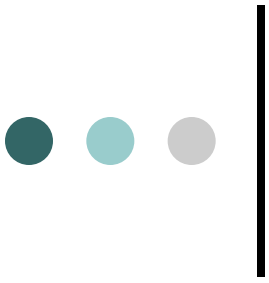


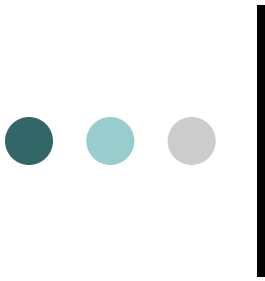
Lecture 4

Virtualization



Chapter 7

Cloud Technologies Basics



Disk Images and VM Images

- A disk image is a copy of the entire contents of a storage device, for example a hard drive or DVD.
- A virtual machine image is a single file which contains a virtual disk. The virtual disk has a bootable operating system installed on it.



Disk Images and VM Images (cont'd)

- Raw-- a bit-for-bit copy of the data of either a disk or volume, with no additions or deletions
 - the “dd” command in Linux will create this
 - (similar tools have been created for Windows)
 - this is an unstructured disk image format



Disk Images and VM Images (cont'd)--vhd

- Virtual Hard Disk (vhd)—originally used by Virtual PC and Windows Virtual Server, see MSDN (2016), now an open format
 - Minimum size is 3MB
 - Versions are:
 - Fixed
 - Maximum size is pre-allocated
 - Expandable/Dynamic/Dynamically Expandable/Sparse
 - Uses only the space needed to store the actual data.
 - The VHD API doesn't check to make sure the physical disk is big enough so it's possible to create an expandable disk that's too big for the physical disk
 - Maximum size is 2,040 GB
 - Differencing
 - A parent virtual disk is unchanged, changes to the parent are stored in a separate child image
 - Maximum size of a differencing virtual disk is 2,040 GB



Disk Images and VM Images (cont'd)—vhd (cont'd)

- Virtual Machine Disk (vmdk)—originally developed by VMware
 - Maximum VMDK file size is 2 TB
 - In Virtual Disk Format version 5, disk volume extent is approx.. 60 TB.
 - Supports more than 100,000 files per volume
 - All disk space needed for a virtual disk's files may be allocated at the time it is created, or it can grow as needed to accommodate new data



Disk Images and VM Images (cont'd)—vdi

VirtualBox Disk Image (vdi)

- originally developed by VirtualBox
 - Fixed size
 - Size allocated up front
 - Better performance than with dynamically expanding
 - Dynamically expanding
 - Created at minimal size, but grows automatically
 - Can have slower performance (if disk is frequently enlarged).



Disk Images and VM Images (cont'd)—iso

iso image—represents the contents of an optical disc (CDs, DVD, etc.).

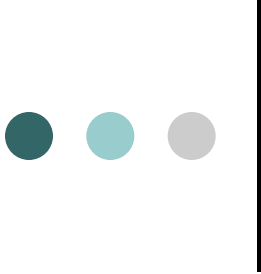
- Data formats supported are specified by:
 - CDs—originally from ISO 9660
 - DVD and BluRay—UDF format, specified in ISO/IEC 13346 and ECMA-167



Disk Images and VM Images (cont'd)— qcow

QEMU Copy on Write (qcow2, qcow)—
originally used by QEMU hypervisor

- qcow2 replaces the original qcow
 - qcow2 supports multiple snapshots
- can grow dynamically as data is added



Disk Images and VM Images (cont'd)— qcow (cont'd)

- Benefits over using raw image include:
 - Smaller file size (raw disk images allocate entire space to a file even if parts of the space are empty)
 - Copy-on-write support,
 - Copy on Write (COW) means:
 - If there are two or more users that need very similar resources, then initially they are given pointers to the same resource. When a user tries to modify its own resource, only then is a private copy created. If no modifications are made, then no copies need be created.
 - Snapshot support, where the image can contain multiple snapshots of the images history
 - Optional zlib compression
 - Optional AES encryption



Disk Images and VM Images (cont'd)— Amazon images—ami, aki, ari

- Machine Images on Amazon
- These consist of:
 - Amazon Machine Image (ami)
 - Amazon Kernel Image(aki)
 - Amazon Ramdisk Image (ari)



Disk Images and VM Images (cont'd)— Amazon images—ami, aki, ari

- According to Amazon Discussion Forum (2011):
 - “The AKI represents the vmlinuz portion of the kernel. It is basically the compiled kernel that gets loaded on boot.
 - The ARI represents the initrd/initramfs. This is the ramdisk that gets loaded with the kernel and has the initial driver modules for the kernel to find the root filesystem.
 - With traditional AKIs/ARIs (you need to match those) you will also need the corresponding kernel modules installed on the file system. Without these the system will normally not be able to boot. You can run a newer AKI on an older system if you have the right kernel modules installed.
 - The NEW way:
 - We launched a set of AKIs called PVGrub. These simulate grub as the kernel and allow you to install your kernel on the AMI and have it perform similar to bare-metal system where it reads the kernel and ramdisk from the filesystem. These PVGrub AKIs do not use ARIs. In fact there are some edge case where using an ARI will prevent PVGrub AKIs from working, but they are rare.”



