Virtualization Technology and Security 21

21.1 Introduction

Virtualization is a process through which one can create something that is there in effect and performance but in reality not there—that is, virtual. It is a physical abstraction of the company's computing resources like storage, network servers, memory, and others. VMware.com, a software developer and a global leader in the virtualization market, defines virtualization as a process in which software creates virtual machines (VMs) including a virtual machine monitor called "hypervisor" that allocates hardware resources dynamically and transparently so that multiple operating systems, called "guest operating systems," can run concurrently on a single physical computer without even knowing it [1]. For example, using software virtualization, one can, using the existing underlying hardware and software resources like operating systems, create and run several independent virtual operating systems on top of one physical operating system using the existing hardware resources to execute independent system tasks. Hardware virtualization also takes the same concept where several servers or client machines can be created based on one underlying hardware. The virtualization concept has been with us for sometime.

The potential power of virtualization in substantially increasing the performance of computing systems such as hardware and software through division of the underlying physical computing resources into many equally powerful virtual machines has increased the popularity of the technology in the last 20 years, and this love continues today. According to the IDC, an IT research firm, 2012 ranking of chief information officer (CIO) priorities, virtualization and the server consolidation that it delivers were the top priority for chief information officers. Forty percent of CIOs picked virtualization and server consolidation, more than any other area of IT [2]. The rush to virtualization is driven by its resulting server consolidation creating savings to be invested in new IT initiatives such as cloud computing, mobility, data analytics, and use of social media for business purposes. This rapid growth is a reflection of the changing benefits of virtualization from being used only

J.M. Kizza, *Guide to Computer Network Security*, Computer Communications and Networks, DOI 10.1007/978-3-319-55606-2_21

as a tactical tool to drive consolidation and higher system utilization to leveraging the mobility of virtual machines to improve management and operations of IT environments. The virtualization concept now includes a host of new use cases that range from high availability and disaster recovery to hosted clients and true utility computing.

21.2 History of Virtualization

The history of virtualization is as amazing as the concept itself. Since computers of the 1960s could do only one task at a time and depended on human operators, increasing system performance was bottlenecked at two points: at the submission stage and at the computation stage. One way to improve the submission stage was to use a batch, where jobs were submitted into a queue and the system picked them from there, thus reducing human intervention and errors. Batching improved some system performance but did not go far enough. This problem, together with creating backward compatibility for customers of older computing systems the ability to bring old functionalities of the old to the new, and thus keep customer royalty, led IBM to begin work on the S/360 mainframe system. The S/360 mainframe was capable of running legacy functionalities of nearly all IBM's older systems, although it was still a batch machine. In the following years, there was a growing need, especially in the research community like at Bell Labs and Massachusetts Institute of Technology (MIT), for a machine that was capable of running tasks of more than one simultaneous user. In response to this growing need for speedup, IBM responded with the CP-40 mainframe which later evolved into the CP-67 system, thought to be the first commercial mainframe to support virtualization. The CP-67 had a unique operating system combination consisting of CMS (Console Monitor System) piggybacked on a control program called rightly CP. The CP/CMS was a small single-user interactive operating system and CP, upon which CMS run, actually run on the mainframe to create the virtual machines which individually run their own copies of CMS. To each virtual machine running CMS, CP allocated parts of the underlying physical machine which formed the virtual machine [4].

When microprocessors made their debut into computing in the 1980s and beyond, creating an era of personal computers which led into desktops and small servers leading to computer networks of varying sizes which seemed to lower the costs of computing and improved system performance, virtualization technology took a back seat and was almost forgotten. The situation did not change until the mid-1990s when the cost of computing skyrocketed again in spite of large-scale distribution of computing by client-server models of computation. There was a growing need to revisit virtualization and rain in the rising costs of information technology.

In 1999, VMware introduced a new kind of virtualization technology which, instead of running on the mainframe, run on the x86 system. VMware virtualization technology was able to isolate the shared hardware infrastructure of the x86

architecture. Today, VMware is the global leader in x86 virtualization which offers desktop, server, and data center [3].

21.3 Virtualization Terminologies

For one to understand the virtualization process, one has to first understand the terminologies used and make up the process. There are several terminologies used specifically in the virtualization process, and they include *host CPU and guest CPU*, *host operating system* and *guest operating system*, *hypervisor*, and *emulation*.

21.3.1 Host CPU/Guest CPU

When a virtualization software is creating a new VM upon which the virtual OS runs, it creates a virtual CPU, known as a *guest CPU*, best on the time slices allowed on the underlying physical, now called a *host CPU* on the host machine. There is corresponding coordination and linkages between the host and guest CPUs. The guest CPU in the VM created is not aware of the host CPU or the host machine supporting it. It is also not aware of its sibling guest CPUs in the sibling VMs.

