Trusted Virtual Machine Identification (TVMI)

John Krautheim, CISSP – UMBC

June 2008
Purpose of Brief

- Inform of TVMI project
- Discuss a new way to manage Virtual Machines (VMs)
- This is a report on ongoing research
  - No releases will available in near term
  - Output will most likely be specification, procedures, and architecture – minimal code
- Still in concept phase – so feedback is appreciated
Introduction

- **Trusted Virtual Machine (TVM)**
  - Does not subvert security policy
  - Does not harm or tamper with other machines
  - Is running on trusted platform with certified hypervisor

- Need way to detect and identify TVMs
  - UUIDs can be modified and forged
  - MAC’s and processor ID’s have same problem
  - No physical binding to Trusted Platform Modules
In a Virtual World, Nothing is Real

- Many environments require tracking and auditing of information flow for legal, regulatory or policy reasons
- Untracked and unmanaged VMs provide security risk of losing control of data and access to network
- Assigning strong cryptographic identities to virtual machines provides a way to ensure all VMs are authorized and authenticated
Who Needs to Trust a VM?

- **System Owners**
  - Information system departments, service providers
  - Need assurance that VM will not harm other VMs and systems on network

- **Data Owners**
  - Own the data that is on the VM
  - Generally a company or agency
  - Data any information used or created by VM
Who Needs to Trust a VM (cont)?

- **Content Owners**
  - Providers of entertainment or other media
  - Movies, Music, Books, Games, etc.
  - Have an interest that content is not illegally distributed

- **Users**
  - Need to trust a VM will not compromise privacy
  - Need assurance that they can complete their jobs/mission.
Requirements for Trusted VM

- Must run on a Trusted Platform
  - Physical TPM is required
- Must securely boot *hypervisor* (HV) and OS
- Protocols to attest and authenticate VM
- Someone must certify and accredit the VM and platform
- Nice to haves
  - Reports of what else is running on platform
  - Other VMs, applications, etc
Example: Virtual COI Network

- A Virtual Community of Interest Network (vCOIN) is a network of VM appliances for a specific task or community.
  - Examples: Multi-agency Task Force, Operational Exercise or other Event

- Each member receives a preconfigured VM
  - VM has preconfigured applications, licenses, and keys for VPNs, secure data stores
  - VM should not be out of control of owner
  - Be able to revoke VM if compromised

- Need to Identify and Authenticate VM
vCOIN Problem
VM Security Issues

- VMs are just “files” that can be manipulated to alter configuration
- VMs are under full control of another entity
  - Hypervisor
  - Host OS
- HV can modify VM’s state at anytime
  - Suspension
  - Snapshot
  - State Rollback
- HV can spoof system calls
- HV can eavesdrop VM’s communication and play “man-in-the-middle” attacks
What are the Problems?

- HV can maliciously manipulate Guest OS
  - State Rollback/Storage/Migration
  - Snapshot and modification of memory
  - Suspension and power states manipulation
- Host OS/Device Emulation need to be trusted
- Virtual Machines can be duplicated and migrated easily
  - Want to control VM duplication and distribution
  - Migration to machine with unknown security properties
Research Description

- Develop a mechanism for identifying and authenticating virtual machines
  - What properties are required to be measured?
  - How to interface with Policy Decision Points (PDPs) and Third Party Authenticators (TPAs)?
- Propose an Attestation Infrastructure
- Protect, detect and report changes to configuration
  - Monitor for VMBRs and Blue Pills
  - Unauthorized changes to configuration
Technology Available for TVMI

- Trusted Computing Systems
  - Secure Execution Technology
  - Trusted Platform Modules
- Virtual Trusted Platform Module
- Special Purpose Service Domains
  - Mini–OS
  - vTPM, TPM Emulator
Virtual TPM

Host OS (dom0)  vTPM Domain (domTPM)  Trusted Virtual Machine (domU)

vTPM Back End  vTPM Front End

Xen Hypervisor

TPM Hardware

Xen Summit Boston 2008
What is a vTPM

- Emulated TPM in a special driver domain
  - Front-end driver in domU
  - Back-end driver in domTPM
  - Emulated TPM in domTPM
- Protected through binding to physical TPM
- One vTPM instance per domU
- Provides all TPM functionality for domU
Problems implementing vTPMs

