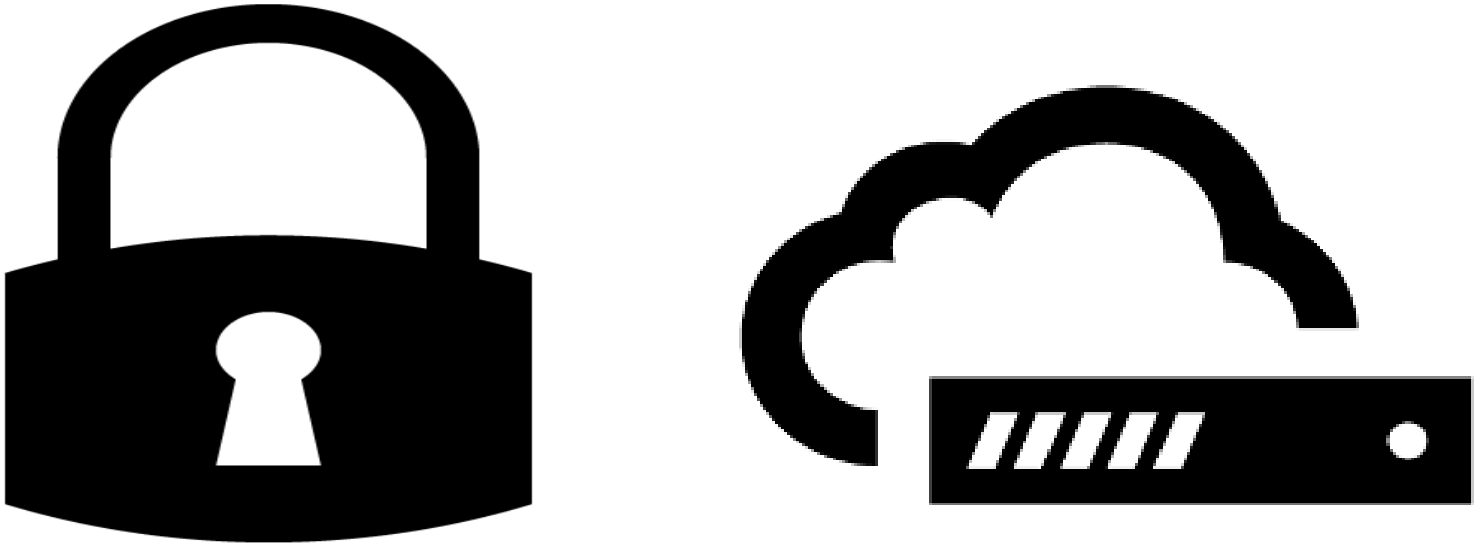
Securing your Virtual Datacenter



Part 1: Preventing, Mitigating Privilege Escalation

NSRC

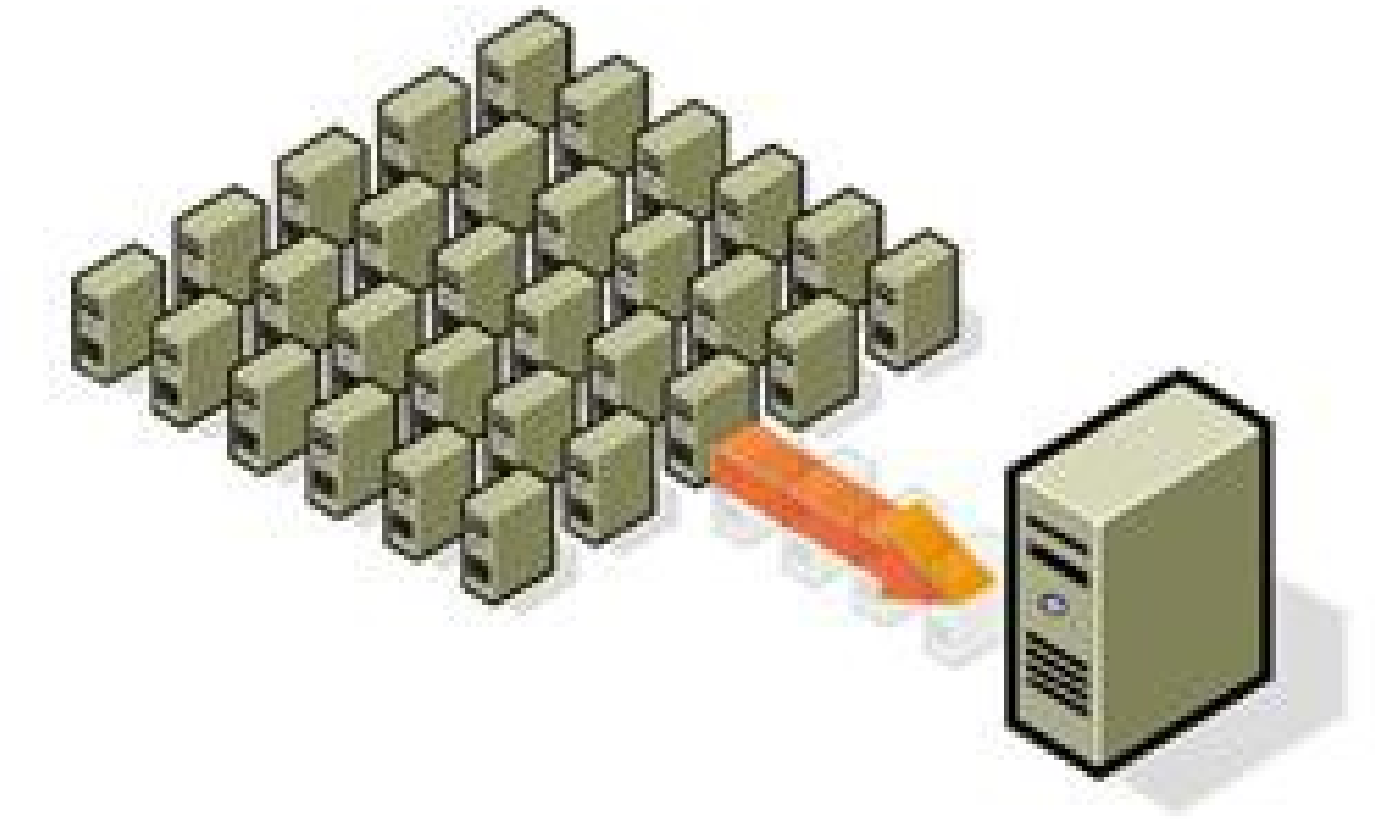
Network Startup Resource Center

UNIVERSITY OF OREGON

Before We Start...