21.3.2 Host OS/Guest OS

During the virtualization process, the virtualization software creates complete VMs based on the underlying physical machine. These VMs have all the functionalities of the underlying physical/host machine. However, during the process, the virtualization software, for each VM created, may or may not create a new/guest operating system or make as a copy of the physical/host operating system. This new operating system, on each newly created VM, is a *guest operating system (guest OS)*, and the physical operating system running on the physical machine is the *host operating system (host OS*). The guest operating system has no knowledge of the existence of either the host operating system or the sibling guest operating systems. All VMs are consistent with each other and the host VM in that each has the same resources, save the guest operating system, like the host machine. The only difference in consistency occurs in disk I/O operations. To solve this problem, there is a required mapping of the guest disk I/O operations with the physical disk I/O operations. For example, users of Windows VMs must interact with it over the network via Windows Terminal Services (RDP), and those using Unix/Linux VMs must interact with them via the network using SSH.

21.3.3 Hypervisor

A hypervisor, as a virtual machine manager, is a software program that allows multiple operating systems to share a single physical hardware host. In creating the virtual machine for each operating system, the hypervisor uses "slices" of the physical host machine's physical components like memory, processor, and other resources to anchor each guest operating system running the virtual machine created. The host physical machine's "slices" allocated to each virtual machine are managed by the hypervisor in amounts and time durations as needed by each operating system.

21.3.4 Emulation

An emulation is a process of making an exact copy of all the functionalities of an entity like a hardware resource of a computing system like a CPU and operating system, I/O devices and drivers, and others. Emulation software is an application software running on a host to emulate the host. Emulators can create guest OS. These emulated OS have no knowledge of the existence of either the host machine and its OS or its siblings. The problem with emulators as opposed to hypervisors is that emulators are slow.

21.4 Types of Computing System Virtualization

There are many types of virtualization including platform, network, storage, and application.

21.4.1 Platform Virtualization

Platform virtualization is the use of server hardware by the virtualization software to host multiple VMs as guest VMs. Each VM is a virtual environment with its operating system (the guest operating system), which may or may not be the same as the physical server's operating system (the host operating system), and emulates the whole physical infrastructure of a computing system including memory, and each VM is independent of other VMs sharing the physical server. Platform virtualization itself is subdivided into two types: workstation and server.

21.4.1.1 Workstation Virtualization

This is also referred to as *desktop virtualization*. It is the abstraction of the traditional workstation with its operating system, by moving it to a remote server system, accessed via a smart or dumb terminal. Desktop virtualization becomes popular to the business world because of its savings resulting from a reduction in desktop sprawl. Desktop virtualization has been around for decades starting in the

days of the timeshare systems. During those days, the technology was known by different names including terminal service computing that included dumb terminals, terminal emulators, and thin-client computing. It was also known as technology which allowed full exploitation of the true power and flexibility of a desktop or laptop computer by making it capable of running multiple operating systems simultaneously on a single processor. With the ability to emulate multiple fully operational "machines" on one computer, one can get the following benefits from that one computer [5]:

- Ability to run a variety of applications specific to individual operating systems not currently running on the physical machine.
- Ability to host legacy applications and overcome platform migration issues.
- Demonstrate multi-tier configurations on a single processor like running SQL Server database server running in one virtual machine, a Web server running on another virtual machine, and several other server-based applications all running on a single host desktop.
- Configure and test new software or patches in an isolated environment, thus reducing deployment risks and costs.
- Automate tasks for software development and testing.

21.4.1.2 Server Virtualization

Server virtualization is the process of having a physical server run a server-based virtualization software called a hypervisor to divide the physical server into multiple isolated virtual environments. Each virtual environment in a virtual machine, is homed on a virtual server, and has all the functionalities of the physical server it is homed on and runs a virtual operating system called a guest operating system. The virtual machines created are known by different names including virtual private servers, guest machines, instances, containers, or emulations.

According to [3], there are three popular approaches to server virtualization: the virtual machine model, the paravirtual machine model, and virtualization at the operating system (OS) layer.

The *virtual machine model* is based on a *host*/*guest* paradigm. Each guest runs on a virtual imitation of the physical hardware layer. This approach allows each guest operating system on each virtual machine to run without *modifications* to the resources of the underlying physical machine. It also allows the different virtual machines to run different guest operating systems. The guest operating systems has no knowledge of the host's operating system because they assume that they are running on the physical hardware. Each guest operating system access to the physical recourses of the host machine is managed by the hypervisor.