Disk Images and VM Images (cont'd)— Amazon images—ami, aki, ari

According to Debian (2016):

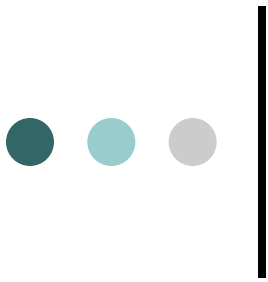
- “An AKI (Amazon Kernel Image) is a preconfigured bootable kernel mini image, that is prebuilt and provided by Amazon to boot instances. Typically one will use an AKI that contains pv-grub so that one can instantiate an instance from an AMI that contains its own Xen DomU kernel that is managed by the user.”
 - pv-grub is a paravirtual boot loader
 - A boot loader is the first software program that runs when a computer starts. It loads the operating system kernel and then transfers control to it.
-



Disk Images and VM Images (cont'd)--ovf

Open Virtualization Format (ovf)

- packages one or more image files and an XML metadata file (.ovf)
 - containing information about the virtual machine
- may also package other files



Disk Images and VM Images (cont'd)— OpenStack Glance

- OpenStack Glance (2016) supports the following disk image formats:
 - raw
 - vhd
 - vmdk
 - vdi
 - iso
 - qcow2
 - aki
 - ari



Just what are Hypervisors and Virtual Machines, Anyway?

Virtual Machine

- emulator of a particular computer system

Two kinds of virtual machines:

- System virtual machine/full virtualization:
- Process virtual machine/application virtual machine/Managed Runtime Environment (MRE)



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

System virtual machine/full virtualization:

- allows execution of a complete OS
- a full up emulator of a particular computer architecture/hardware system

Process virtual machine/application virtual machine/Managed Runtime Environment(MRE):

- Allows a single application (program) to run, provides a platform independent execution environment
- Examples include the Java Virtual Machine (JVM) and the .NET Common Language Runtime (CLR)



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

- A Hypervisor creates and runs virtual machines.
- TechTarget (2016) defines a hypervisor as follows:
 - “A hypervisor, also called a virtual machine manager, is a program that allows multiple operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and other resources all to itself. However, the hypervisor is actually controlling the host processor and resources, allocating what is needed to each operating system in turn and making sure that the guest operating systems (called virtual machines) cannot disrupt each other.”



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

- Hypervisors come in two main types:
 - Type 1 (native, bare metal)
 - Runs directly on host hardware
 - Controls the hardware and manages the guest OS
 - Type 2 (hosted)
 - Runs within a conventional OS environment



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

- A “Host machine” is the physical host
- A “Guest machine” or “guest VM” is the VM (virtual machine) installed on top of the hypervisor



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Hypervisors can be categorized based on the kind of virtualization they provide as follows:

- Full virtualization
- Paravirtualization



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

With full virtualization, the guest operating system thinks it is running direct on the hardware:

- the hypervisor provides hardware emulation that is a complete simulation of the hardware that the guest operating system expects
- any calls to the underlying hardware must be trapped and emulated
- thus the guest operating system does not have to be modified in any way.



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

With paravirtualization, the kernel of the guest operating system has been modified

- so that instead of handling calls to the hardware itself using a privileged instruction:
 - a privileged instruction
 - a special assembly language instruction that can only be used when the processor is in a special supervisory mode (generally called ring 0)
 - normally only available to the operating system
- it calls the hypervisor instead to do this work.
 - These calls to the hypervisor are called *hypercalls*



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Hardware Assisted Virtualization

- a type of full virtualization
- processor manufacturers (such as Intel and AMD) implemented an extra-supervisory mode (called ring -1)
- this allows hypervisors to run at ring-1, and thus privileged instructions in the operating system now trap automatically to the hypervisor



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

- Hybrid Virtualization combines paravirtualization with hardware assisted virtualization
 - parts of the guest operating system use paravirtualization for certain hardware drivers
 - but hardware assisted virtualization (or full virtualization if hardware assisted) is not available



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Examples of Type 2 Hypervisors:

- Oracle VirtualBox
- QEMU (Quick Emulator)

.



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Oracle VirtualBox is an open source hypervisor

- can use hardware-assisted virtualization
 - when the hardware supports it
- supports paravirtualization for some Linux and Windows guests
 - can improve performance



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

QEMU has two basic operating modes:

- full system emulation
- user mode emulation



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

QEMU full system emulation:

- Emulates a full system including processors and associated peripherals
- Can be used to launch different operating systems without rebooting host PC
- supports the emulation of numerous different kinds of hardware and associated peripherals. These include:
 - PC
 - MIPS
 - ARM
 - Sparc32
 - Sparc64



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

- QEMU User mode emulation:
 - can launch one process that was compiled for one CPU on another CPU
 - only provides a subset of full system emulation
 - assumes the host system is doing some of the work
- QEMU emulation libraries are used with individual binaries,
 - these think the host computer is their original computer
 - see directories, etc., that they would have expected if running on the original computer.