- Today's discussion is by no means an exhaustive discussion of the security implications of virtualization
- Recommendations for securing infrastructure may or may not fit in your environment
- If you have any questions there may be time after the talk, otherwise, please feel free to approach me after

Virtualization?



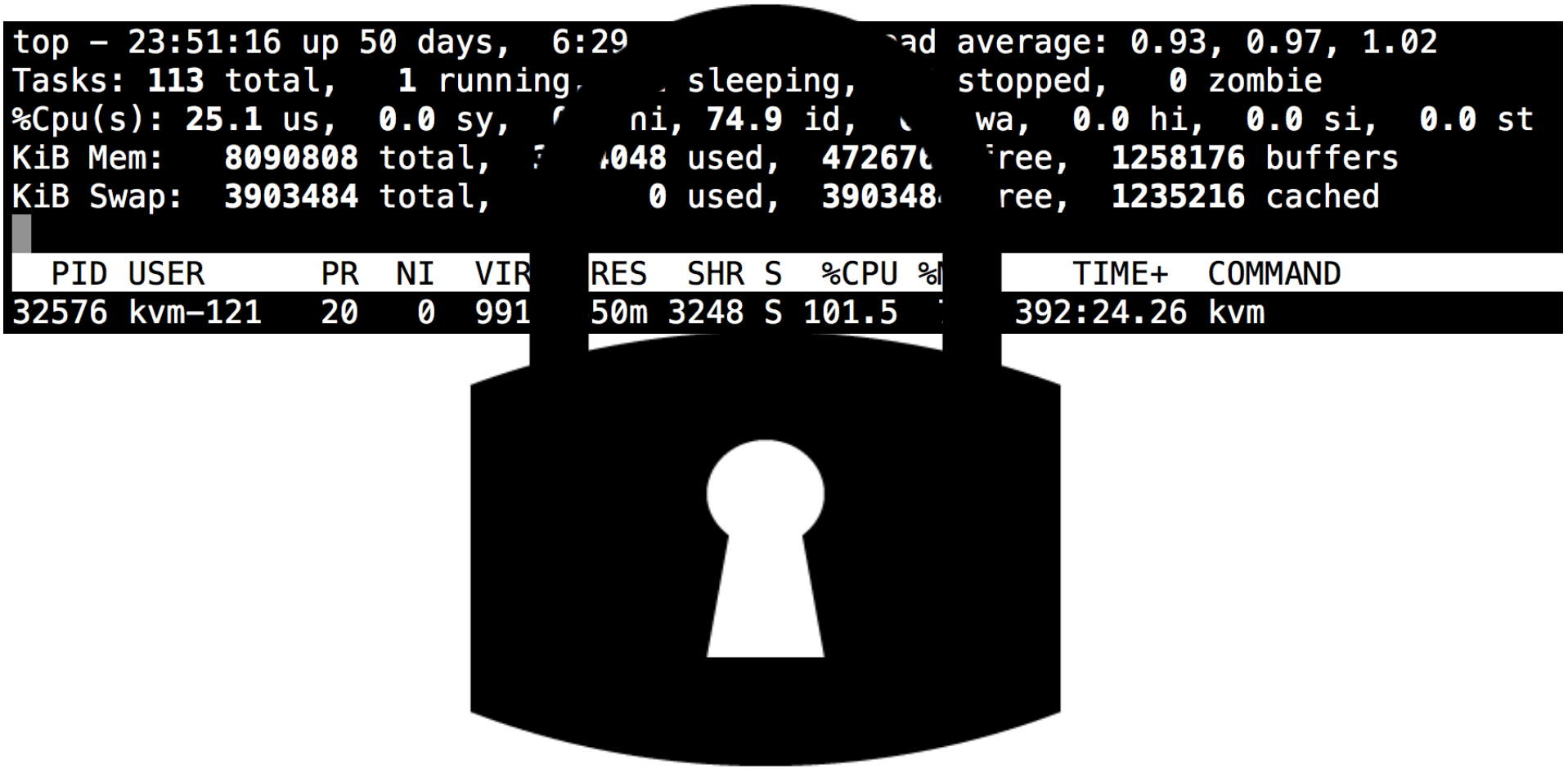
Why?

- Consolidation
  - Most systems are under-utilized, especially the CPU is idle for much of the time
  - Do more work with less hardware
  - Reduced space and power requirements
- Management
  - Snapshot/restore, cloning, migration
  - Increased isolation between services

Servers...

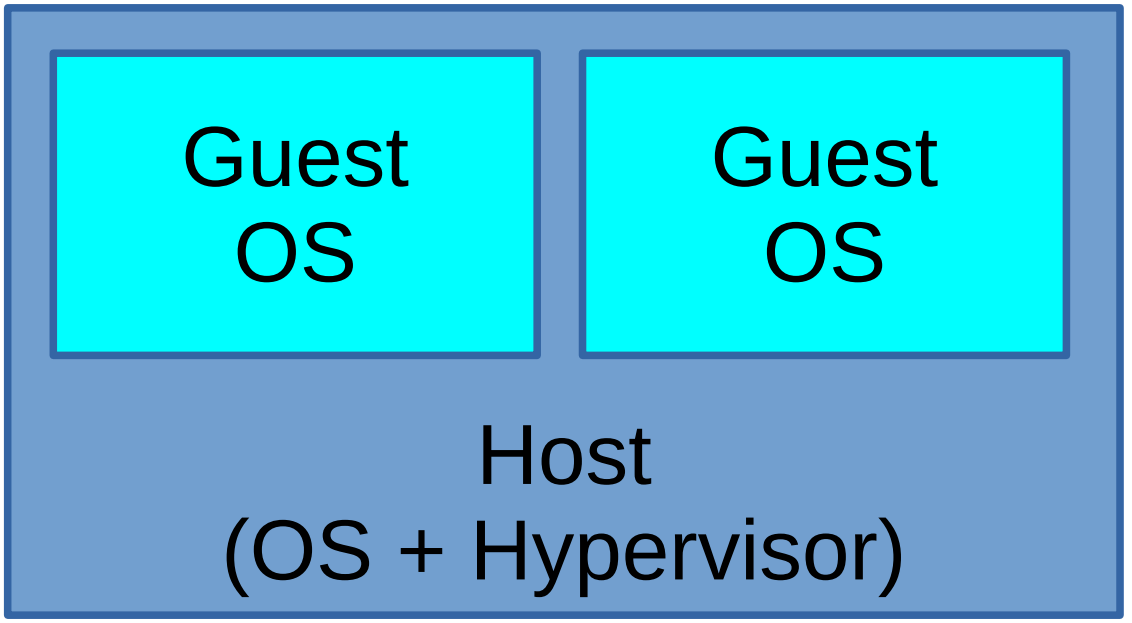
Servers.

