KVM and Hypervisor Security

David Shepard and Matt Gaston
CMU/SEI Cyber Innovation Center

February 2012

Overview

CMU/SEI Cyber Innovation Center

Dynamic On-Demand High-Performance Computing System

KVM and Hypervisor Security

Recommendations

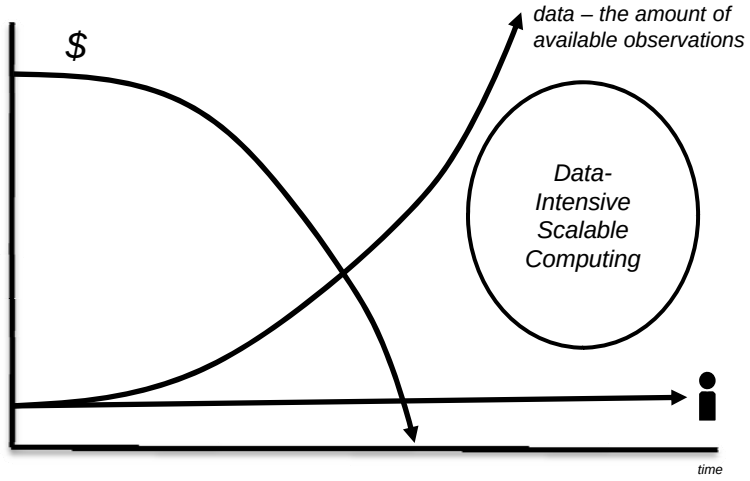
Acknowledgements

CMU/SEI Cyber Innovation Center

Accessing and leveraging leading-edge software capabilities for the Department of Defense and the Intelligence Community

We focus on identifying, demonstrating, and applying innovative technologies for critical information and computational needs of the DoD and the IC.

- Shape and leverage academic and industrial research wherever possible
- Employ creative solutions to finding, assessing, and proving technology capabilities for mission applications
- Develop and extend software technologies tailoring them for application to government mission needs
- Promote government awareness and knowledge of emerging technologies and their applications



Motivation: A Heterogeneous HPC Utility Cloud

Dynamic On-Demand High-Performance Computing System (DODCS)

Heterogeneous On-Demand Processing

Tiled Processor:

•(10) Tilera TILEmpower

GPU Cluster:

•(3) Tesla S2050

Shared Memory:

•(1) SGI UV100

Commodity Cluster and Storage

HPC Cluster

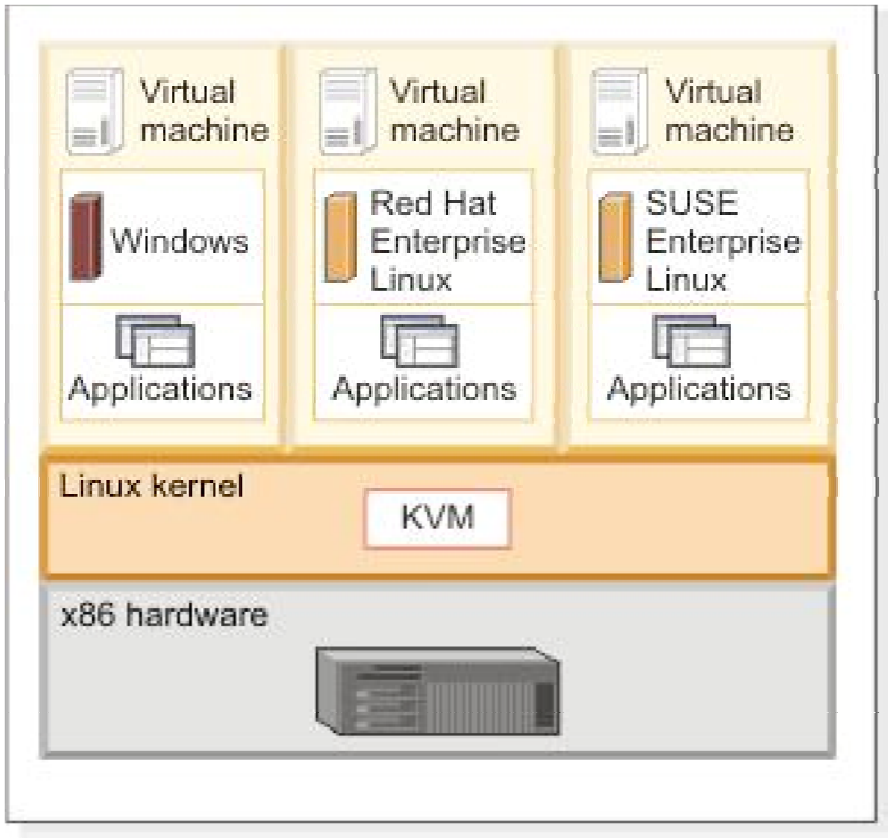
Storage Array

openstack™ CLOUD SOFTWARE

USC INFORMATION SCIENCES INSTITUTE

Dr. Steve Crago et al.