The paravirtual machine (PVM) model is also based on the *host*/*guest* paradigm. The two models are very much alike. The only difference though between the virtual machine and the paravirtual machine models lies in the fact that this time, the hypervisor can modify the guest operating system's code through a process called *porting*. With porting, the hypervisor can prioritize and utilize privileged system calls between the guest operating system and the physical processor.

Unlike the virtual machine and paravirtual machine models, the *OS-level* virtualization model is not based on the *host*/*guest* paradigm. In the OS-level model, the host runs a single OS kernel as its core and exports operating system functionality to each of the guests. Guests must use the *same* operating system as the host, although different distributions of the same system are allowed. This distributed architecture eliminates system calls between layers, which reduces CPU usage overhead. It also requires that each partition remain strictly isolated from its neighbors so that a failure or security breach in one partition isn't able to affect any of the other partitions. In this model, common binaries and libraries on the same physical machine can be shared, allowing an OS-level virtual server to host thousands of guests at the same time. Virtuozzo and Solaris Zones both use OS-level virtualization. Although we stated earlier that there are no modifications by the hypervisor of the characteristics of the underlying physical resources given to each virtual machine, there is in fact a limited modification by the hypervisor. The hypervisor actually modifies the guest operating system's code. This modification is called porting. Porting supports the hypervisor so it can utilize privileged system calls sparingly [10].

Whether workstation or server virtualization, platform virtualization is the more popular form of virtualization and it is growing fast. The table below lists a good number of platform virtual machine (VM) software packages, their host CPU, guest CPU, host operating systems, guest operating system, and type of license it carries [6]. Note licensing is classified as follows [9]:

- LGPL—GNU Lesser General Public License (formerly the GNU Library General Public License) is a free software license published by the Free Software Foundation (FSF). It was published in 1991 and adopted the version number 2 for parity with GPL version 2, renamed in 1997 as the GNU Lesser General Public License. It places *copyleft* (a general method for making a program (or other work) free, and requiring all modified and extended versions of the program to be free as well) restrictions on the program governed under it but does not apply these restrictions to other software that merely link with the program. It is primarily used for software libraries.
- GPL—The GNU General Public License (or simply GPL) is the most widely used free software license. Currently in version 3, the GPL is the first copyleft license for general use, which means that derived works can only be distributed under the same license terms.
- CDDL—Common Development and Distribution License (CDDL) is a free software license, produced by Sun Microsystems, based on the Mozilla Public License (MPL), version 1.1. Files licensed under the CDDL can be combined with files licensed under other licenses, whether open source or proprietary.
- BSD—BSD licenses are a family of permissive free software licenses. The original license was used for the Berkeley Software Distribution (BSD), a Unix-like operating system after which it is named.

Table 21.1 Platform virtualization software packages

Name	Responsible party	Host CPU	Guest CPU	Host OS(s)	Guest OS(s)	License
FreeBSD Jail	FreeBSD	Any running FreeBSD	Any running FreeBSD	FreeBSD	FreeBSD, Linux ABI	BSD
Hyper-V Server 2008 R2	Microsoft	x86-64 + hardware-assisted virtualization (Intel VT-x or AMD-V)	x86-64, x86 (up to eight physical CPUs)	Windows 2008 w/Hyper-V Role, Windows Hyper-V Server	Supported drivers for Windows 2000, Windows 2003, Windows 2008, Windows XP, Windows Vista, Linux (SUSE 10 released, more announced)	Proprietary
iCore Virtual Accounts	iCore Software	x86	x86	Windows XP	Windows XP	Proprietary
Integrity Virtual Machines	Hewlett-Packard	IA-64	IA-64	HP-UX	HP-UX, Windows, Linux (OpenVMS announced)	Proprietary
LynxSecure	LynuxWorks	x86, Intel VT-x, Intel VT-d	x86	No host OS	LynxOS, Linux, Windows	Proprietary
PikeOS	SYSGO AG	PowerPC, x86, ARM, MIPS, SPARC, SuperH	Same as host	PikeOS	PikeOS native, Linux, POSIX, AUTOSAR, ANDROID, RTEMS, OSEK, ARINC 653 APEX, ITRON	Proprietary
QuickTransit	Transitive Corp.	x86, x86-64, IA-64, POWER	MIPS, PowerPC, SPARC, x86	Linux, Mac OS X, Solaris	Linux, Mac OS X, Irix, Solaris	Proprietary

(continued)

Table 21.1 (continued)