Where does the lock go?



Terminology

- The host is the physical machine running the virtual machine
- The guest is the emulated (virtual) machine
- One host could be running many guests



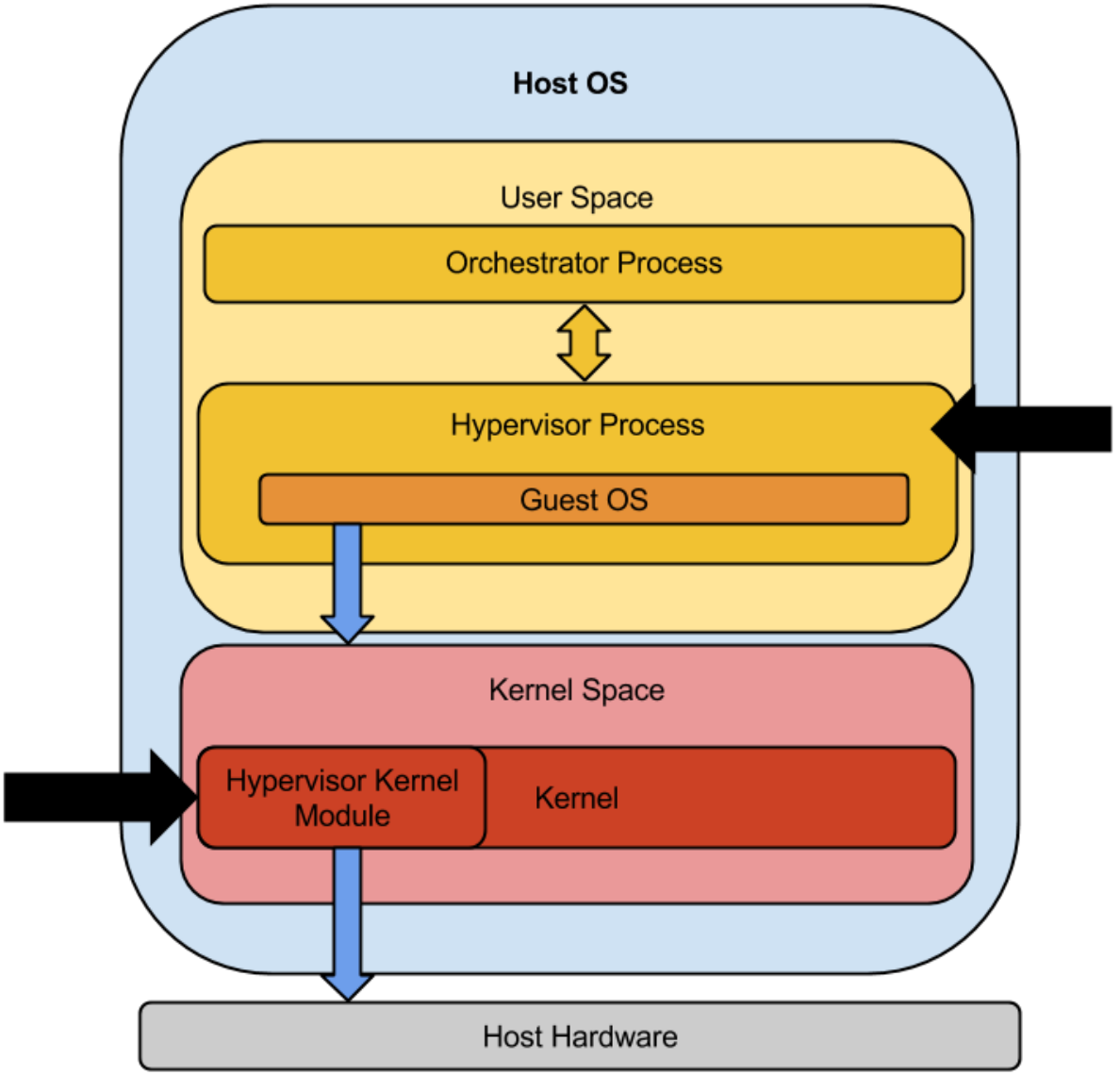
Hypervisor

- Hypervisor emulates hardware for guest
- Hypervisor allocates some real system RAM to each VM, and shares the CPU time

Orchestrator

- APIs for administering guest operations
  - Start / Stop
  - Creation / Destruction
  - Failover / Migration
- Examples
  - Libvirt
  - Ganeti