Kernel-based Virtual Machine Architecture



Source: Linux Information – Virtualization on Linux - KVM, IBM, 2011

BLUF

It is possible to secure a KVM-based hypervisor against an appropriate risk target for a defined level of security.

Compelling advantages of KVM:

- The majority of the footprint is running as an unprivileged user-mode process
- Draws on a huge base of drive support from the Linux kernel
- Open source

When considering hypervisor security, the Type 1 vs. Type 2 distinction is not helpful. Rather, the focus should be on security aspects of virtualization technology and how specific implementations address these aspects.

Note: There is no such thing as perfect security.

Type 1 vs. Type 2?

Performance: KVM uses hardware support for virtualization, effectively running on 'bare metal.' KVM allows the guest VM to run at processor ring level zero for performance.

Security:

- Type 1 hypervisors make use of privileged guest VMs for maintenance and management – a compromise of a guest compromises the hypervisor
- KVM uses privilege "de-escalation" – there is no privileged guest VM - only user requests that require privilege escalation use it
- The KVM software stack is minimal – only the kernel, a few system daemons, and QEMU (comparable to Xen)

The Threat Model

The threat considered: “regular” users – insiders

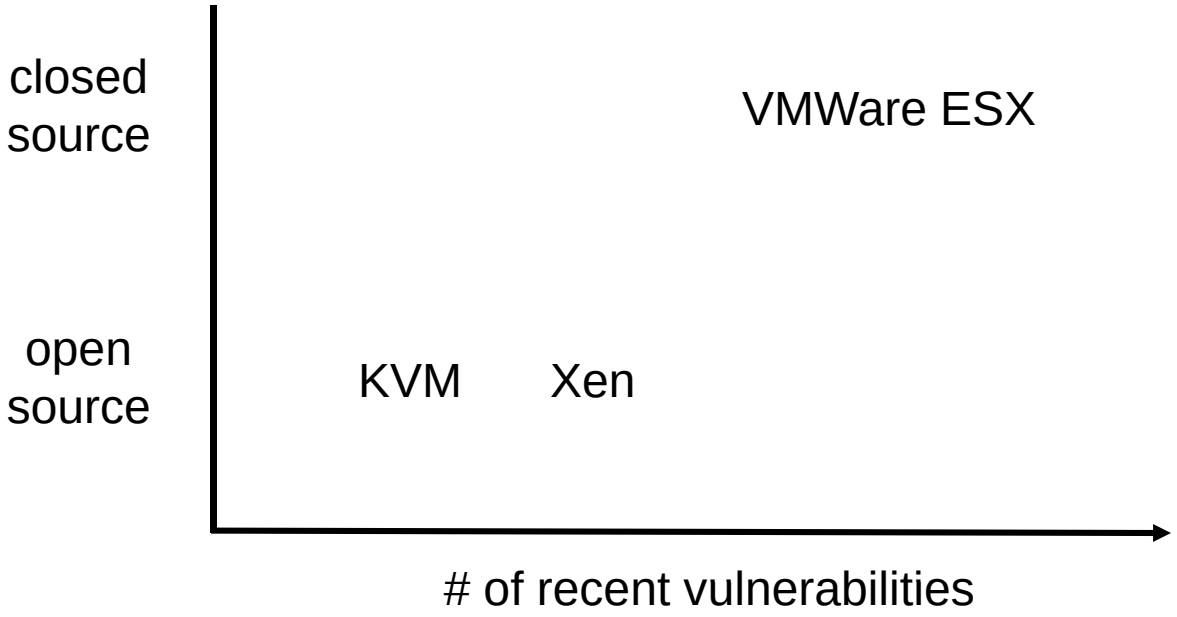
Types of security breaches:

- **VM Escape** – compromise of the hypervisor and assumption of control over all VMs
- **Privilege Escalation** – an exploit that allows an unprivileged VM to execute code in a privileged context
- **Denial of Service** – system crash or access to system resources are denied