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

QEMU also has two hosted modes, with:

- KVM
- Xen



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Type-1 hypervisors include:

- Kernel Virtual Machine (KVM)
- Xen



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Kernel-based Virtual Machine (KVM)

- ships as part of the Linux kernel.
- truly is a type-1 hypervisor (runs on bare metal)
 - because when launched, takes over the hardware, but still works with the Linux kernel for processing
- When running on x86 hardware ring -1 instructions
- KVM began on x86 but has been ported to other processors, including ARM, MIPS, and PowerPC



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

- KVM by itself does not perform hardware emulation.
- A common operating mode for KVM is to combine it with QEMU
- RedHat provides a commercial product, Red Hat Enterprise Virtualization (RHEV), that is based on KVM



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Xen hypervisor

- originally developed at the University of Cambridge and released in 2003
- original mode was paravirtualization
 - the concept of paravirtualization was originally introduced in Xen
- has since added hardware-assisted virtualization
 - in Xen known as a Hardware Virtual Machine
- When running on Xen, QEMU is used only for hardware emulation



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Xen supports the following operating modes:

- Paravirtualization (PV)
- Hardware Virtual Machine (HVM)
- PVHVM
 - HVM guests using special paravirtual device drivers
 - these optimized drivers bypass emulated disk and network I/O and thus improve performance.
- PVH
 - a PV guest that uses PV drivers to boot and for I/O.
 - otherwise uses HW virtualization extensions, without the need for emulation



Just what are Hypervisors and Virtual Machines, Anyway? (cont'd)

Xen creates domains:

- unprivileged domains, called domUs
- privileged domain, called dom0. The dom0 domain:
 - includes trusted kernel and hardware drivers
 - controls other domains, this includes:
 - creating
 - destroying
 - saving, etc.
 - controls network and peripherals
 - these are assigned to kernel or to domUs
 - when using paravirtualization, dom0 must include a paravirtualized-ready kernel
 - several Linux kernels now support being used as a Xen dom0 kernel



libvirt

libvirt is a management tool and API used to manage hypervisors:

- A C library but with bindings to other languages
- Supports several hypervisors, including:
 - KVM/QEMU
 - Xen
 - VMware ESX
 - VMWare Server/GSX,
- there are several GUIs that interact with libvirt
 - one example is virt-manager (Virtual Machine Manager)



Software Defined Networking and Network Virtualization

There is some argument about whether or not there is a difference between:

- Software Defined Network
- Network Virtualization
- (also possibly) Network Functions Virtualization



Software Defined Networking and Network Virtualization (cont'd)—first of all, what is “tunneling”

According to Enterprise Networking Planet (2006):

“A tunnel is a mechanism used to ship a foreign protocol across a network that normally wouldn't support it. Tunneling protocols allow you to use, for example, IP to send another protocol in the "data" portion of the IP datagram. Most tunneling protocols operate at layer 4, which means they are implemented as a protocol that replaces something like TCP or UDP.”



Software Defined Networking and Network Virtualization (cont'd)

Garrison (2014) discusses *network virtualization* as follows:

“Network virtualization literally tries to create logical segments in an existing network by dividing the network logically at the flow level (it is similar to partitioning a hard drive).

NV is an overlay; it's a tunnel. Rather than physically connecting two domains in a network, NV creates a tunnel through the existing network to connect two domains. NV is valuable because it saves administrators from having to physically wire up each new domain connection, especially for virtual machines that get created.”



Software Defined Networking and Network Virtualization (cont'd)

Garrison (2014) discusses *network functions virtualization* as follows:

“If NV offers the capability to create tunnels through a network and use per-flow service thinking, the next step is to put a service on a tunnel. NFV is virtualizing Layer 4-7 functions such as firewall or IDPS, or even load balancing (application delivery controllers).”

and further:

“If you have a specific tunnel you’re punching through the infrastructure, you can add a firewall or IDS/IPS to just that tunnel.”



Software Defined Networking and Network Virtualization (cont'd)

Garrison discusses *software defined networks* as follows:

“While NV and NFV add virtual tunnels and functions to the physical network, SDN changes the physical network, and therefore is really a new externally driven means to provision and manage the network.”



Software Defined Networking and Network Virtualization (cont'd)

However, Baldwin (2014) says:

“So there’s really no reason to introduce fear, uncertainty, and doubt about a supposed difference between SDN, NFV, and network virtualization. Because there isn’t one.”



Software Defined Networking and Network Virtualization (cont'd)

McCouch (2014) says *Software Defined Networking* means separating a data network's control functions from its packet forwarding functions

- independently developed products can easily work together
- the user can mix and match vendors.
- separating control functions (the control plane) allows an easy way to configure and monitor a large heterogeneous network
- allows network programmability
- allows APIs to control the network.



Software Defined Networking and Network Virtualization (cont'd)

- McCouch (2014) says further that Network Virtualization refers to isolation of applications or tenants through creating virtual instances of a physical device
 - virtual routers and switches can be created
 - can be used to create logical networks (for ex., VLANs)
 - Network Functions Virtualization consists of running a function such as a firewall or load balancer in virtual machines on the virtual server infrastructure



Software Defined Networking and Network Virtualization (cont'd)

The Open Networking Foundation (2016) defines *Software Defined Networking* as follows:

“SDN is a new approach to networking in which network control is decoupled from the data forwarding function and is directly programmable.”