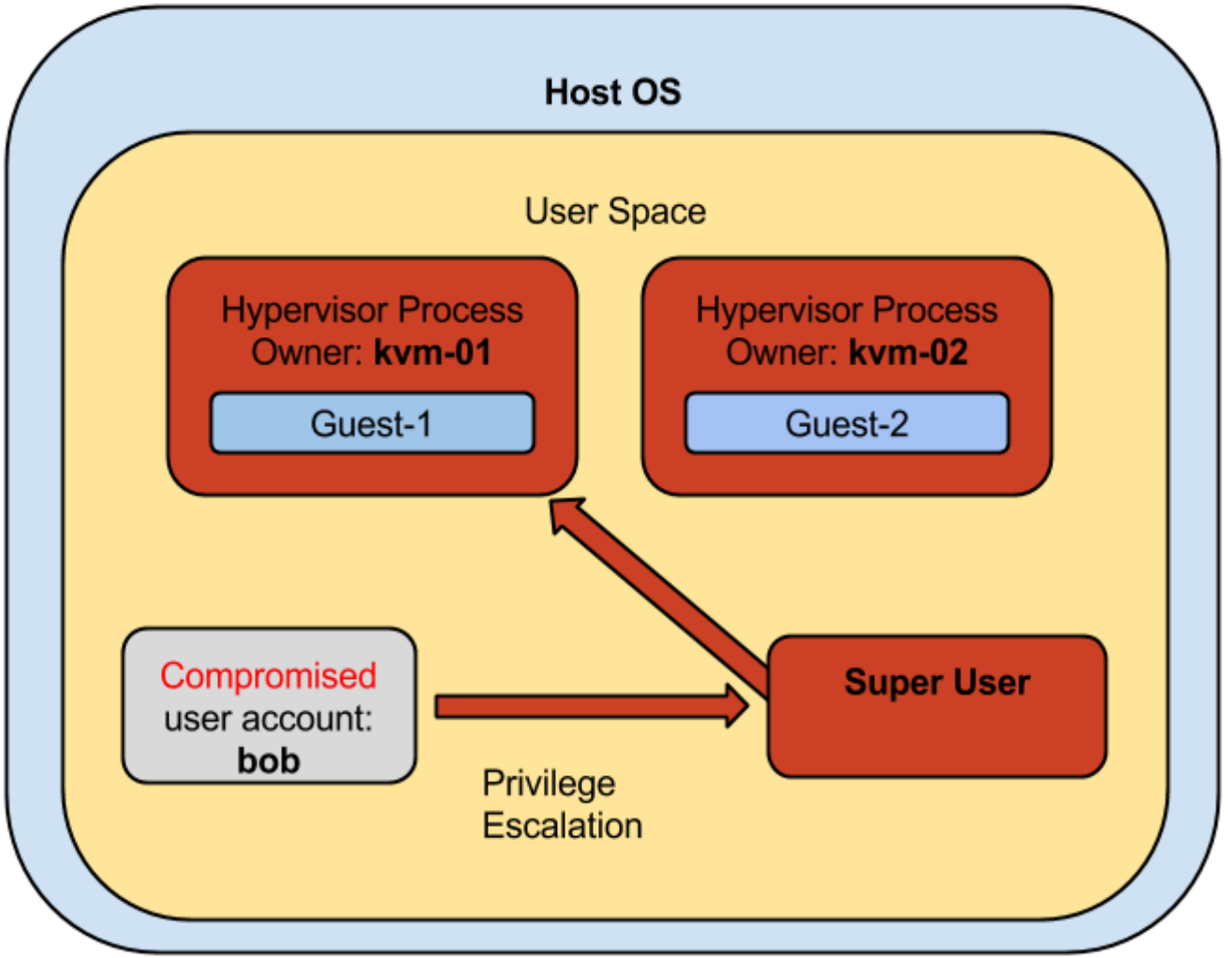
Visualizing the Virtualization Stack



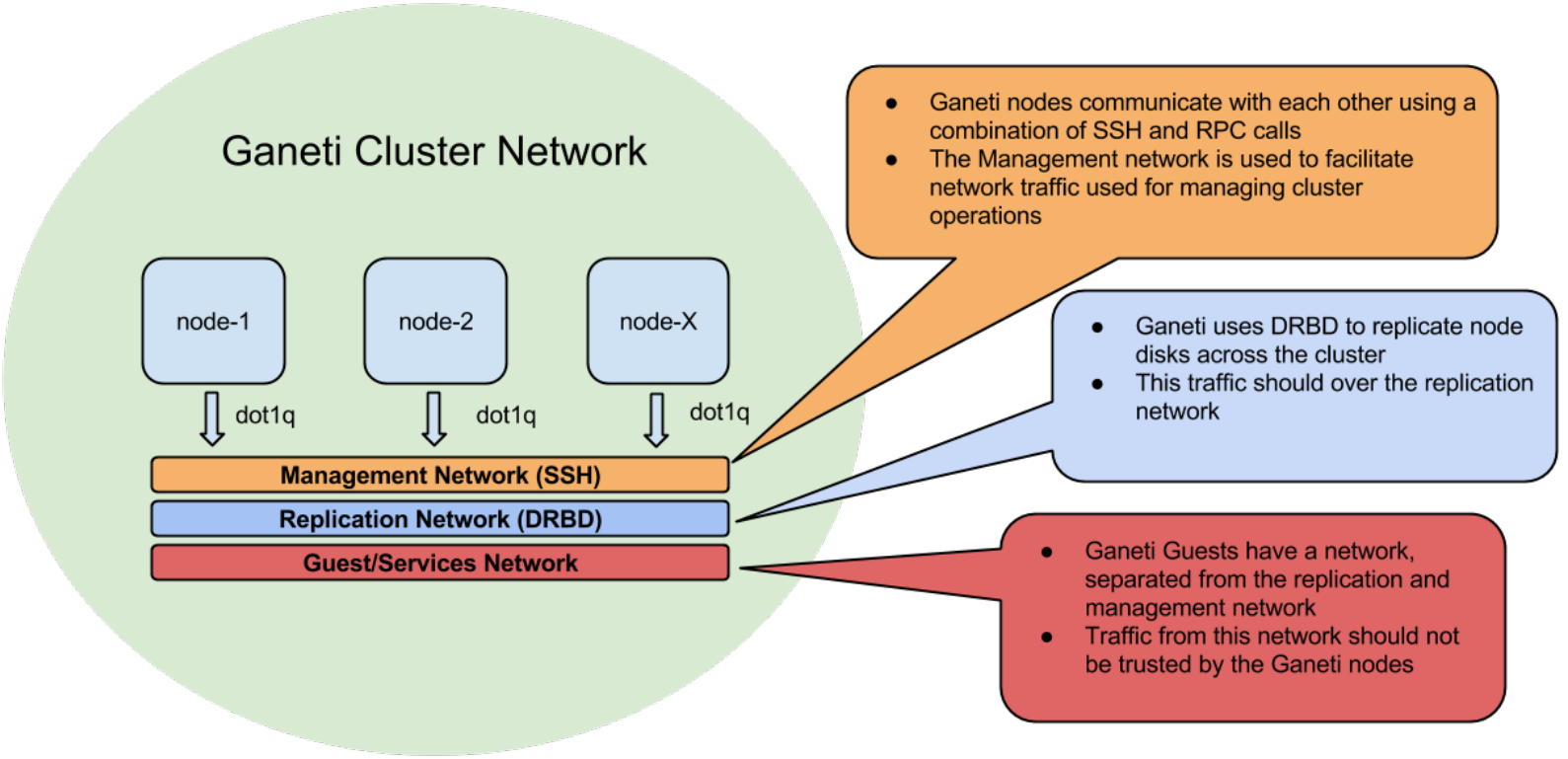
Threat Model: What is possible?

