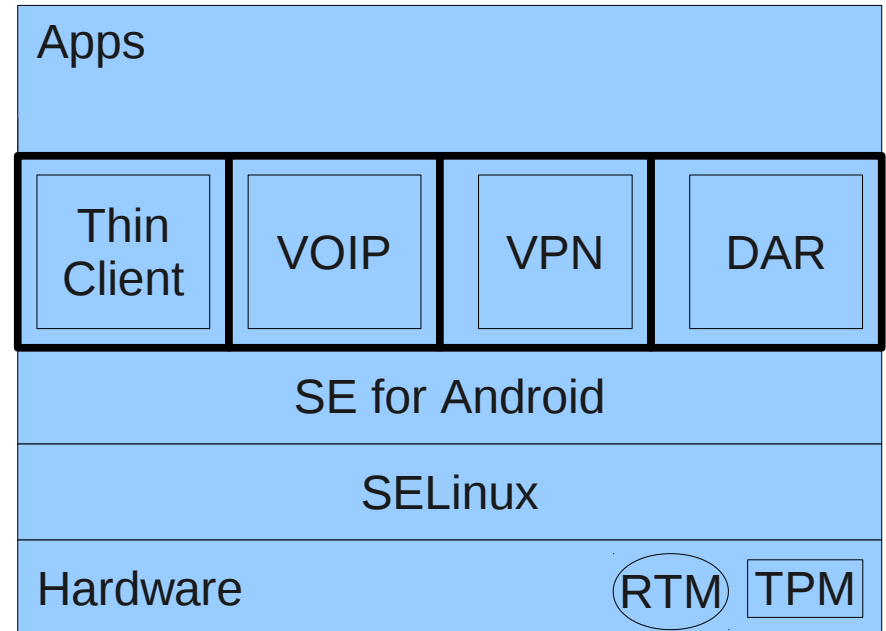
- Virtualization does not eliminate the need for a secure OS.
 - Unable to enforce security goals within guest OS.
 - Does not address need for controlled sharing.
 - Does not protect the data as it is being processed.
 - Still need to protect shared services & control plane.
 - Limited scalability and flexibility.



Trusted Computing

Security Benefits

- ✓ Verifiable, trustworthy report of loaded software & configuration.
- ✓ Protection of long term secrets from leakage or misuse by unauthorized software.
- ✓ Hardware roots of trust.

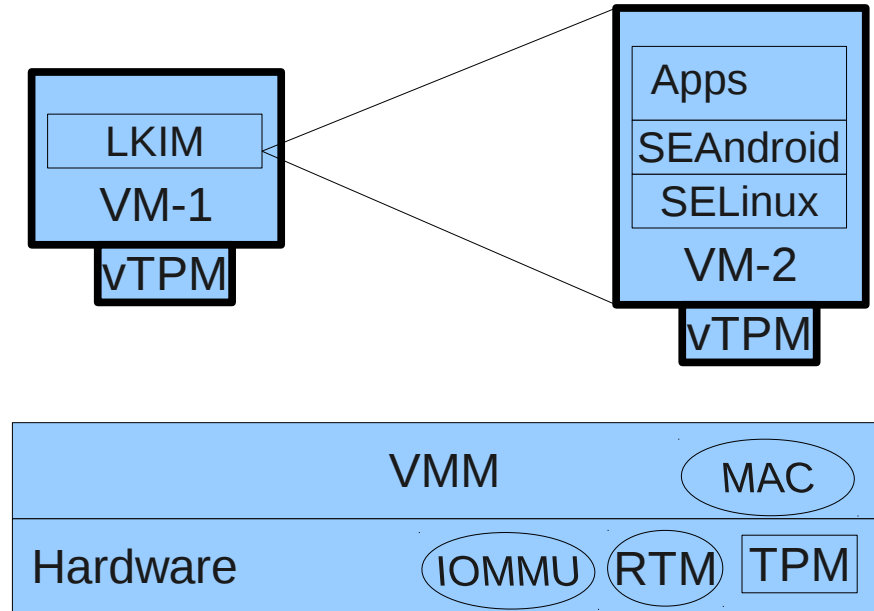




Trusted Computing & Virtualization

Security Benefits

- ✓ Extend same benefits to each VM.
- ✓ Scalable measurement & attestation.
- ✓ Runtime integrity measurement of VMs.





