Trusted Secure Isolation
For Resource-Constrained Embedded Linux

ELC Europe 2007
Nov. 2nd and 3rd, 2007
Johannes Kepler University
Linz, Austria

Hadi Nahari
Chief Security Architect
Agenda

• Isolation Overview
• Isolation vs. Separation
• Isolation Requirements
  – What’s Missing
• What’s Secure Isolation?
• MontaVista Xen-ARM Project
• Future Enhancements
• References, Announcements
Current Isolation Solutions High-Level Design

Dom_0

Hardware Architecture

Userland

Kernel

Virtual Machine Monitor (aka “Hypervisor”)

Dom_0 App

Dom_N

Userland

Kernel

Dom_N App

Virtual Machine 0

Virtual Machine N

- Minimal Security: Only MMU
- Secure Isolation?
- VMM Access Control?
- Secure Communication?
- Secure Services?
- VM Mediated Sharing?
- Attestations by VM?
- Integrity Guarantees?
Secure Isolation Requirements

Isolation Technologies Should Provide

• Execution Segregation: Running Trusted Code
  – Along With Or Inside Untrusted Environment

• Work Across Different ARM Cores
  – With Or Without TrustZone HW

• Provide Security Controls Within Hypervisor
  – Fine Grained Enough To Guarantee Isolation
  – Coarse Grained Enough To Guarantee Performance

• GPL Jailhouse: No GPL Contamination For IP Code
• **Access Control Granularity Is Important**

• **IBM's sHype**
  - A Step In The Right Direction
  - Available On Xen
  - VMWare ESX & MS Viridian Likely To Adopt Same Style
  - Not Fine-grained Enough
  - More Work Needed: (Mainline?)

• **XSM (Xen Security Modules)**
  - NSA & NIARL Working on it
  - Includes: FLASK, ACM (sHype), dummy (default)
  - FLASK Module: Fine-grained, SELinux-like MAC
  - Interesting Approach, More Work Needed.
Secure Isolation: What’s Missing?

The Notion of Identity

- **security_context(Dom_n_id)**
  - Lacks Individual Application Identification Within a Domain

- **security_context(Dom_n_id, App_m_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 “thin layer”?
• High-Level Architecture
• Design Objectives
• Unique Consumer Benefits
• Further Enhancements
High-level Design

Domain 0
- User Space
  - App. 1
  - App. n
  - Stand-alone App.
  - Framework / Middleware
- Kernel Space
  - Back-End Drivers
  - Native Drivers
  - VMI

Domain 1
- User Space
  - App. 1
  - App. n
  - Stand-alone App.
  - Framework / Middleware
- Kernel Space
  - Front-End Drivers
  - Native Drivers
  - VMI

Secure Hypervisor

ARM Core

Security Governance Tools

Inter-Domain Communication Manager

Memory Manager

Resource Allocator

Domain Manager

Security Manager

Access Control
MontaVista Xen-ARM Design Objectives

1. Delivers a unique and timely implementation of Secure Isolation Technology for ARM Architecture, targeting the emergent Linux-based ARM cores

2. Comprises A Complete, Optimal And Linux-centric Secure Isolation Technology

3. Designed For Tight, Efficient Integration With MontaVista Mobilinux 5.x Edition On ARM Cores

4. To Be The Premier Linux-Based Secure Isolation From The Leading Embedded Linux Company.
MontaVista XEN-ARM: Unique Consumer Benefits

• Provides A Secure Isolation Solution for ARM Cores That:
  – Provides Guest Domains With TCB
    • Hypervisor Proper Small & Verifiable
    • Includes MAC (*Work In Progress*)
  – Is Optimized For
    • MontaVista Mobilinux 5.x
  – Is Linux-Centric
    • Easy To Integrate, And Efficient
  – Is Robust & Extensible
    • Is Based On Active, Advanced And Open Source Technologies
    • Has A Secure, Layered, Pluggable And Extensible Architecture
    • Paravirtualization Independent Of VT-Enabled Hardware
    • Dom0 Can Run Even During Guest Upgrade (e.g. FOTA)
  – Has A Light-Weight Implementation
    • Memory Footprint: ~2M (Hypervisor)
  – Includes Non-GPL Environment To Enforce IP Segregation
Status: What’s Already Completed

- Low-level Serial Console Debug
- Initial MMU Setup Hardwired For Xen Start-of-day
- ARM Exception Handlers
- ARM Interrupts
- ARM Timer Interrupts
- Xen Scheduler
- Xen Idle Domain
- Mini-OS Builds For ARM
- Common Hypercalls
- Memory Allocation
- Pseudo-Physical Memory
Wish List & Enhancements

• STIP API Implementation
• Trusted Interpreter
• Power Management
• Secure Native Services
  – Secure E2E OTA (End to End Over The Air) Update Infrastructure
  – Remote Destruction of Sensitive Data Mechanism
  – Remote Enablement/Disablement Infrastructure
  – SecureVault Functionality
  – Cryptographic Key Management Infrastructure
  – Backup & Restore Mechanism
• Addition of Crypto, Flash, and Legacy HAL to Secure kernel
• Debut: XEN-ARM ML On Oct. 29, 2007
• http://lists.xensource.com/cgi-bin/mailman/listinfo/xen-arm
Thank You

- Questions
- Comments
- Contact: hnahari [at] mvista [dot] com