- **Privilege Escalation**
  - Exploit that allows unprivileged subject access to guests through hypervisor, orchestrator services
- Denial of Service
  - Attacker denies other users access to shared resources

Example:
Escalate User Privileges, Access Guests



Example: Guest Access Orchestrator API

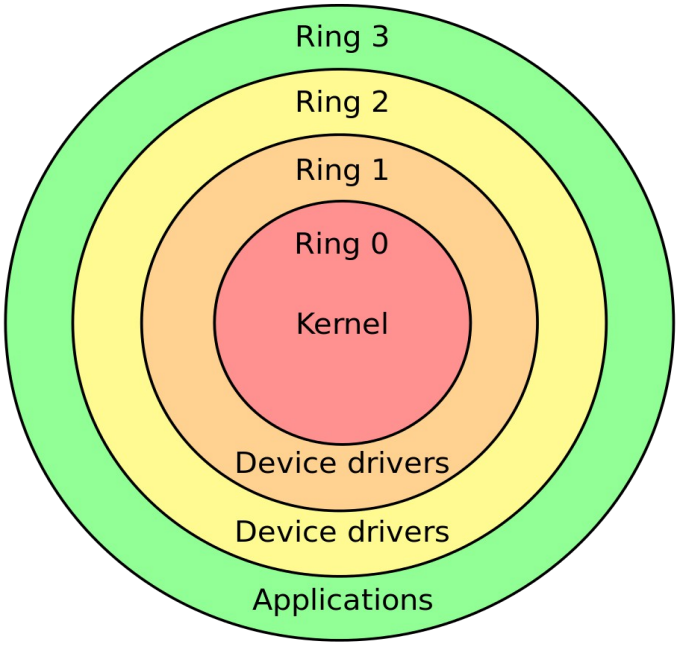


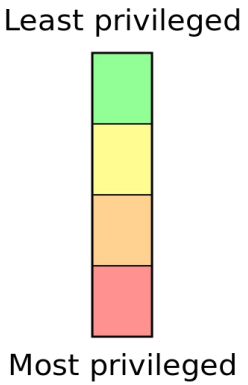
VM Escape

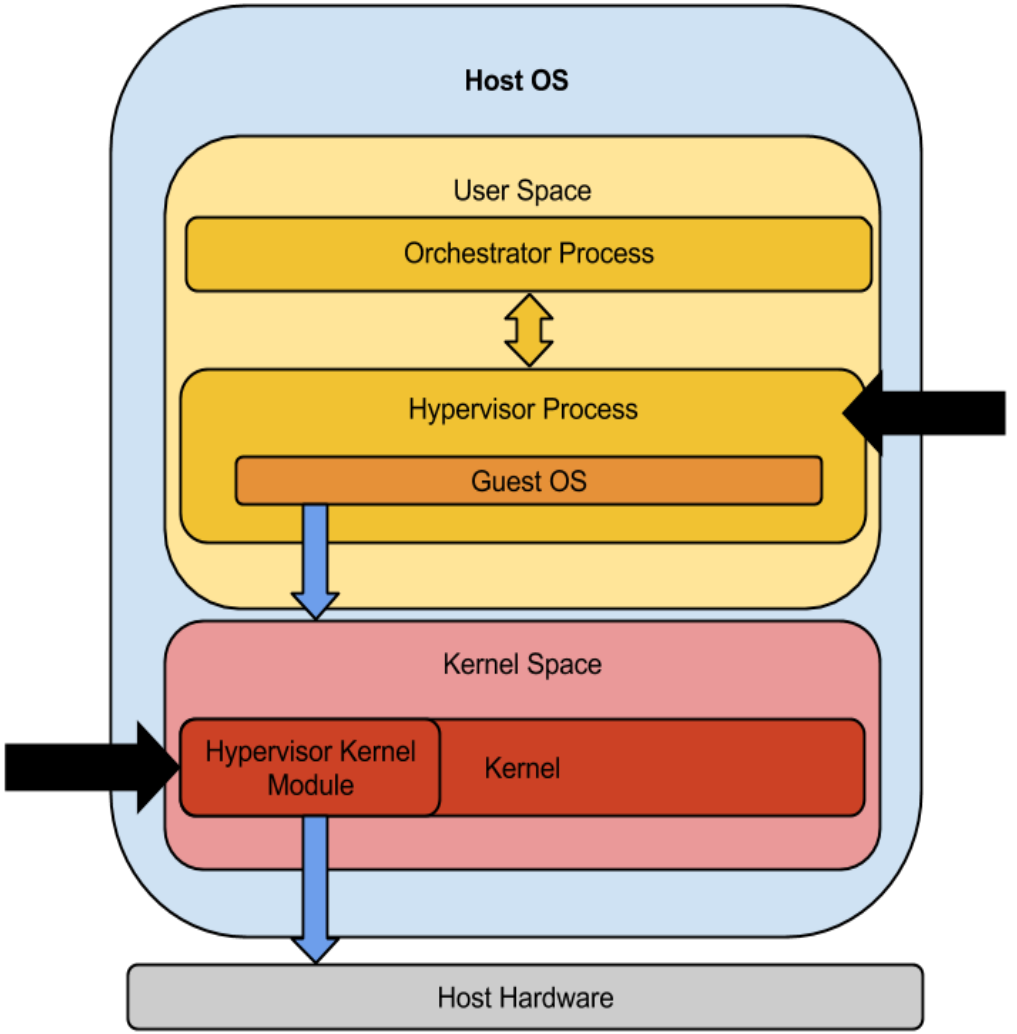
Breaking out of Guest and interacting with and/or executing code on the host.

- Spawn a shell
- Open a network backdoor
- ...