Known Vulnerabilities

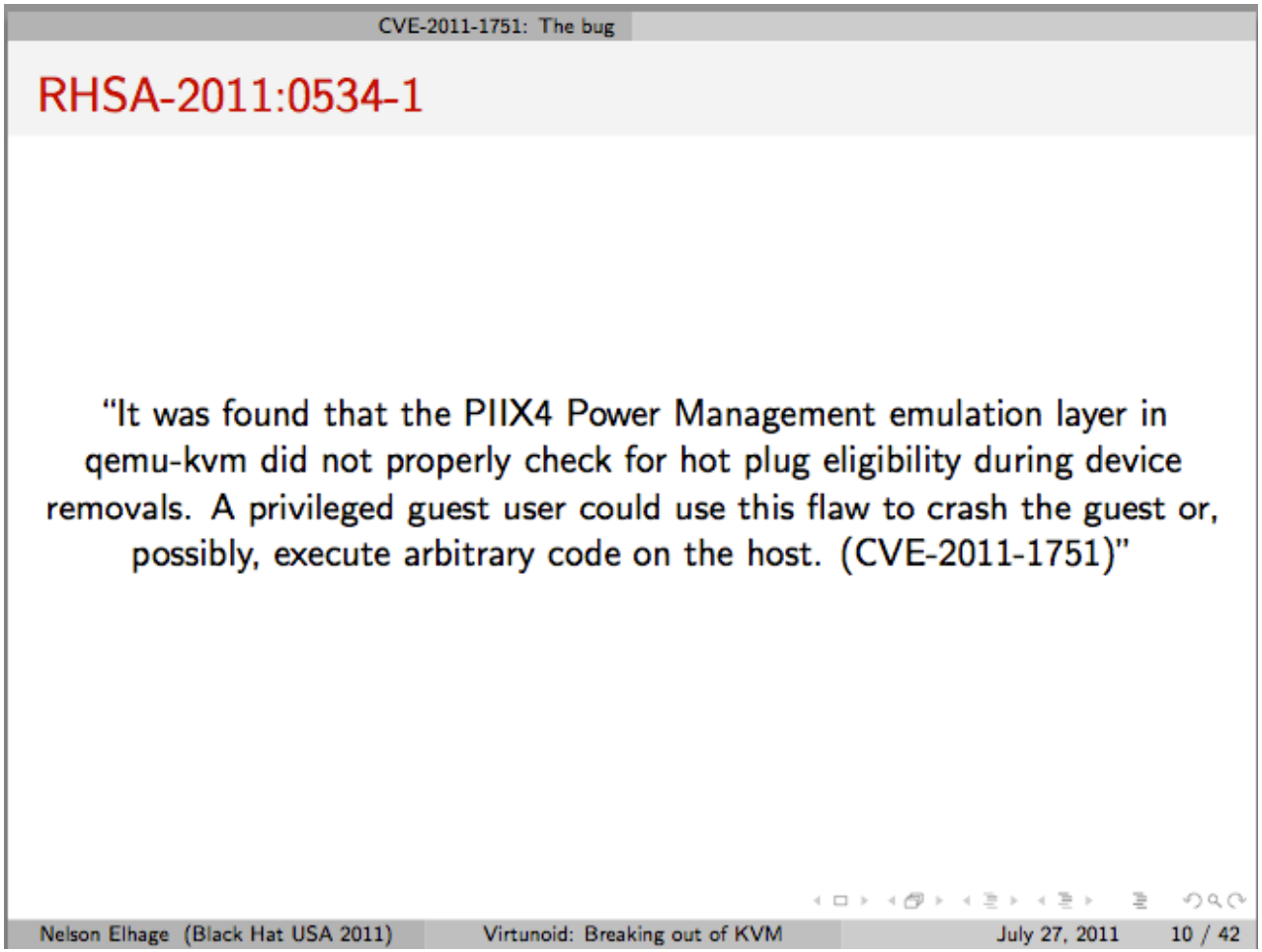
Vulnerability databases:

- NIST National Vulnerability Database
- MITRE Common Vulnerabilities and Exposures
- USCERT Vulnerability Database



As of June 1, 2011

An Example KVM Vulnerability



General Recommendations

- Apply a proper SELinux security policy to strengthen resource separation.
- Patch, patch, patch.
- Use physical separation for different authorities.
- Keep logs of guest behavior
- Keep logs of administrative activities
- Backup critical systems.

Security Considerations for KVM

- It does not appear unsafe to support access to physical hardware in support of provisioning heterogeneous processors as the access is limited to the VM guest.
- More robust to failure since guest VMs run as unprivileged processes
- To compromise the “host” from a guest VM (unprivileged) is exceedingly difficult
- Small code footprint

Acknowledgements

- CMU CyLab
- Redhat
- USC/ISI
- NSA IAD
- Aerospace
- SEI/CERT

And Remember, there is no such thing as perfect security.