Just Trust me!
Internet Enabled Devices with Integrity

Stacy Cannady, scannady@cisco.com

Technical Marketing, Trustworthy Computing

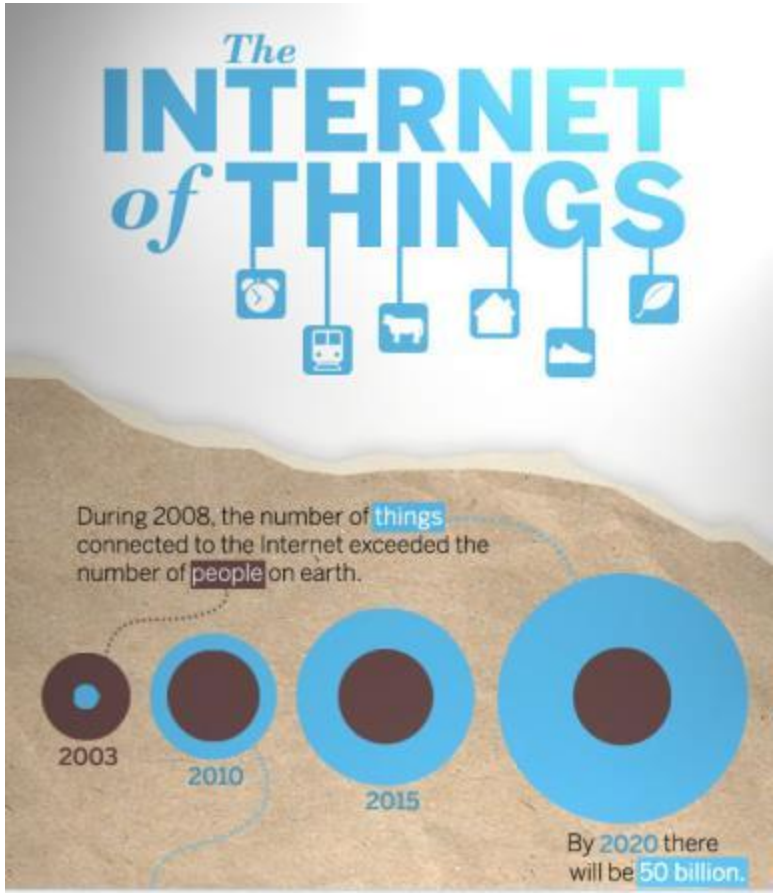
Agenda

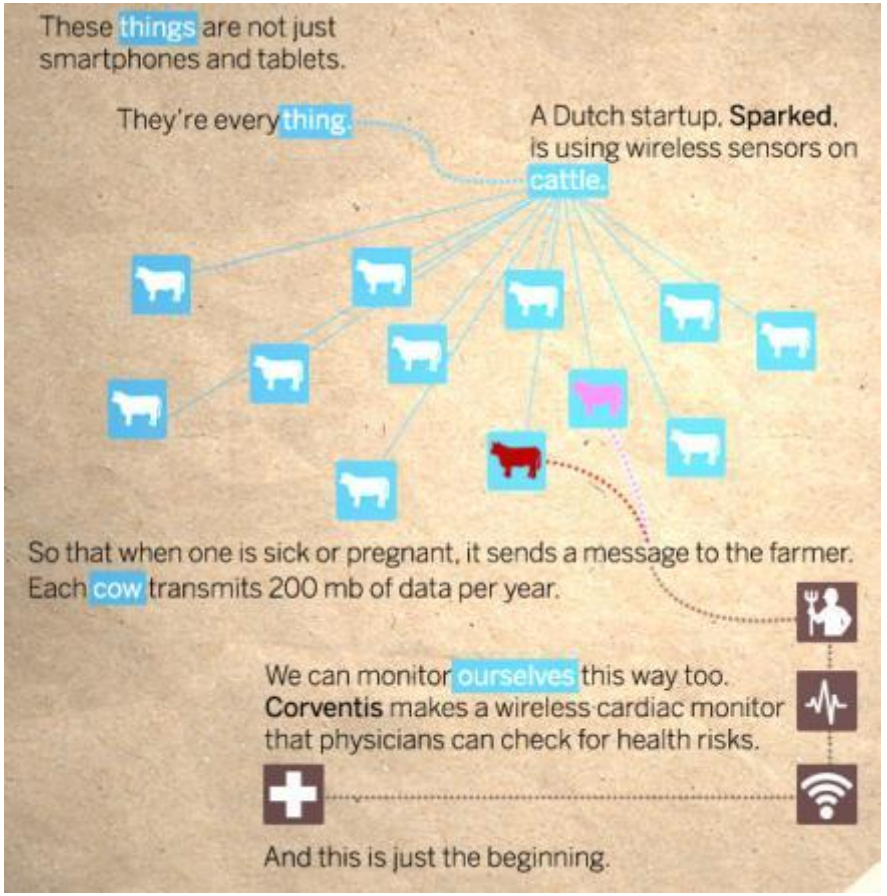
- Background – What's the Point?
- First Point – Identity and Integrity
- Second Point – Mandatory Access Control
- Third Point – Virtualization
- Resources

Background – what's the point?