Guests vs. Hypervisor





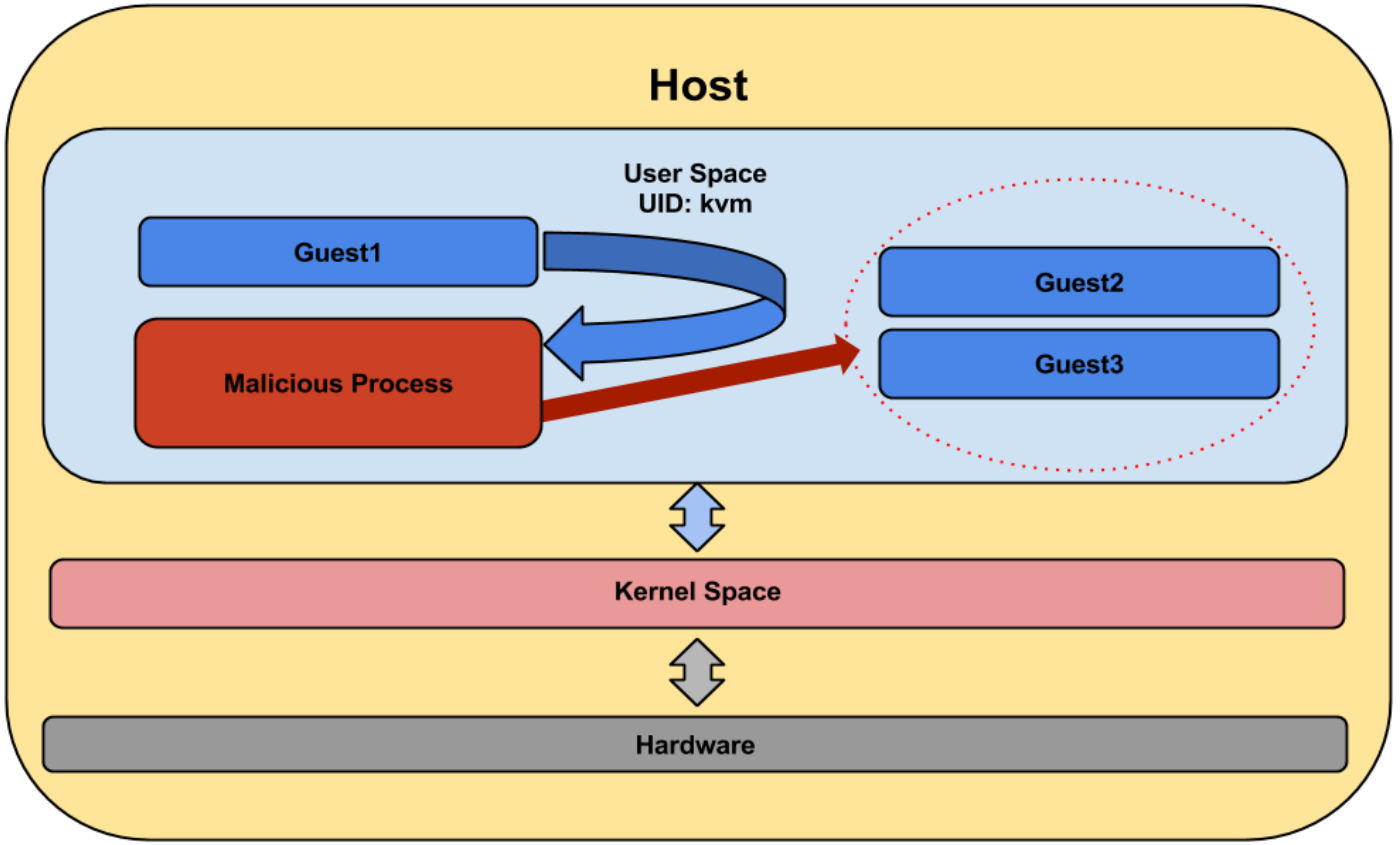


CVE-2015-3456, “Venom”

- Flaw in emulator's implementation of Floppy controller
- Doesn't require the presence of a floppy drive on the system, just access to send I/O to controller
- Allows Guest user to execute arbitrary code in “user” space on Host

Escape to User Space

- Attacker can execute code and access resources as the owner of the **hypervisor** process

