Providing E2E Security in Linux

Hadi Nahari
Security Architect
MontaVista Software, Inc.
Agenda

- Introduction
- Objectives
- Discussion
- Conclusion
MontaVista Software, Inc.

- Leading Global Supplier of Production-quality Embedded Linux OS and Development Tools

- Expert in Developing Software-intensive Products: Mobile Phones, Telecom Infrastructure Equipment, Other Embedded Devices
  - Over 20 Million Phones Shipped with MV Mobilinux
  - DoCoMo Infrastructure Built with MV CGE Linux

- Results in Increased Software Development Productivity and Reduced Time-To-Market
MontaVista Linux Products

Specialized Solutions

Communications Infrastructure

Carrier Grade Edition (CGE)
- Wireless and IP – Core/Edge
- Broadcasting
- Wireline Optical

Professional Edition (Pro)
- Medical
- Military/Aerospace
- Instrumentation & Control
- General Embedded
- Office Equipment
- DRM Set-top Boxes
- Digital Home

General Embedded

Mobile Devices

Mobilinux
- Cell Phones
- Portable Media Players
- Digital Cameras

General Purpose Embedded Solutions

Switches & Routers
- Enterprise Datacom
- Line Cards

VOIP phones
Linux: Not Just Desktop & Server!

- Linux Is Highly Active In Embedded World
- Embedded Linux Developers' Facts:
  - Estimates Are 70% Of New Semiconductor Devices Are Linux-enabled
  - 100,000~150,000 Embedded Linux Developers
  - Emerging Software Professionals Are Linux-savvy And Linux-comfortable
  - A Great Number Of Them Enjoy Hacking!
Myth Buster (10 Cardinal Sins...)

- Security Means Different Things To Different People
- Closed Source More Secure Than Open Source
- Security Could Be Achieved By Obscurity
- Software-Only Security Is Good Enough
- Security Staff Are Pain In The Neck
- Security Is A Set Of Components
- Can Protect Against All Attacks
- Encryption Equals Security
- Can Add Security Later
- Hackers Are Clueless
Objectives

- Fundamental Definitions
- Describing Problem Domain
- Proposing Possible Solution
Fundamental Definitions

- **Security Requirements?**
  - What's to be Achieved.

- **Security Assets**
  - Identify Them First!

- **Attacks**
  - Compose Attack Tree Next!
  - Devise The Protection Profile
    - What About Hardware Attacks?

- **Multilevel Security (MLS)**
  - A Must!
  - But What Does It Mean?

- **MAC & DAC**
  - What Are They? Always Need MAC?

- **Protection Strategy**
  - Access Control Mechanisms
  - Infrastructure, Application, Framework, Middleware Security
  - Intrusion Detection/Prevention Services (IDPS)
  - Hardware Security (HSM, TPM, …etc)
  - …
Objectives

- Fundamental Definitions
- Describing Problem Domain
- Proposing Possible Solution
What’s Needed

Security Infrastructure Should Provide

- Static/Dynamic Security Asset Protection
- Strong Authentication Mechanisms *(e.g. Secure Key Management)*
- Access Control Mechanisms *(e.g. Role/Name/Lattice/Vector Based Access Control)*
- Effective Containment *(i.e. Jailhouse Execution Environment)*
- Secure Update Mechanism *(i.e. Verification Prior To Installation)*
- Secure-Vault, Encrypted Filesystem
- Remote Sensitive Data Destruction Services
- Virtualization/Container Security
- Distributed Security Infrastructure

And Be

- Simple
- Flexible & Extensible
- Layered & Scalable
- Light-weight & High-performance
Objectives

- Fundamental Definitions
- Describing Problem Domain
- Proposing Possible Solution
Challenge: Establishing Trust

Leveraging “Root Of Trust” To Augment “Chain of Trust”

- SecureBoot Code
- Integrity Checking Of Basic Parameters (e.g. Public Keys)
- Validating System Images (kernel, libs, etc.)
- Authorizing Applications
- Hardware Security Device (HSM, TPM, etc.)

Additional Services?

Access Control

TPM Services

Hardware Root of Trust
The Notion of Identity

- \texttt{security\_context(Dom\_n\_id)}
  - Lacks Individual Application Identification Within a Domain
- \texttt{security\_context(Dom\_n\_id, App\_id)}
  - Individual Applications Within a Domain Identified
  - But Who Handles
    - Identity Management?
    - Access Control Definition & Enforcement?
- What's The Mediation Mechanism Across Domains??
- Who Arbitrates & Attests The Identities?
  - Hypervisor? Could It Still Be Considered “microkernel”?
Virtualized Trust Chain

Virtual Machine Monitor + Domain_level Access Control *(sHype, etc.)*

VMM/ Hypervisor Integrity Verification Code

Hardware Security Device, Virtualization-enabled

Dom_0 Access Control
- Dom_0 App Authorization
- Dom_0 System Image Validation
- Dom_0 Integrity Checking Of Basic Parameters (e.g. Public Keys)

Virtual Machine 0

Dom_N Access Control
- Dom_N App Authorization
- Dom_N System Image Validation
- Dom_N Integrity Checking Of Basic Parameters (e.g. Public Keys)

Virtual Machine N
Virtualized Trust Chain

- Granularity Is Important
- IBM's sHype Is a Step In The Right Direction
  - Available on Xen
  - VMWare ESX & Microsoft Viridian Likely to Adopt The Same Style
  - Still Not Fine-grained Enough
- Not Ready Yet: More Work Needed
Thank You