The point is that we can't trust the IoT!

Internet of Things and the Risks We Face





http://blogs.cisco.com/diversity/the-internet-of-things-infographic/

Trust a computer?
Of course we trust computers!

- No, we don't **trust** our computers. We **hope** that they will do what we bought them to do and nothing more.

- We have no evidence that computers we use shall perform only the tasks we bought them to perform

- We often have evidence that they have been compromised to do something else…..

Hacking IoT for Fun!

- Texas students fake GPS signals and take control of an $80 million yacht

  http://blog.chron.com/sciguy/2013/07/texas-students-fake-gps-signals-and-take-control-of-an-80-million-yacht/

- Polish teen derails tram after hacking train network

  http://www.theregister.co.uk/2008/01/11/tram_hack/

- You may hate parallel parking, but you're going to hate it even more when somebody commandeers control of your car with you in it.

  http://news.cnet.com/8301-1009_3-57596847-83/car-hacking-code-released-at-defcon/

- Hacking insulin pumps and other medical devices

  http://www.forbes.com/sites/ericbasu/2013/08/03/hacking-insulin-pumps-and-other-medical-devices-reality-not-fiction/

Don’t Confuse User-Facing Devices with IoT

User Facing:

Android Phones

- App writers for phones notorious for poor security practices
- App stores not policed well
- Phone users notorious for risky behavior
- Phone makers and carriers can’t control any of those problems

IoT devices:

Linux / Android Embedded

- App writers can be trained to write secure code
- No App stores
- No users, or users can be constrained
- Device makers can control all of these problems

The software in an IoT device is often static
That means security can be built in by the OEM …..
and not maintained by the customer

Securing IoT:
First Point – Identity & Integrity

Do I know you?
Can I trust you?

TCG and two basic security problems for computers

- **Identity** – An asymmetric private key stored in secure hardware inside the device

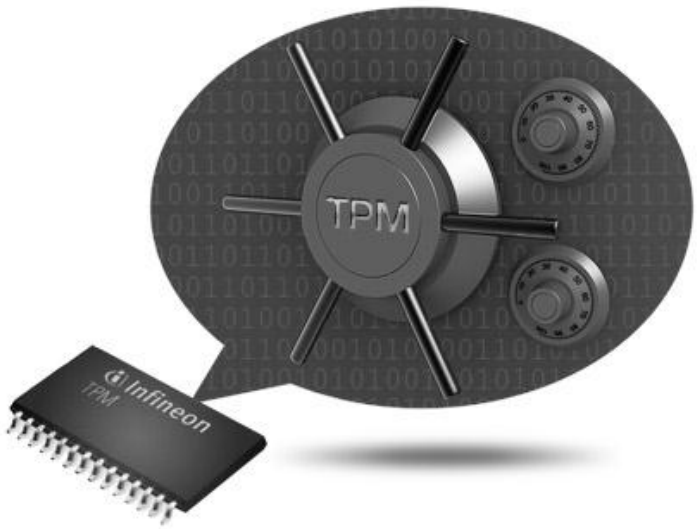
- **Integrity** – Measure code before executing it

  Hash(application.dex) vs. Golden(application.dex)
  Hash(firmware) vs. Golden(firmware)
  Hash(OS Kernel) vs. Golden(OS Kernel)
  Hash(config files) vs. Golden(config files)

  <u>If Hash() = Golden(), then the code can be trusted.</u>

"Golden" means the expected measurement, assuming the code hasn't been changed

www.trustedcomputinggroup.org



Example: TPMs Help Avoid Stranger Danger

- Problem: Pharmaceutical Company requires high level of confidence that **all** end points in the network belong to them

- Solution:
  - VPN Logon requires a digital certificate
  - Certificate is protected by a TPM
  - Therefore only company owned end-points can connect to the network

Example: The value of integrity Measurement
Google's Chromebook –A Self-Healing Computer

- Security hardware and firmware measure firmware at boot
- Measurements are internally verified
- If a mismatch is found, the offending module is rolled back to the Last Known Good version, kept on board
- Then boot continues

The computer always comes up in a known state

Second Point – Mandatory Access Control

Enforcing process and data isolation

Isolate process from each other and from the OS

- SE Linux and SE Android –
  - "Security Enhanced" Linux and Android –
    - Kernel mods, tools and configuration files
    - Initial work done by the NSA, then open sourced
  - SE Android is built on top of SE Linux