Software Defined Networking and Network Virtualization (cont'd)

The Open Networking Foundation (2016) says further:

- software defined networking architecture decouples the network control and forwarding functions
 - enables the network control to become directly programmable
 - allows the underlying infrastructure to be abstracted
- The SDN architecture is therefore:
 - directly programmable
 - agile
 - centrally managed
 - programmatically configured
 - open standards-based and vendor-neutral



Software Defined Networking and Network Virtualization (cont'd)

For our own definition, we will combine the three terms:

- use the term *Software Defined Network* loosely to refer to a virtual network
 - includes physical connections and switches
 - Includes logical connections and switches
- data plane (packet forwarding) will be separated from the control plane (decisions on where packets will be forwarded)
- control plane is controlled programmatically through an API
 - so the behavior of the network can change dynamically
- VLANs will include both physical and logical (virtual) connections
- Additional services can be applied to switches or connections
 - including load balancing and firewall support



Open vSwitch/OpenFlow and Linux Bridge

For a Virtual Machine (VM) to do useful work, typically it must be connected to a physical network, and also to other VMs

- This must be done through the hypervisor.
- For Linux-based hypervisors, in the past the Linux bridge was typically used
 - the Linux bridge is a virtual switch that is included in the linux kernel



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

More recently, Open vSwitch has come to be often used instead of a Linux bridge

Open vSwitch is targeted at networks that include multiple servers and run virtual machines in a highly dynamic environment:

- a cloud environment is a typical example of the networking characteristics that Open vSwitch was intended to address



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

Open vSwitch:

- allows migration of live network state between different hosts
- supports monitoring of network events
 - this uses the Open vSwitch Database (OVSDB) which stores network state and supports remote triggers.
 - this allows Open vSwitch to react to and track network events such as VM migrations.
- includes support for offloading packet processing to hardware chipsets



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

Open vSwitch supports various kinds of monitoring:

- NetFlow
- sFlow
- SPAN, and RSPAN



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

Cisco NetFlow:

- examines IP packets at an interface for IP traffic flow analysis including:
 - network usage
 - traffic routing
 - security
- works by aggregating multiple packets into a flow (packet sequence) then analyzing the flow
 - however, most of the data field is lost in this aggregation
 - mainly contains source and destination IP addresses, protocols, type, QoS, etc.
 - later versions can also examine layer 2



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

sFlow

- a packet sampling technology
- provides continuous statistics on any protocol (L2, L3, L4, and up to L7)
- can either:
 - monitor interface counters and CPU usage
 - capture the first 256 bytes (quantity of bytes configurable) of each frame from 1 in N (with N configurable) frames
- information is sent in a UDP datagram to a collector to be analyzed



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

SPAN, and RSPAN

- mirror traffic from one interface on a switch to another interface on the same switch (layer 2)
- Remote SPAN (RSPAN) mirrors traffic from an interface over a dedicated VLAN to an interface on a different switch



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

Open vSwitch also supports Quality of Service (QoS):

- traffic queueing and shaping
 - security in terms of traffic filtering and VLAN isolation
-
- Open vSwitch works with most hypervisors (including Xen and KVM) and container systems (Docker)



Open vSwitch/OpenFlow and Linux Bridge (cont'd)--OpenFlow

OpenFlow breaks a network connection into a separate data plane and control plan

- in traditional network equipment, both these activities were in the same device
- control plane:
 - determines where a packet is routed
 - stores the routing information in the flow table
- data plane
 - routes packets according to the flow table



Open vSwitch/OpenFlow and Linux Bridge (cont'd)—OpenFlow (cont'd)

An OpenFlow flow table contains:

- a set of headers
- actions that are taken when a packet with the specified headers arrived

Headers plus associated actions are called “flows” or “flow rules” or “flow entries”

- also includes some additional information, such as priority



Open vSwitch/OpenFlow and Linux Bridge (cont'd)—OpenFlow (cont'd)

Headers examples:

- port where the packet arrived
- Ethernet—Media Access Control (MAC)—source address
- Ethernet (MAC) destination address
- IP source address
- IP destination address
- VLAN identifier



Open vSwitch/OpenFlow and Linux Bridge (cont'd)—OpenFlow (cont'd)

Actions examples:

- send a packet to an outgoing port
- modify an IP address in a packet and send that packet to an outgoing port
 - this is an example of Network Address Translation
- send a packet to the controller
 - this is a “packet in” event
 - it would happen in situations where there is not a flow to handle this packet (no flow defined for the packet):
 - that packet will be sent to the controller
 - the controller will create a new flow to handle the packet
- drop packet



Open vSwitch/OpenFlow and Linux Bridge (cont'd)—OpenFlow (cont'd)

