

# KVM and Hypervisor Security

David Shepard and Matt Gaston  
CMU/SEI Cyber Innovation Center

February 2012

© 2012 by Carnegie Mellon University. Published  
by The Aerospace Corporation with permission.



Software Engineering Institute

Carnegie Mellon

# Overview

---

CMU/SEI Cyber Innovation Center

Dynamic On-Demand High-Performance Computing System

KVM and Hypervisor Security

Recommendations

Acknowledgements

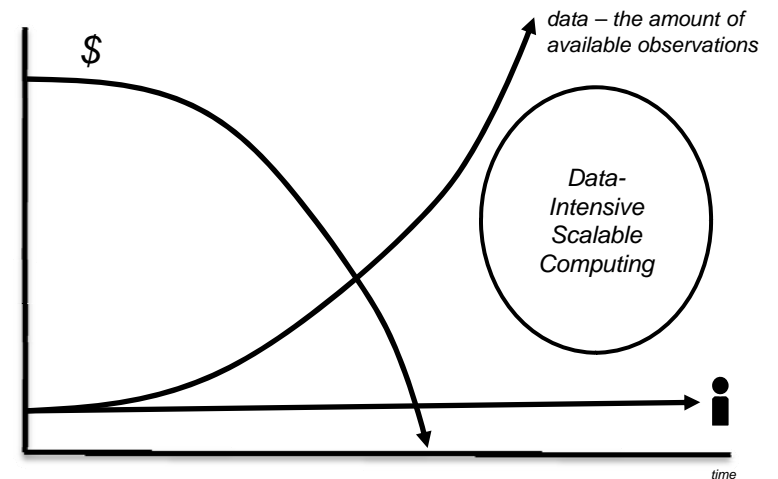


# CMU/SEI Cyber Innovation Center

*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

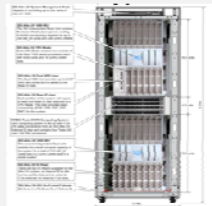


### Tiled Processor:

- (10) Tilera TILEmpower

### GPU Cluster:

- (3) Tesla S2050



### Shared Memory:

- (1) SGI UV100

### Commodity Cluster and Storage



HPC Cluster



Storage Array



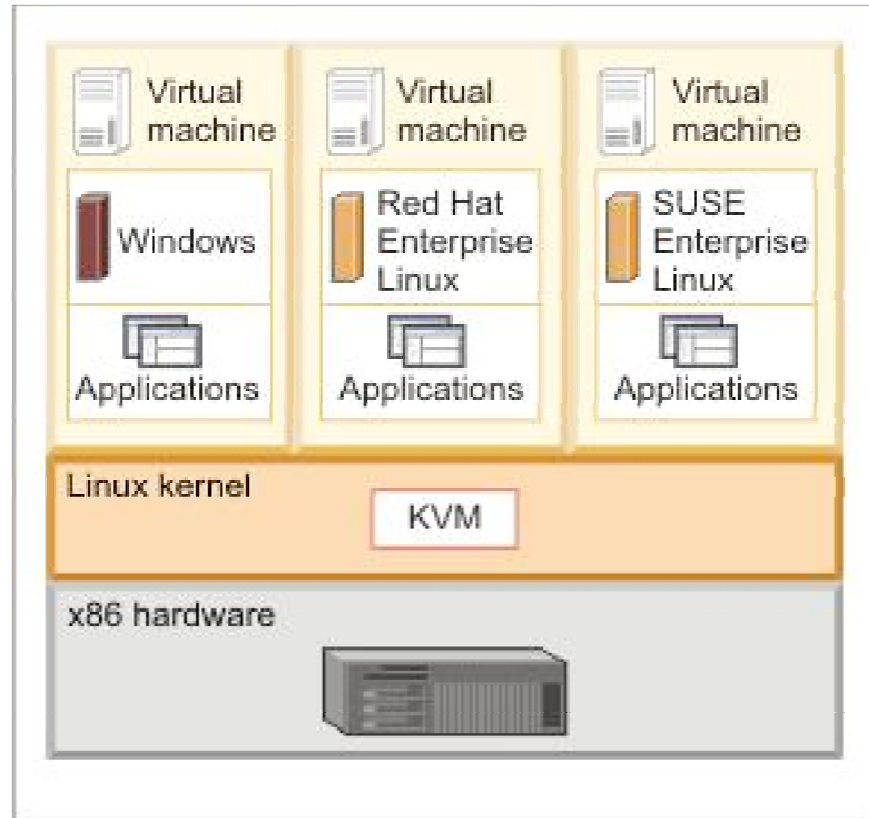
Dr. Steve Crago et al.