- SE Security Model:
  - Mandatory Access Control – **nothing** happens unless it is allowed to happen
  - The basic security model: there are **Subjects / Actions / Objects**
    - Subjects are processes
    - Actions are anything a process might do to an object
    - Objects are anything a process might take action on
  - The Security Server process must permit a Subject to take Action on an Object -

    Subject="Stacy" Action=Write Object=Syslog ALLOW

Application Security

- Secure by design
  - Threat modeling
  - Use Least Privilege
  - Use sandboxing
- Secure Coding – see Resources page
- Include security as part of App testing
  - Define the attack surface
  - Exercise that surface with the right tools
  - Fuzzing and robustness tests
  - Red Team hacking
  - Static analysis

Example, Motorola's AME 2000

- Smartphone user experience
  Commercial, off-the-shelf devices offer the latest capabilities, form factors and user interface
  Secure deployable data
  Extend the security and functionality of the network to the field via integrated Suite B IPSec VPN and Data at Rest protection
- Defense in depth
  Integrated security layers provide confidentiality, integrity and availability of VoIP and data communications
  Hardware root of trust
  Hardware security module provides tamper protection for keys, tokens
- Integrated security solution
  Complete end-to-end solution with single-source accountability for complete security of voice, video and applications

From Motorola's AME 2000 product page, at http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Assured+Mobile+Environment/AME2000

Third Point – Virtualization: Security that saves the OEM money

Third Point: Virtualization

- Business value: Separate hardware from software
  - Saves software migration costs as HW evolves
  - Maximizes use of available resources
  - Virtualization saves the OEM **$$$$**

- Security value:
  - Process isolation
  - Ability to create a layered security model within the embedded device

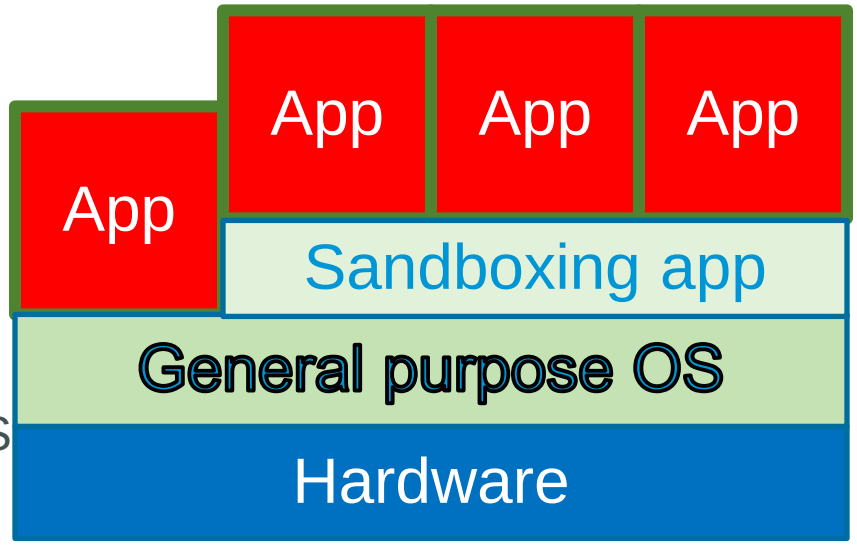
“Sandboxing” –

This is how application isolation is done in Android / iOS today.

Might be better thought of as hardened process isolation.

- Pros:
  - Widely supported under Linux
  - Lightweight and fast
  - Can support lots of virtual instances.
- Cons:
  - Weak isolation of instances and data
  - All instances must support the host OS

Example: All iPhones



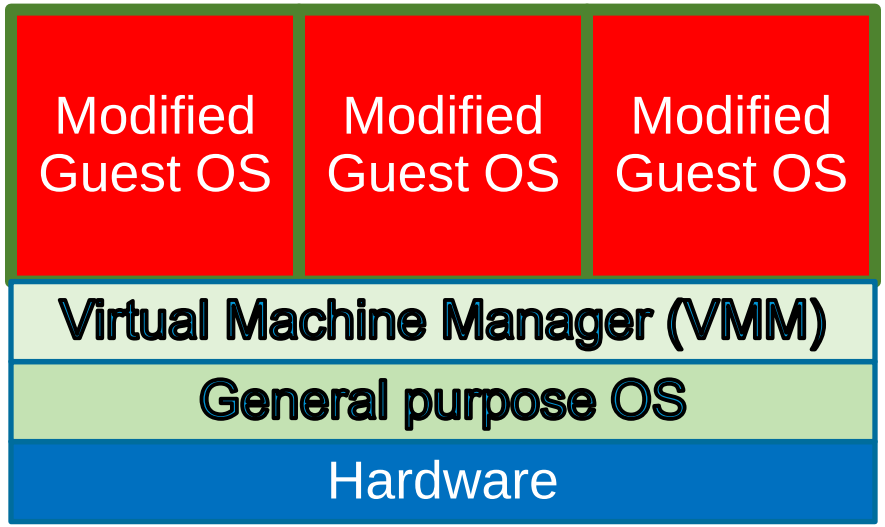
Paravirtualization or Type 2 Virtualization