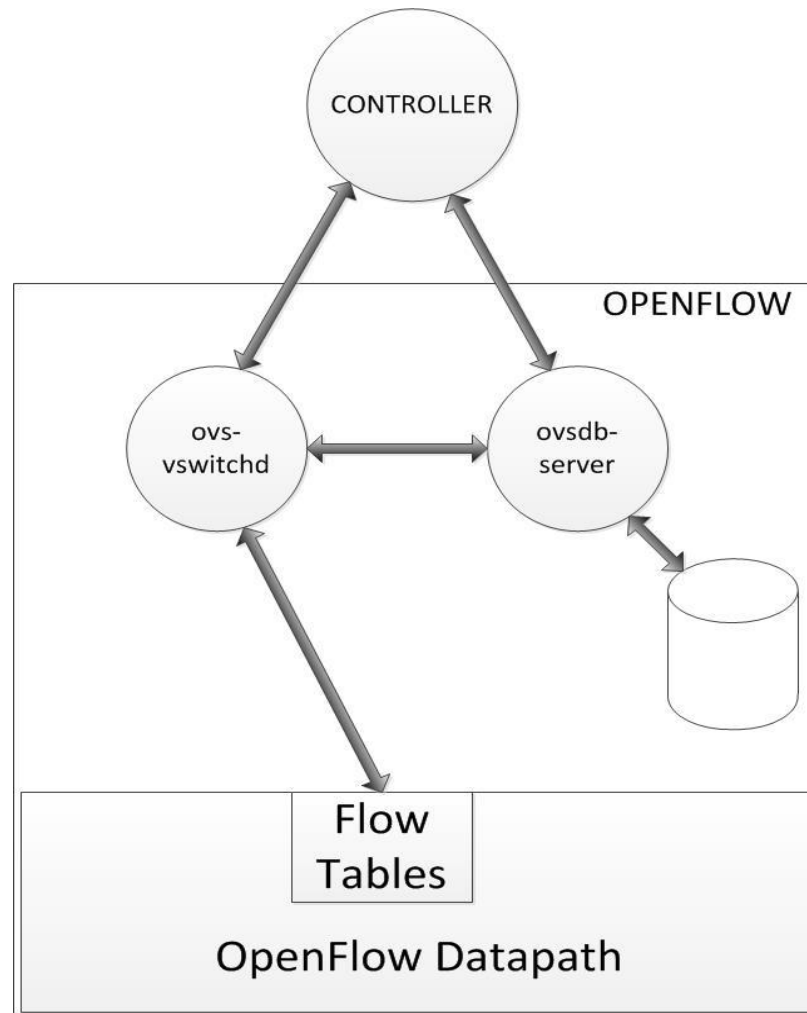
- Each flow rule has a priority associated with it.
 - In the case where a packet matches multiple flow rules, the flow rule with the highest priority is the rule that is applied
- Flow rules can have timeouts:
 - In the case of an idle timeout, if a flow has not received a packet in a certain amount of time, the flow is removed from the flow table
 - In the case of a hard timeout, after a certain amount of time the flow is removed from the table
 - whether or not it is still receiving packets



Open vSwitch/OpenFlow and Linux Bridge (cont'd)—OpenFlow (cont'd)

- per-table counters, per-flow counters, per-port counters, and per queue counters:
 - count the packets that have come through
- OpenFlow itself is focused on flow-based control of a switch
 - All configuration information related to creating or destroying OpenFlow switchings, adding or removing ports or queues, etc. is handled through the configuration database

Open vSwitch/OpenFlow and Linux Bridge (cont'd)—Open vSwitch





Open vSwitch/OpenFlow and Linux Bridge (cont'd)

- Referring to previous figure:
 - ovs-vswitchd is a daemon that manages the Open vSwitch switches on one computer
 - ovsdb-server controls a database
 - that stores switch configuration, including
 - definitions of bridges (a logical datapath is referred to as a bridge)
 - interfaces
 - tunnels
 - there is also a separate manager that configures the Open vSwitch instance, the manager is not shown in the previous figure



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

- Management and configuration on the Open vSwitch instance is performed using the OVSDB management interface
- The operations on this interface include:
 - Creation, modification, and deletion of OpenFlow datapaths (bridges)
 - Configures the set of controllers to which an OpenFlow datapath connects
 - Configures the set of managers to which the OVSDB server connects
 - Creation, modification, and deletion of ports on OpenFlow datapaths



Open vSwitch/OpenFlow and Linux Bridge (cont'd)

- The ovsdb-server uses the OVSDB protocol to talk to the ovs-vswitchd daemon and to the controller
- At startup, ovs-vswitchd contacts the ovsdb-server to retrieve configuration information
 - Based on this information, it sets up the datapaths (bridges) and sets up the switching in the flow tables
 - When the database changes, ovs-switchd will automatically update its datapaths and flow tables



Virtualization Security—Guest OS Isolation

- See NIST guidelines, Scarfone et al. (2011), for using and configuring virtualization technologies
- Issues with guest operating system isolation:
 - hypervisor is responsible for partitioning resources so that each guest operating system can see only its own resources.
 - Resources may be partitioned
 - physically, by assigning guest operating systems to separate physical devices
 - logically, by the hypervisor securely allocating resources
 - Logical isolation is also known as the guest operating system being in a “sandbox”



Virtualization Security (cont'd)—Guest OS Isolation (cont'd)

isolation of guest operating systems prevents side channel attacks that monitor usage patterns of hardware resources

- a side channel attack is an attack that is based on knowledge of the system and how it works



Virtualization Security (cont'd)—Guest OS Isolation (cont'd)

- an attacker can escape from the guest operating system in the virtual machine, in order to attack another virtual machine or the hypervisor itself
 - hypervisor is a single point of failure,
 - if the attacker can gain control over the hypervisor then the attacker can gain control over all virtual machines
- sharing resources between different guest operating systems can present additional attack vectors



Virtualization Security (cont'd)— Monitoring Guest OSes in VMs

Hypervisor should monitor all guest operating systems as they run, including memory usage, network traffic, processes, etc.