Software Engineering Institute

Carnegie Mellon

# 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 driver 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



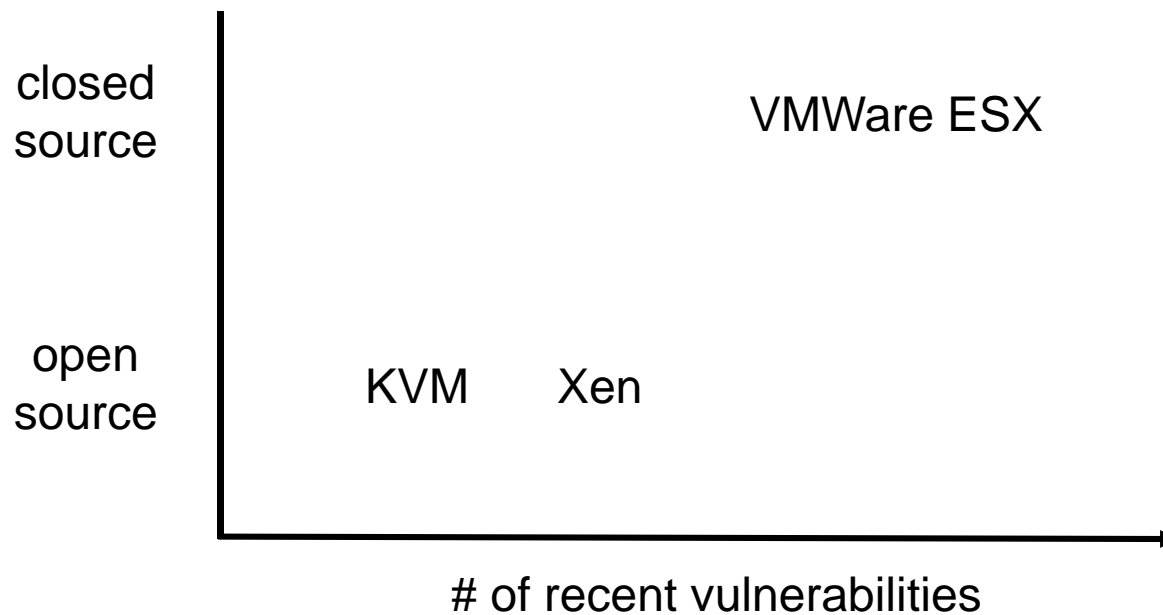


# Known Vulnerabilities

---

Vulnerability databases:

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



*As of June 1, 2011*



# An Example KVM Vulnerability

---

CVE-2011-1751: The bug

**RHSA-2011:0534-1**

"It was found that the PIIX4 Power Management emulation layer in qemu-kvm did not properly check for hot plug eligibility during device removals. A privileged guest user could use this flaw to crash the guest or, possibly, execute arbitrary code on the host. (CVE-2011-1751)"

Nelson Elhage (Black Hat USA 2011)    Virtunoid: Breaking out of KVM    July 27, 2011    10 / 42



# 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.*



---

Copyright 2012 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

*Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.*

*Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.*

*External use. This document may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).*

*This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.*

*For information about SEI reports, please visit the publications section of our website (<http://www.sei.cmu.edu/publications/>).*