Trusted Computing instead of SE for Android?

- Trusted Computing \neq Secure Computing.
 - Does not remove vulnerabilities in design or implementation.
- Provides a way to validate system assumptions for secure computing.
 - Did the device boot the expected secure OS?
 - Is the secure OS running in the expected state?
- Not a substitute for a secure OS.

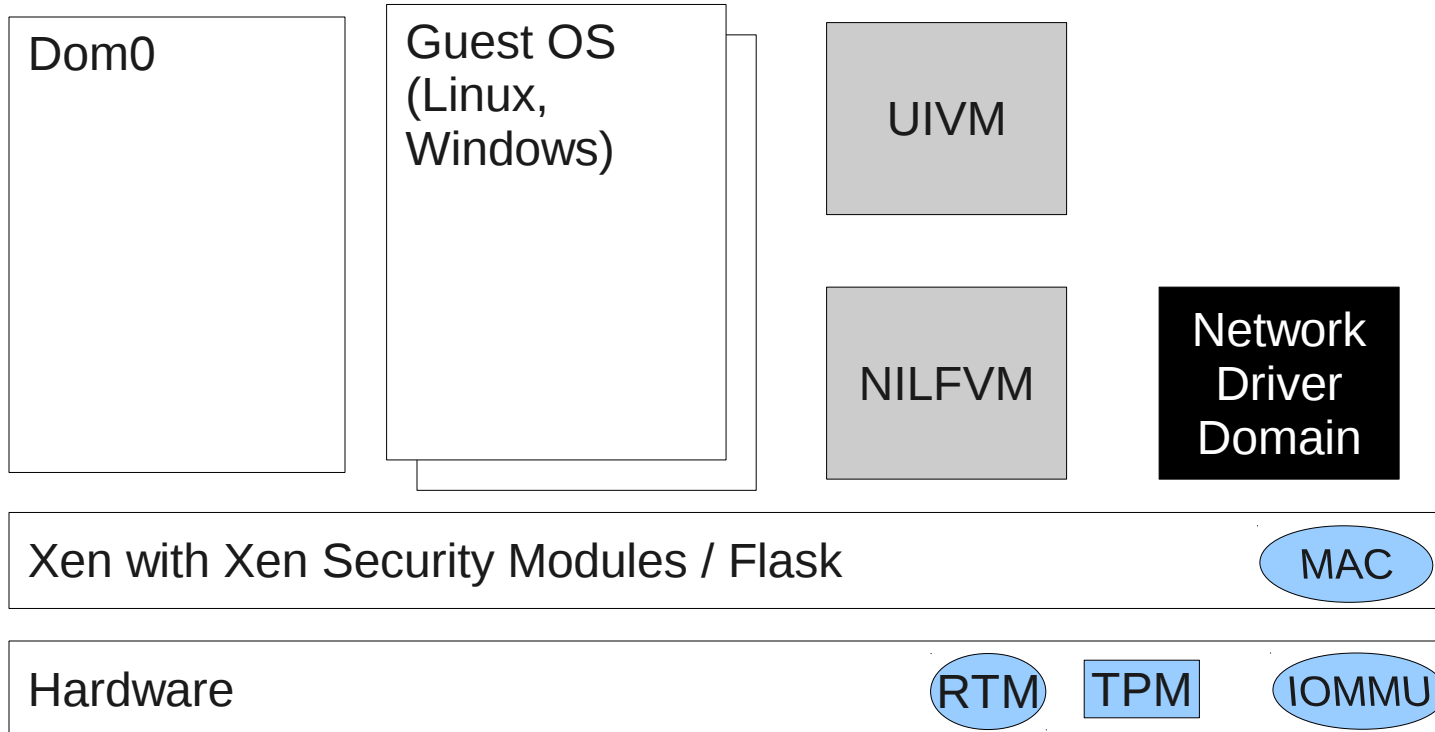


SVP Technology Transfer

- Some SVP concepts and code contributed to open source.
 - Xen Security Modules / Flask, vTPM, Linpicker
 - openAttestation
- Partial realization in commercial products and solutions.
 - XenClient XT product
 - AFRL SecureView solution