- TPM does not have an Endorsement Key
  - Cannot vouch for its authenticity
  - Someone else needs to do it
- Need to bind to physical TPM
- Need to assure that TPM has valid EK
  - A TPM emulator could be used to bind to a TPM
- How to handle migration
- How to handle duplication
  - Do not want to have two machines with identical identities
Linking vTPM to TPM

- Problem: How does one “lift” the trust inherent in the hardware to the virtual machine?
- What are the identity requirements for the virtual machine?
- Proposed Solutions
  - Remotely attest the VM’s configuration to a Policy Decision Point (PDP) or Third Party Authenticator (TPA)
  - Assure that the configuration of the platform adheres to a specified security policy.
  - Receive authorization to launch applications
  - Enforce security policy and detect tampering through secure HV and hardware enforcement
What is TVMI Project

- Architecture and specification for identifying TVMs
- Proof of concept implementation
- Part of larger Virtual Centurion Project
  - Trusted Virtual Machine Environments
  - High-integrity environments required for protecting information and privacy
  - Managing and operating VMs for use in vCOINs, multi-level security systems, and other applications
TVMI Project Goals

- Provide uniquely identifiable VMs for use in trusted environments
- Determine what is an acceptable attestation for verifying the configuration the machine
  - Determine the identity parameters
  - What parts of platform need to be identified?
- Show how the identity of the VM is generated, stored, and reported
- Develop protocol for vTPM to TPM binding
Why is TVMI different than vTPM?

- Creates unique identity for each VM
  - Two “identical” VMs could have same measurement
  - Different than attestation identity
  - vTPM is used to store identity
  - The identity is provides an “experience” of the VM
    - As each VM evolves, it’s identity tells something about it
      - It’s migrations, duplications, configuration changes
      - Performed through extending PCRs
    - The identity provides a “genealogy” of the VM
      - The identity can be associated to it’s ancestors, like DNA
Leverage Current Technology

- Hardware based VM Monitoring and Containment
- Trusted Computing Group Specifications
  - Trusted Platform Module
  - Trusted Reporting Mechanism
- Secure HV and virtual TPM
  - Extend current work to support VM identification
Requirement for TVMI

- Each TVM should have an associated domTPM
  - Machine should not function without vTPM presence
  - TVM domain should not contain any other drivers
- domTPM must migrate/duplicate with TVM
  - Tightly coupled
  - Reassignment of keys should update identity
- Host platform must be able to attest it’s configuration to TVM
  - Hypervisor Version
  - OS Version, Patch levels
Notional Architecture

- Components used to build TVMI
  - Dell Optiplex 755 vPro
    - Intel VT, VT-d, TXT, and AMT
  - Fedora 8
  - Xen 3.2
  - vTPM
  - MiniOS
  - sHype
  - TrouSerS
  - Trusted GRUB
Protocol Overview

- Create Unique Master Key (Identity) for vTPM
  - Protects vTPM from malicious modification
  - Seal other cryptographic keys in TVM with master key (VPN, Storage, etc.)
- Bind Master Key to physical TPM through seal
  - Must have a-priori knowledge of TPM
  - Must have ownership privileges of TPM
- Use vTPM to securely launch TVM
  - TVM will only run on platform with bound TPM
  - Migration possible by using migratable keys
- Machine is now truly identifiable and unique
Advantages of TVMI

- TVMI Provides Unique Identity for each TVM
- Detects migration and duplication
- Ensures proper host properly configuration
- Provides authenticated network connect
- Provides fine-grained control of TVM network
- No additional hardware required
  - Runs on currently available COTS systems
Feedback Appreciated

- Value and feasibility of project
- Ways to improve project
- Collaboration and funding opportunities
- Send comments to
  - jkrautheim@gmail.com
  - http://cyberlab.cs.umbc.edu/feedback
    - Registration required
Conclusion

- TVMI is a new method to identify and manage VMs
- Unique cryptographic identities enhance manageability of virtual networks
  - vTPMs used to create and track identities
  - Identities are fully migrateable and allows duplication