- this is called introspection
- monitoring network traffic can be more difficult since traffic between virtual machines often does not pass over a physical network



Virtualization Security (cont'd)--Images

Images and snapshots of guest operating systems can be security risks, since they contain passwords and personal data

- Snapshots contain the contents of RAM at the time the snapshot was taken
 - so additional sensitive information may be included in the snapshot
- When images and snapshots are moved around and stored
 - maintaining good security procedures to prevent unauthorized access is very important
- It is easy to create images, so unnecessary images may be created
 - each of these images is a possible security breach



Virtualization Security (cont'd)--Images

- Stored images will likely not receive ongoing operating system security patches,
 - so if those images are retrieved and run, security holes that are well known to potential attackers may very well be present
 - The longer an image is stored, the more vulnerabilities it will possess



Virtualization Security (cont'd)—images and migration

- If a virtual machine contains malware, and an image or snapshot is taken of that virtual machine, and then migrated elsewhere
 - then that malware will be spread
 - images should be periodically monitored using cryptographic hashes to make sure that no malware has been added while the image is stored
- Organizations should implement formal image management that takes these problems into account
- Migration of a virtual machine from one host to another represents a possible security threat
 - since if the virtual machine had malware it then spreads that to a new host



Virtualization Security (cont'd)— Hypervisor Security

- Hypervisor management communications should be protected
 - have a management network
 - that is separate from all user networks
 - can be accessed only by system administrators
 - limiting access to the hypervisor is critical to the whole system's security.
 - hypervisor should be carefully monitored for signs that it could be compromised
- disconnect any physical hardware when it is not being used, particularly Network Interface Cards (NICs).
 - similarly, for virtual machines disconnect any unused virtual hardware (virtual CDs, etc.).
- If a guest operating system has been compromised, assume that all other guest operating systems on the same hardware have been compromised



Virtualization Security (cont'd)— Hypervisor Security

- See Chandramouli (2014) for NIST draft standard on hypervisor security
- According to Mimoso (2014):

“The 22 security recommendations in the draft are mapped to each of the five primary hypervisor functions, and run the gamut from suggestions for reducing a hypervisor’s attack surface to determining which drivers are allowed to run emulation code, memory rules, monitoring recommendations, access control and permissions, patch and vulnerability management, and logging of security events among others.”



Virtualization Security (cont'd)— Hypervisor Security

From Chandramouli (2014), the five primary hypervisor functions are:

- HY-BF1: Execution Isolation for Virtual Machines (VMs)
- HY-BF2: Devices Emulation & Access Control
- HY-BF3: Execution of Privileged Operations for Guest VMs
- HY-BF4: Management of VMs
- HY-BF5: Administration of Hypervisor Platform and Hypervisor Software



Virtualization Security (cont'd)— Hypervisor Security

**HY-BF1: Execution Isolation for
Virtual Machines (VMs):**

- Scheduling VMs
- Managing CPU and memory related to processes running in VMs
- Processor context switching during running of applications inside VMs



Virtualization Security (cont'd)— Hypervisor Security

HY-BF2: Devices Emulation & Access Control

- Emulating network devices as expected by native drivers in the VMs
- Emulating storage devices as expected by the native drivers in the VMs
- Controlling how different VMs access physical devices



Virtualization Security (cont'd)— Hypervisor Security

HY-BF3: Execution of Privileged Operations for Guest VMs

- Execute privileged operations in hypervisor instead of on host hardware



Virtualization Security (cont'd)— Hypervisor Security

HY-BF4: Management of VMs

- Configure VM images
- Configure VM states (start, stop, pause, etc.)



Virtualization Security (cont'd)— Hypervisor Security

HY-BF5: Administration of Hypervisor Platform
and Hypervisor Software

- Configure parameters for user interaction with hypervisor host



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-1: A Type 1 hypervisor has fewer security vulnerabilities than a Type 2 hypervisor

- since there is no host operating system that can be attacked



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-2: A hypervisor with hardware assisted virtualization (instruction set and memory management) has fewer security vulnerabilities than a hypervisor with only software assisted virtualization because:

- Buffer overflow and similar attacks are prevented through good control over memory management
- Hardware-based memory protection and privilege isolation helps better control shared devices
- Guest operating systems can be easily updated, since they need not be modified to run on paravirtualized platforms
- Support of virtualization in hardware results in smaller hypervisor code, which is easier to examine for vulnerabilities



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-3 (optional): The hypervisor that is launched should be part of a platform and an overall infrastructure that contains a Measured Launch Environment (MLE) and a Trusted Platform Module (TPM) in order to ensure boot integrity:

- A Measured Launch Environment (MLE) is a hardware boot sequence
 - compares the hashes of the components being booted (firmware, BIOS, and hypervisor modules) to make sure the components are correct and unmodified
- includes a Trusted Platform Module (TPM)
 - stores the results of the measurements
 - allows reporting of discrepancies



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-4: The hypervisor management console should be kept small with few exposed interfaces

- to provide fewer vulnerabilities
- to be easier to examine for vulnerabilities



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-5: The hypervisor boot configuration should be settable so that non-certified drivers are not allowed

- when possible hardware emulation (QEMU) should be run in an unprivileged VM
 - so application VM is not impacted by a faulty device driver



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-6: A hypervisor normally meets memory requirements through a combination of physical RAM and swap files

- A VM normally would not require all its configured memory all the time.
 - Therefore, it's reasonable to have the total memory configured to belong to all VMs on a host to exceed the total physical RAM available on the host.
 - However, if the amount of VM-configured memory is too large compared to the physical RAM
 - then performance may degrade
 - the host may not be available for certain VM workloads.
 - the ratio of combined configured VM memory to RAM memory should typically be around 1.5 to 1