XenClient XT/SecureView



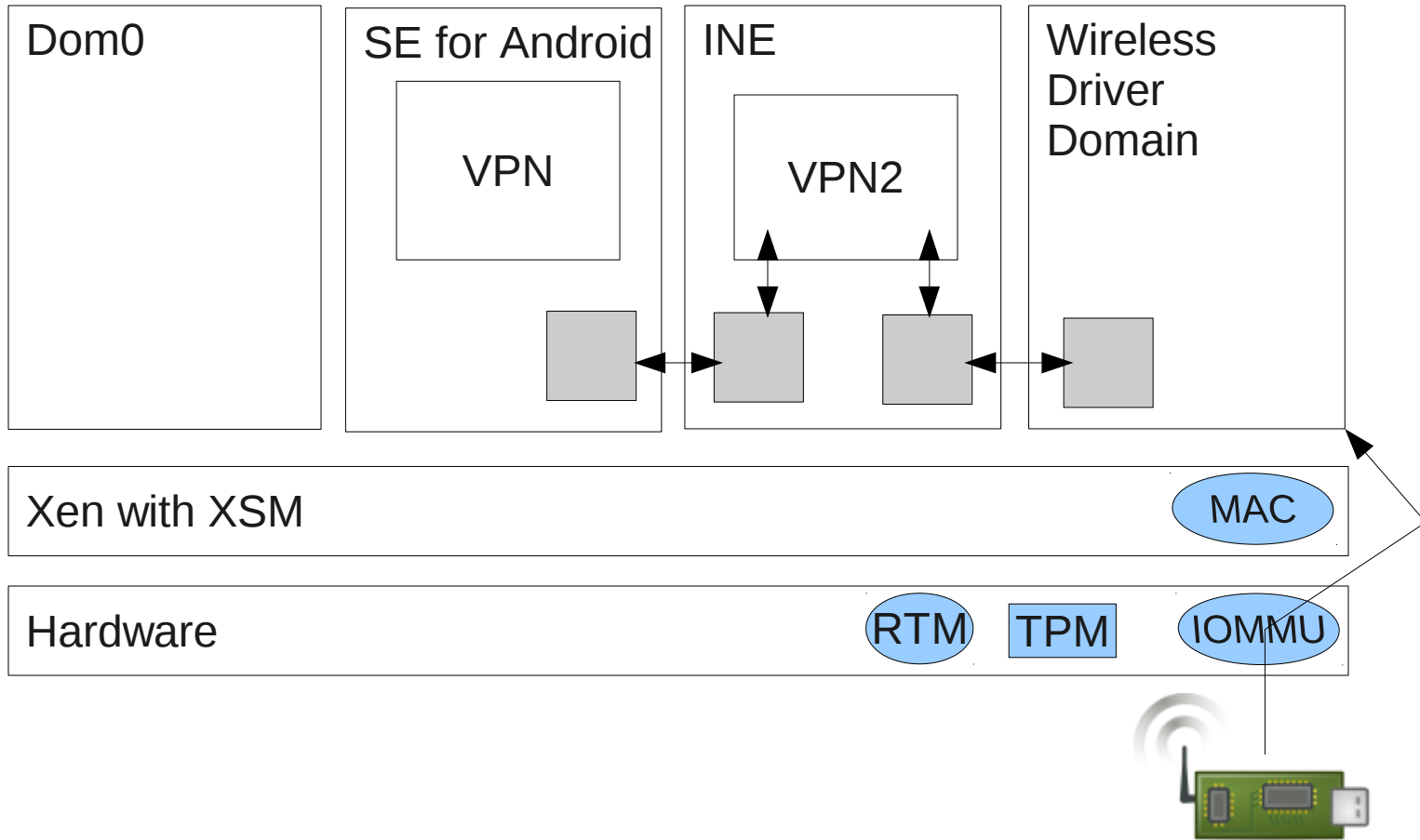


SVP: Going Mobile

- Originally implemented on PC hardware.
 - Able to leverage PC hardware primitives for virtualization and trusted computing.
 - Including TPM, RTM, IOMMU capabilities.
- Directly transferred to laptops.