A host OS runs the modified guest as an application

The guest OS is modified to be aware that it is running under another OS

- **Pros:**
  - Lightweight and fast
  - Guest OS Images are significantly smaller
  - Can be used on processors that do not support virtualization
- **Cons:**
  - Guest OS must support hypercalls instead of native functions.

Example: Samsung Knox, today



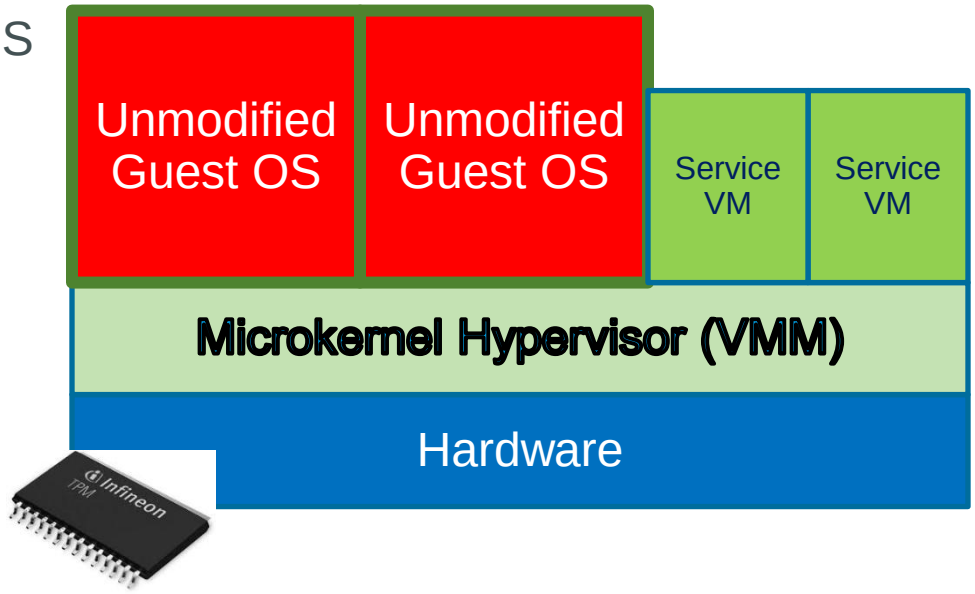
Full Virtualization or Type 1 Virtualization

The VMM represents itself as real HW to the Guest OS. The Guest OS does not know it is a VM

- **Pros:**
  - Can support any OS, without modification to that OS
- **Cons:**
  - Requires virtualization hardware (at least memory re-mapping)
  - Requires full installation of the OS

Example: Motorola AME 2000

Samsung Knox, roadmap



Resources 1/2

- Trusted Computing: www.trustedcomputinggroup.org
- SE Linux and SE Android:
  - http://selinuxproject.org/page/Main_Page
  - http://selinuxproject.org/page/SEAndroid
  - http://www.centos.org/docs/4/pdf/rhel-selg-en.pdf
- Secure coding:
  - www.safecode.org
  - http://www.android-permissions.org/
  - http://www.cert.org/secure-coding/
  - http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf
  - http://msdn.microsoft.com/en-us/library/ms995349.aspx

Resources 2/2

- Virtualization / paravirtualization primers
  - Solving real time needs in a virtualized world: http://embedded-computing.com/articles/the-multiprocessor-multi-os-systems/
  - http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf
  - http://softwarekishorekoney.blogspot.com/2011/06/full-virtualization-vs-para.html
  - http://msdn.microsoft.com/en-us/library/dd430340.aspx (Image reference)
- Virtualization – hypervisors
  - Source for a bare-metal hypervisor for ARM-7: http://dev.b-labs.com/
  - http://www.xen.org/products/xen_arm.html
  - http://www.sysgo.com/products/pikeos-rtos-and-virtualization-concept/
  - www.ok-labs.com
  - http://www.ghs.com/products/rtos/integrity_virtualization.html
  - http://www.windriver.com/products/hypervisor/
- Paravirtualization –
  - http://www.linux-kvm.org/page/Main_Page
  - http://en.wikipedia.org/wiki/Operating_system-level_virtualization
- Samsung Knox
  - http://www.samsung.com/global/business/mobile/solution/security/samsung-knox#con01