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

- Security Recommendation HY-SR-7: there should be a way for the hypervisor to configure a guaranteed quantity of physical RAM and also a limit on the quantity of physical RAM for every VM
 - hypervisor should also be able to prioritize RAM resources among multiple VMs



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-8: Each VM should have a way to guarantee that it will eventually run

- a VM may need two cores in order to run and have to wait until those cores are available at the same time
- therefore a hypervisor should have a way to reserve a minimum CPU allotment
- a VM should have an upper bound on its CPU allotment in order to prevent a (possibly compromised) VM from taking up all CPU resources and not allowing other VMs to run
 - thus the number of virtual CPUs allocated to any VM must be less than the total number of cores in the host machine



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-9: every VM must have a lower bound and a lower bound on CPU clock cycles, and each VM must have a priority

- this will enable scheduling VMs when they compete for CPU resources



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-10: VM images can be a big security threat

- the VM image library should be located elsewhere outside the host machine running the hypervisor
- each image should be digitally signed
 - to ensure it is authentic
 - that it has not been compromised



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-11: Running VMs should be monitored (introspection by the hypervisor) to search for:

- malware inside VMs
- malicious network traffic to or from VMs.

This would employ anti-virus and intrusion detection and prevention.



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-12: Security monitoring and security enforcement should use introspection (monitoring of the VMs by the hypervisor)

- could be done by running a security tool (Security Virtual Appliance) in a trusted VM



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-13: Access control security permissions should be able to be granted to a single VM, to a physical group of VMs, or to a logical group of VMs

- should be possible to selectively deny access of group members to individual VMs within a group
 - if those VMs have sensitive data (perhaps on a temporary basis).



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-14: The number of user or system administrator accounts accessing the hypervisor should be kept very low

- perhaps two or three



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-15: All accounts that access a hypervisor should be heavily authenticated through the company's user directory

- addition and deletion of accounts immediately updated
- all password, etc. policies should be enforced



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-16: It should be possible:

- to prevent remote access to the hypervisor administrative console
- to deny root account access
- to restrict remote access to a small list of administrative accounts.



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-17: It should be possible:

- to define a complete set of known good configuration settings
- to automatically apply these known good settings to a new hypervisor installation or to an existing hypervisor installation
- to check an existing hypervisor installation to make sure it matches known good settings



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-18: All hypervisor patches should be applied immediately or very soon.



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-19: The hypervisor's firewall should only allow ports and network traffic that are actually needed for services that are enabled in the hypervisor.



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-20: Hypervisor logs should be kept in a standard format so they can be easily analyzed.

- Logs should be located on an external machine in case the machine the hypervisor is running on crashes or its security is compromised



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-21: The hypervisor management interface should be accessed by an isolated virtual network.

- Incoming traffic into the management interface should be controlled by a firewall
 - for example, only traffic into the management interface from certain subnets should be allowed



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Security Recommendation HY-SR-22: To prevent bottlenecks in network communication, multiple communication channels from a given VM to the outside network should be created.

- Usually this is done by providing multiple physical network interface cards (NICs) that traffic from a given VM can flow through



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

- Chandramouli (2014) provided a table in an appendix that mapped the security recommendations to the baseline functionalities
- One thing to learn easily from the way Chandramouli's table is drawn is that HY-SR-1 and HY-SR-3 don't map to any of the hypervisor baseline functionalities
 - HY-SR-1 is just saying that if you have less code there's less to attack
 - HY-SR-3 says it's good to make sure the right code always is loaded in when you boot



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

Table 7.1 turns Chandramouli's table backwards

- maps baseline functionalities to the security recommendations
 - can see that none of the security recommendations map to HY-BF3
 - this is just because HY-BF3 says your hypervisor should replace operating system calls to the hardware with hypercalls to the hypervisor
 - this is just typical hypervisor operation
- However, as Chandramouli says, when you have problems here it's because the hypervisor has faulty code
 - Since this is buried inside the hypervisor, cannot make security recommendations related to configuration, deployment, or procedure



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

**Table 7.1 Baseline Functionalities Supported by Which Security
Recommendations**

HY-BF1	HY-BF2	HY-BF3	HY-BF4	HY-BF5
HY-SR-2	HY-SR-5		HY-SR-4	HY-SR-4
			HY-SR-6	HY-SR-14
			HY-SR-7	HY-SR-15
			HY-SR-8	HY-SR-16
			HY-SR-9	HY-SR-17
			HY-SR-10	HY-SR-18
			HY-SR-11	HY-SR-19
			HY-SR-12	HY-SR-20
			HY-SR-13	HY-SR-21
				HY-SR-22



Virtualization Security (cont'd)— Hypervisor Security—Security Recommendations

From Table 7.1, see that most of the security recommendations are related to

- management of VMs (HY-BF4)
- administration and configuration of the hypervisor itself (HY-BF5)



Virtualization Security (cont'd)—Cloud Security

- Considerable overlap between “virtualization” security and “cloud” security, in that clouds employ virtualization
- some aspects of cloud security go beyond virtualization For example:
 - control of user accounts can be considered a more general security problem than a virtualization problem
 - denial of service attack is a cloud problem but is not necessarily directly related to virtualization



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

Cloud Security Alliance (CSA):