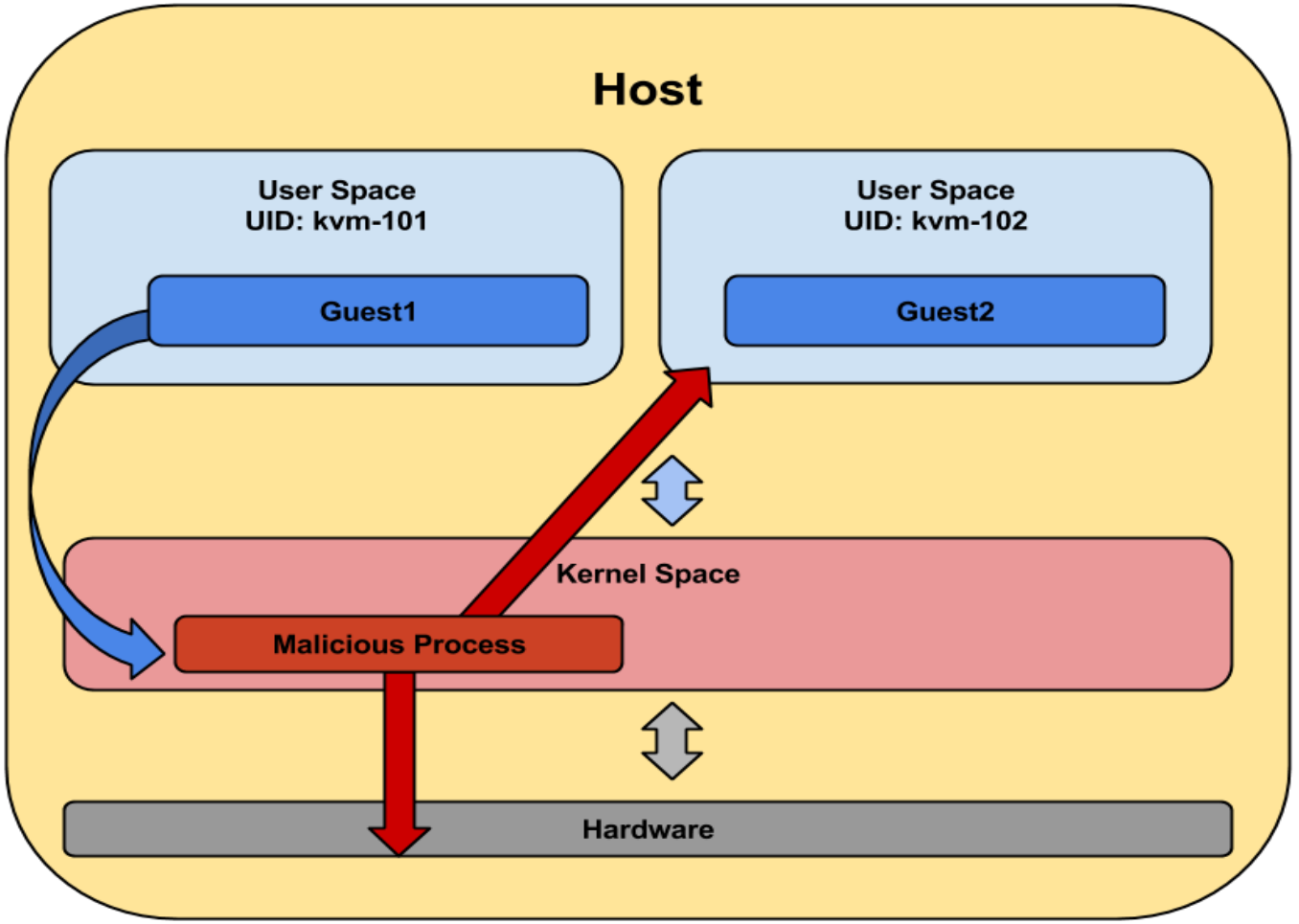


CVE-2012-0217 kernel: x86_64

- Affects Xen Hypervisor
- When a **guest** is run paravirtualized, it runs a modified kernel that passes some instructions directly to **host** kernel
- Flaw in system call in **host** kernel that allows **guest** to execute arbitrary code in kernel space on **host**

Escape to Kernel Space

- Attacker can execute code and access resources as the owner with Kernel privileges

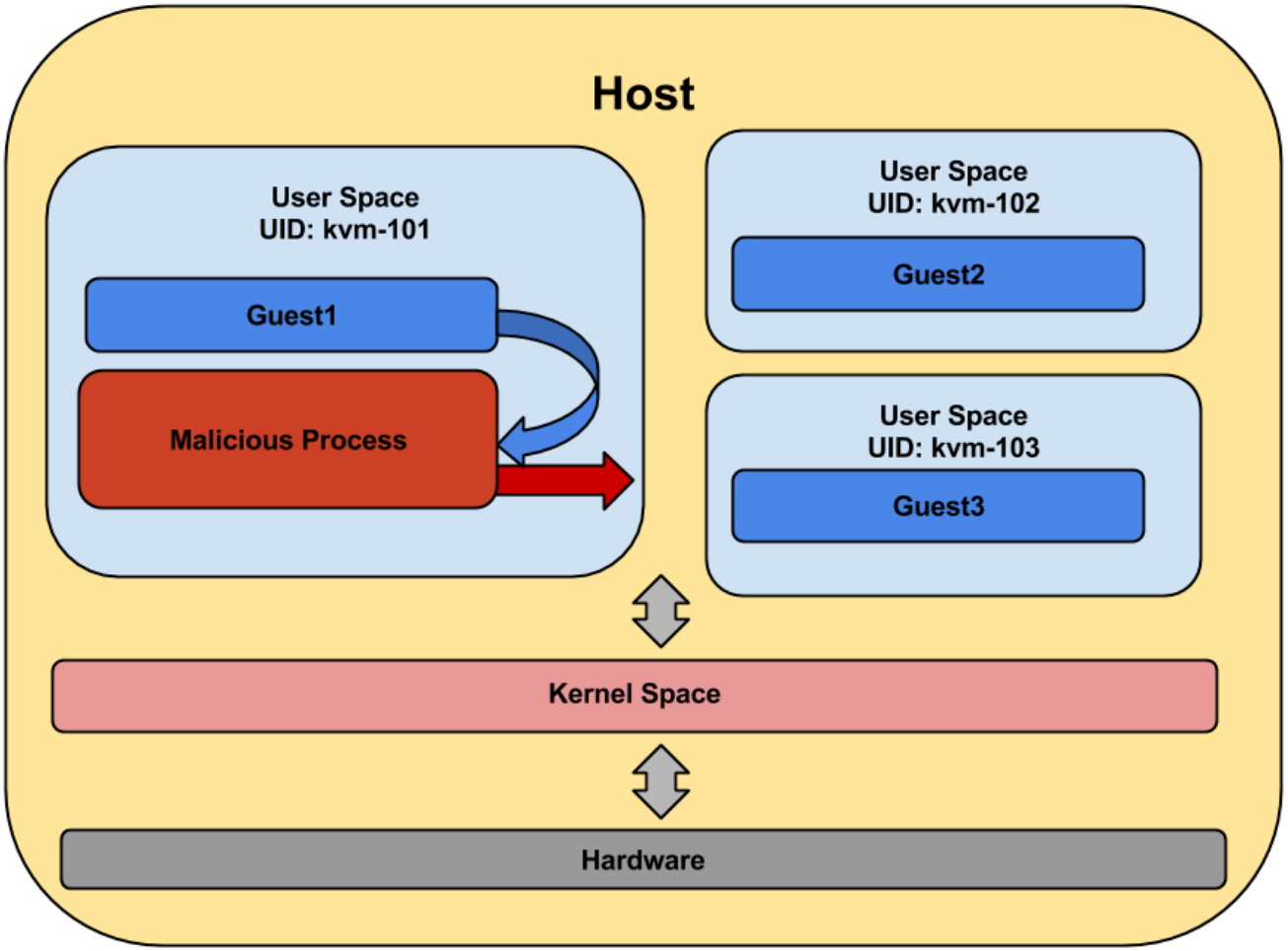


Privilege Escalation Mitigation

Segregate Guest Execution Space

- Execute VM's as non-privileged, service users
  - No home directories
  - No password
  - No Shell
- Allocate one user account per guest
  - In the case of VM Escape, every guest is isolated

Segregate VM Execution Space



Discretionary Access Control (DAC)

- Linux's “stock” access control policy is referred to “discretionary access control”
- “Restricts access to objects based on the identity of subjects and/or groups to which they belong.”
- Allows privilege sets to be inherited from parent process

Strong DAC with Libvirt

- Can set a per-host default setting in the **/etc/libvirt/qemu.conf** configuration file via the user=$USERNAME and group=$GROUPNAME parameters. When a non-root user or group is configured, the libvirt QEMU driver will change uid/gid to match immediately before executing the QEMU binary for a virtual machine.