 - Being leveraged in real solutions.
- Successfully ported to x86-based tablets.



Tablet (x86) Architecture



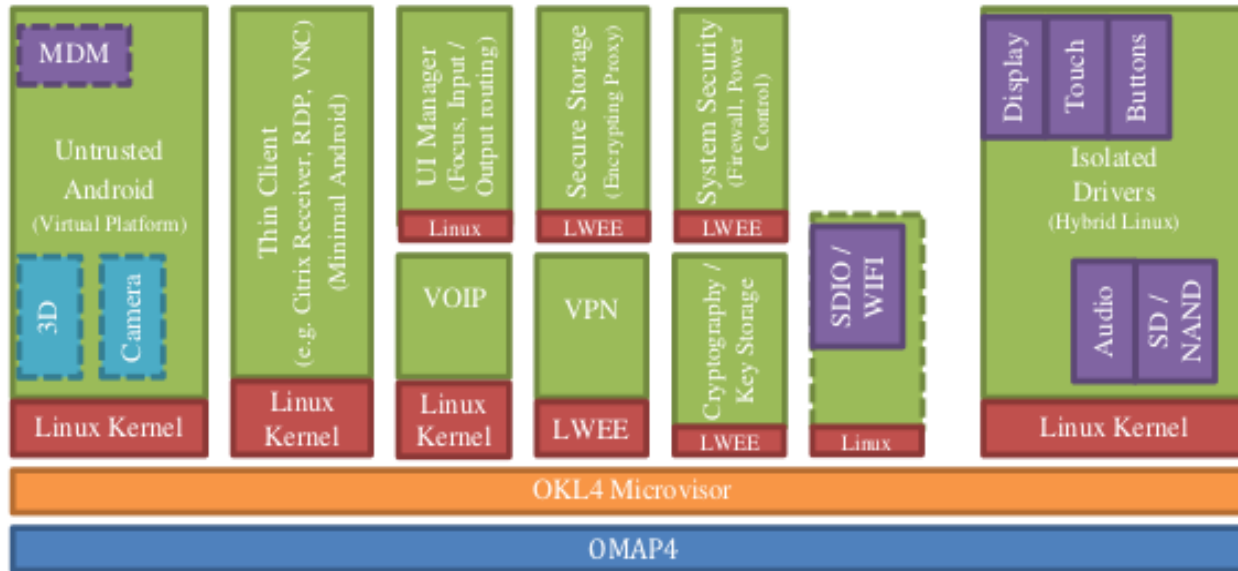


SVP for ARM: Virtualization

- Para-virtualization / microkernel hybrids
 - OKL4, Integrity
- ARM Virtualization Extensions
 - Xen, Linux KVM