- a non-profit organization
 - Numerous corporations participate, including (as of 2016): Microsoft,
 - Rackspace,
 - VMware,
 - HP Enterprise,
 - Cisco,
 - Citrix,
 - RedHat,
 - Symantec
- formed in 2008



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

Cloud Security Alliance (CSA):

- defines best practices for cloud security
 - provides a third party assessment of a cloud service provider's security through:
 - CSA Star Certification
 - assessment process



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

- CSA Star registry indexes the security features supported by cloud providers
 - Numerous cloud providers are listed on this registry, including among several others (as of 2016):
 - Microsoft Azure
 - RedHat OpenShift
 - VMware
 - Citrix Sharefile



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

- CSA sponsors cloud symposiums:
 - Cloud Security Summit
 - CSA Federal Summit
- CSA provides cloud computing certifications:
 - Certificate of Cloud Security Knowledge
 - Certified Cloud Security Professional
- CSA Top Threats Working Group performs ongoing analyses of the top threats to cloud security



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

From CSA Top Threats Working Group (2016) the top threats from 2016 are:

1. Data Breaches
2. Insufficient Identity, Credential, and Access Management
3. Identity access management systems cannot scale to the size needed.
4. Insecure Interfaces and APIs
5. System vulnerabilities
6. Malicious Insiders
7. Advanced Persistent Threats
8. Data Loss
9. Insufficient Due Diligence
10. Abuse and Nefarious Use of Cloud Services
11. Denial of Service
12. Shared Technology Issues



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

1. Data Breaches

- a data breach is when an unauthorized individual can see and use data

2. Insufficient Identity, Credential, and Access Management

- Identity access management systems cannot scale to the size needed.
- Authentication should be performed using multiple kinds of identifying information
- Weak passwords are used
- Cryptographic keys, passwords, and certificates are not rotated automatically
- Credentials and cryptographic keys are embedded in source code
- Public Key Infrastructure systems are needed to ensure key management is appropriately performed



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

3. Insecure Interfaces and APIs
 - All APIs must be well designed because overall cloud security depends on the basic interfaces. These APIs are heavy attack targets.
4. System vulnerabilities
 - Bugs in programs are a big problem.
 - Since systems from different organizations share resources, this leads to a new attack vector
5. Account Hijacking
 - Phishing and fraud are ongoing problems
 - Credential sharing among users should be prohibited
 - Two or more factor authentication should be used where possible
 - All accounts should be monitored and traceable to a human



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

6. Malicious Insiders

- A malicious system administrator is a big problem
- System administrator duties should be separated, access controlled by role
- Users should control encryption and keys themselves
- Administrator activities should be logged, monitored, and audited



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

7. Advanced Persistent Threats
 - Attackers infiltrate systems to smuggle data and intellectual property
8. Data Loss
 - Permanent loss of a user's data
 - Cloud provider must take care to follow best practices for data backup and recovery, including off site storage
 - Users must take care not to lose their encryption keys



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

9. Insufficient Due Diligence
 - When companies move to the cloud they must carefully analyze commercial, technical, and legal issues (data privacy, etc.) involved
10. Abuse and Nefarious Use of Cloud Services
 - cloud service deployments without sufficient attention to security, free cloud service trials and fraudulent accounts can lead to misuse of cloud computing resources, including:
 - denial of service attacks
 - phishing and email scams
 - bitcoin mining
 - brute force computing attacks
 - hosting pirated content



Virtualization Security (cont'd)—Cloud Security—Cloud Security Alliance

11. Denial of Service

- Denial of service attacks can be performed against the cloud provider itself, or specifically against a user of the cloud provider
- According to Rashid (2016), in regard to the CSA 'dirty dozen' security threats: "DoS attacks consume large amounts of processing power, a bill the customer may ultimately have to pay."

12. Shared Technology Issues

- Underlying hardware and software may not have been designed to provide the isolation required when shared by multiple users
- An in depth strategy to enforce user isolation at all levels is necessary



Virtualization Security (cont'd)—Cloud Security—Physical Data Center Security

A few ways to steal data without hacking:

- a team armed with machine guns breaks into your data center, loads up a truck with several storage devices, and drives off
- one person sneaks a gun into the data center and holds it to a system administrator's head to make him upload a lot of data
- a person breaks into your data center at night, and collects a lot of data on a few USB drives
- a person goes to the garbage dump where your company has been throwing away its old hard drives and collects a few



Virtualization Security (cont'd)—Cloud Security—Physical Data Center Security

Possible causes of data loss or data unavailability:

- earthquakes
- hurricanes
- tornado eats your data center
- tornado eats power lines leading to your data center
- someone physically cuts the power lines or phone lines leading into your data center
- backhoe digs up the fiber line leading into your data center
- arson
- bombs sent in packages



Virtualization Security (cont'd)—Cloud Security—Physical Data Center Security

Lawton (2014) and Scalet (2015) discuss ways to ensure physical security:

- don't build data centers in earthquake or hurricane zones
- don't advertise that your data center is a data center—no company logos
- Make your data center hard to find
- All computer rooms in interior of the building
- All computer rooms should be in the interior of the building, not at outside walls.
- Crash proof barriers and buffer zones around building
- Fences
- Armed guards
- Surveillance cameras
- Restrict important areas such as utility areas
- Redundant utilities
- No drop down ceilings
- Employees may not have weapons in the buildings
- Employees may not bring in usb drives
- Background checks for all employees