POSIX Capbilities, Libvirt

- The libvirt QEMU driver has a build time option allowing it to use the libcap-ng library to manage process capabilities. If this build option is enabled, then the QEMU driver will use this to ensure that all process capabilities are dropped before executing a QEMU virtual machine.
- The Linux capability feature is thus aimed primarily at the scenario where the QEMU processes are running as root.

Mandatory Access Control (MAC)

- Modifications to Linux Kernel that apply mandatory, system level access control above and beyond discretionary policy
- Requires strict definition of allowed access
- Integration in common virtualization orchestrator library Libvirt

SELinux/AppArmor + Libvirt

- By default, SELinux + libvirt provides protection between host and guest
- No protection between guests

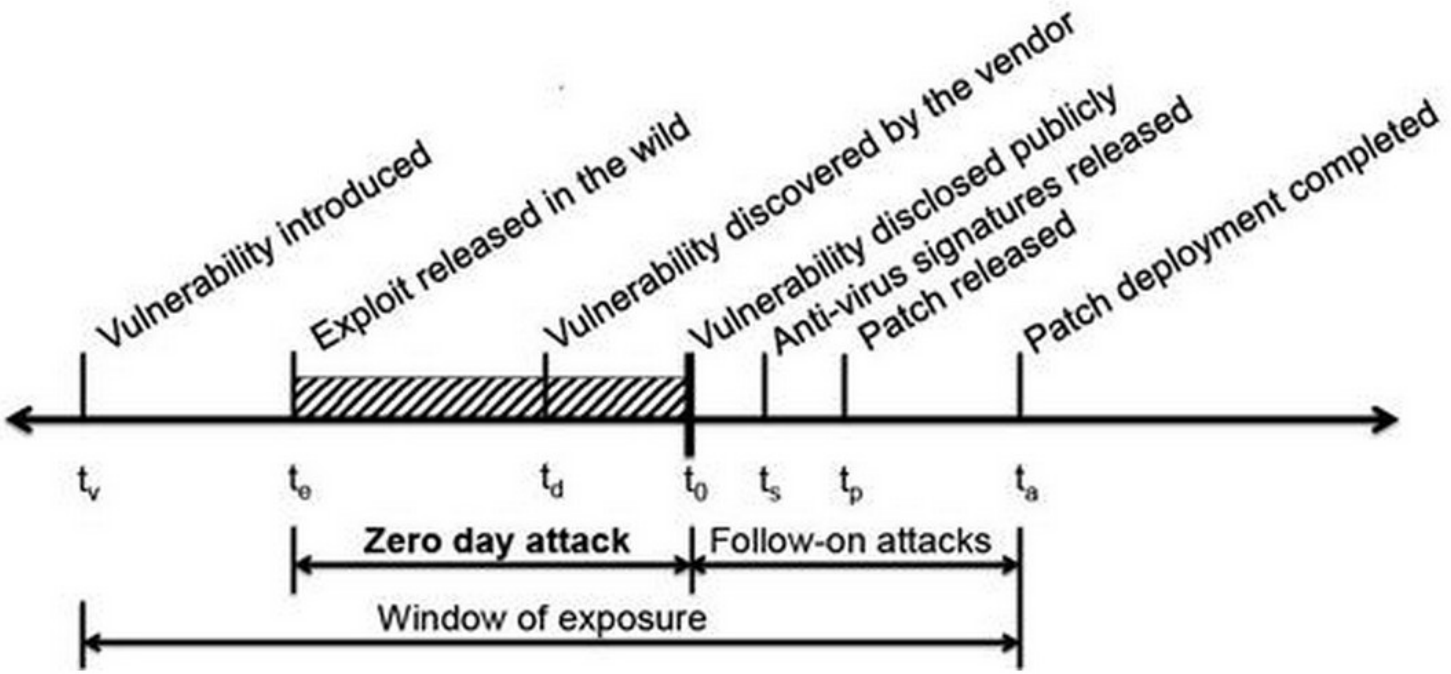
SELinux/AppArmor + Libvirt + sVirt

- Each QEMU virtual machine runs under its own confined domain
- Prevents one QEMU process accessing any file resources that are prevent to another QEMU process

Patch your systems!

- Hypervisor vulnerabilities are mitigated by patches to the hypervisor software

Mind the Vulnerability Timeline



http://blog.coresecurity.com/2013/02/27/a-world-of-vulnerabilities-guest-blog-post-from-infosec-institute/

Audit System Logs

- Audit privilege escalation
  - Sudoers
  - Root logins
- Audit orchestrator events
  - Start / Stop / Restart Guest
  - Attach / detach storage
  - Changes to network interfaces

Segregate Services

- Apply the principle of least privilege:
  - Which users need access to the host? Guests?
  - What network communication is strictly necessary?
  - What communication channels could jeopardize hypervisor?
  - What communication channels could jeopardize data?

Harden Network Services

- Enable authentication and encryption for remote access protocols
  - VNC
  - Orchestrator APIs
- Isolate Guest network from Host network
- Scope the listening interfaces for network services where possible

DoS Mitigation

Controlling Resource Consumption

- Know your service's resource needs
- Intelligently allocate resources and create policy for limiting the potential for resource exhaustion
  - Distribute resource intensive operations across physical hardware
  - Be careful not to over-subscribe resources
- Leverage tools like **virtio** that assist your virtual machines in allocating resources intelligently

Controlling Resource Consumption

- Depending on your platform, you have a number of options for controlling resource utilization
  - Relative controls – Control the priority of resource allocation in relation to other virtual machines.
  - Absolute controls – Control the absolute amount of a resources allocation

cgroups for Resource Management

- Used to allocate and segregate resources among containers and VM by libvirt, lxc, etc.
- Options including:
  - **cpuset** - assigns individual CPUs and memory nodes to cgroup
  - **cpu** - schedules CPU access to cgroups
  - **memory** - generates automatic reports on memory resources used by the tasks in a cgroup, and sets limits on memory use of those tasks
  - **blkio** - controls and monitors access to I/O on block devices by tasks in cgroups
  - **devices** - allows or denies access to devices by tasks in a cgroup

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide

Summary

- Virtualization has many benefits but there can be many security risks if managed improperly
- Most vulnerabilities arise as a result of sharing hardware
- Patching accompanied by strong access control around the hypervisor and orchestrator can limit the damage caused by privilege escalation and DoS
- Auditing and monitoring orchestrator and OS logs will help you know when and if you are being attacked and perhaps even if the attack was successful