OKL4-based Architecture





Concerns with ARM virtualization

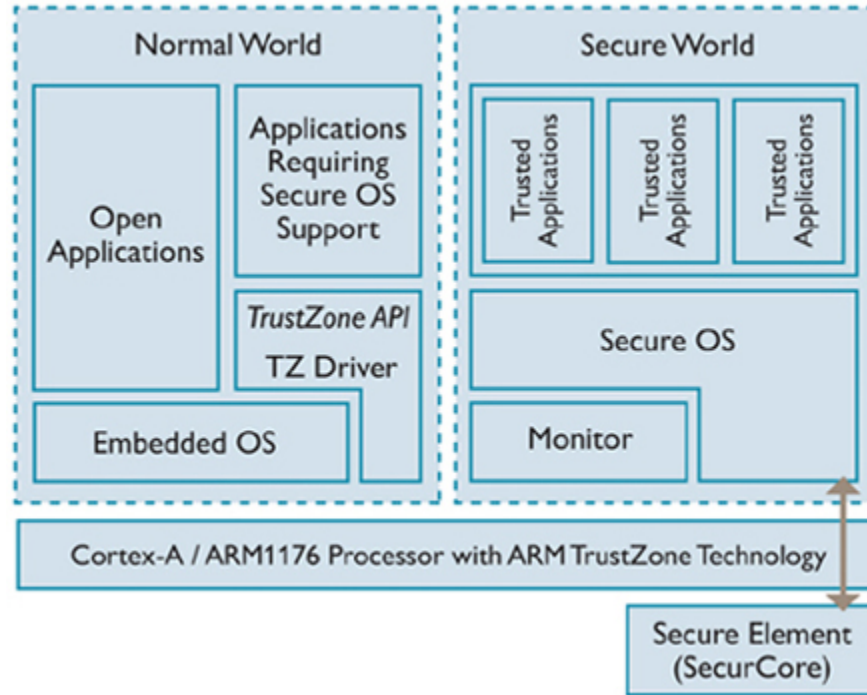
- Lack of mature, deployed virtualization solutions for ARM.
- Need for OEM cooperation.
- Frequent lack of IOMMU support.
- Static configuration of VMs.



SVP for ARM: Trusted Computing

- TrustZone as the likely foundation.
 - Becoming more commonly available.
 - Provides support for isolated execution and protected storage.
 - Possible to tie to hardware root of trust.
 - Possible place to host a MTM.

TrustZone



Source: www.arm.com/products/processors/technologies/trustzone.php



Concerns with TrustZone

- No measured launch or attestation for secure monitor and secure world OS.
- Lack of widely available MTM implementations with standard APIs.
- Lack of / unclear state of separation of trusted applications.
- Lack of public details on many aspects of implementation important to security.
- Variability across hardware.



TrustZone instead of SE for Android?

- Cannot address all security concerns of interest.
 - Cannot protect data and processing in the normal world.
 - Similar to discussion of virtualization.
 - Would lead to functional and API bloat of secure world, making it less secure.
- Also requires secure OS functionality for the secure world.



TrustZone instead of Virtualization?

- Only supports secure world vs non-secure world partitioning.
- Cannot support multiple VM architecture for security.
- Would likewise end up pushing too much functionality into TrustZone secure world.

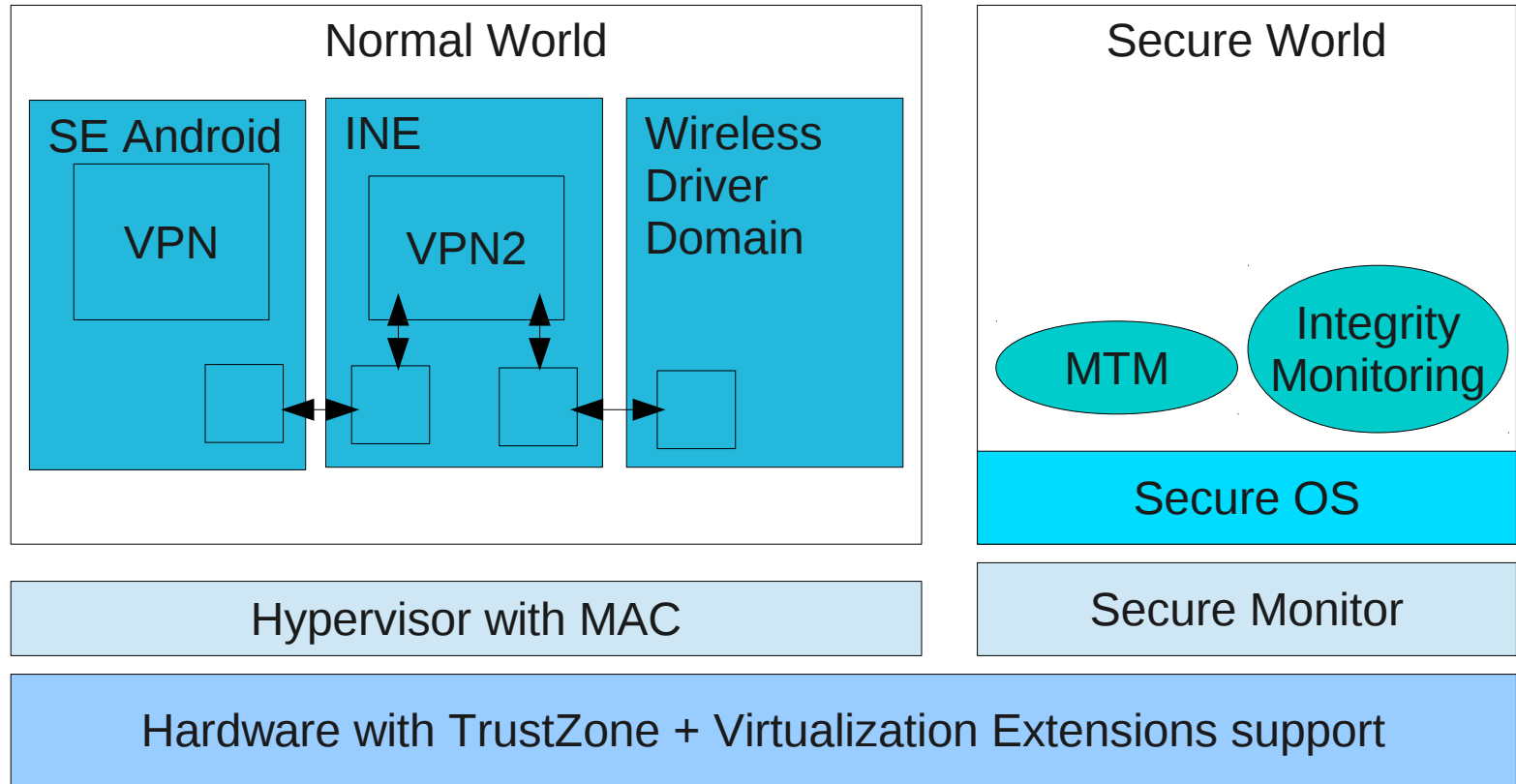


TrustZone Correctly Applied

- Measured launch for normal world hypervisor and control plane.
- Provide MTM functionality.
- Safe place for runtime integrity monitoring of hypervisor.
- Protect the underpinnings of a system with virtualization and secure OS functionality.



Putting it all together





Reaching the Goal

- Processor, SOC and device makers:
 - Make virtualization and trusted computing primitives ubiquitously available.
 - Enable use of virtualization and trusted computing by third party developers.
- Mobile platform developers:
 - Include secure OS functionality.
 - Leverage virtualization and trusted computing for security.
 - Enable third party developers to leverage this functionality / extend to applications.



Avoiding the PC malware plague

- PC industry did not address these threats early.
 - Plagued with malware as a result.
 - Trapped in a quagmire of legacy / compatibility requirements.
- Don't make the same mistake for mobile devices.
 - Device OEMs and mobile OS developers have an opportunity to do it right.
 - Mobile device ecosystem makes it possible to still change.



Questions?

- My email: sds@tycho.nsa.gov
- SE for Android project:
<http://selinuxproject.org/page/SEAndroid>
- Public SE for Android list: Send “subscribe seandroid-list” to
majordomo@tycho.nsa.gov.
- NSA SE for Android team:
seandroid@tycho.nsa.gov