Name	Responsible party	Host CPU	Guest CPU	Host OS(s)	Guest OS(s)	License
RTS Hypervisor	Real-Time Systems	x86	x86	No host OS	Windows 7, Windows XP, Windows Embedded, Windows CE, Linux, Android, VxWorks, OS-9, RTOS-32, QNX, RTEMS, T-Kernel, proprietary	Proprietary
Safe Virtual Machine, SVM	Altreonic, www.altreonic.com	Any	Any	OpenComRTOS or any other (RT)OS	N.A.	Binary, Open Technology License
Simics	Virtutech	x86, x86-64, SPARC v9	Alpha, ARM, IA-64, MIPS 32/64, MSP430, POWER, PowerPC 32/64, SPARC v8/v9, x86, x86-64, TI TMS320C64xx	Windows, Linux, Solaris	Depends on target machine, VxWorks, OSE, QNX, Linux, Solaris, Windows, FreeBSD, RTEMS, TinyOS, many others	Proprietary
Containers, or Zones	Sun Microsystems	x86, x86-64, SPARC (portable: not tied to hardware)	Same as host	Solaris 10, Solaris 11 Express, OpenSolaris 2009.06	Solaris (8, 9, 10, 11), Linux (BrandZ)	CDDL
Sun xVM Server	Sun Microsystems	x86-64, SPARC	Same as host	No host OS	Windows XP, 2003 Server (x86-64 only), Linux, Solaris	GPL version 3
Virtual PC 2007	Connectix	x86, x86-64	x86	Windows Vista (Business, Enterprise, Ultimate), XP Pro, XP Tablet PC Edition	DOS, Windows, OS/2, Linux (SUSE, Xubuntu), OpenSolaris (Belenix)	Proprietary

Virtual PC 7 for Mac	Connectix	PowerPC	x86	Mac OS X	Windows, OS/2, Linux	Proprietary
Virtual Server 2005 R2	Connectix	x86, x86-64	x86	Windows 2003, XP	Windows NT, 2000, 2003, Linux (Red Hat, SUSE)	Proprietary
VirtualBox	Innotek, acquired by Oracle Corporation	x86, x86-64	x86, (x86-64 only on VirtualBox 2 and later with hardware virtualization)	Windows, Linux, Mac OS X x86, Solaris, FreeBSD, eComStation	DOS, Linux, Mac OS X Server,[6] FreeBSD, Haiku, OS/2, Solaris, Syllable, Windows	GPL version 2; full version with extra enterprise features is proprietary
VMware ESX Server	VMware	x86, x86-64	x86, x86-64	No host OS	Windows, Linux, Solaris, FreeBSD, OSx86 (as FreeBSD), virtual appliances, Netware, OS/2, SCO, BeOS, Darwin, others: runs arbitrary OS[7]	Proprietary
VMware ESXi	VMware	x86, x86-64	x86, x86-64	No host OS	Same as VMware ESX Server	Proprietary
VMware Fusion	VMware	x86, x86-64	x86, x86-64	Mac OS X x86	Same as VMware ESX Server	Proprietary
VMware Player 3.1	VMware	x86, x86-64	x86, x86-64	Windows, Linux	Same as VMware ESX Server	Proprietary, free of charge for personal noncommercial use[8], [2]
VMware Server	VMware	x86, x86-64	x86, x86-64	Windows, Linux	Same as VMware ESX Server	Proprietary

(continued)

Table 21.1 (continued)

Name	Responsible party	Host CPU	Guest CPU	Host OS(s)	Guest OS(s)	License
VMware Workstation 7.1	VMware	x86, x86-64	x86, x86-64	Windows, Linux	Same as VMware ESX Server	Proprietary
Wind River hypervisor	Wind River	x86, PowerPC	Same as host	No host OS	Linux, VxWorks, unmodified guests (including MS Windows and RTOSs such as OSE, QNX, and others), bare metal virtual board	Proprietary
Windows Virtual PC	Connectix	x86, x86-64 with Intel VT-x or AMD-V	x86	Windows 7	Windows XP, Windows Vista, Windows 7, Windows Server 2003, Windows Server 2008	Proprietary
Xen	XenSource	x86, x86-64, IA-64	Same as host	NetBSD, Linux, Solaris	FreeBSD, NetBSD, Linux, Solaris, Windows XP and 2003 Server (needs vers. 3.0 and an Intel VT-x (Vanderpool) or AMD-V (Pacifica)-capable CPU), Plan 9	GPL
z LPARs	IBM	z/Architecture	z/Architecture	Integrated in firmware of System z mainframes	Linux on zSeries, z/OS, z/VSE, z/TPF, z/VM, MUSIC/SP, and predecessors	Integrated in firmware of System z mainframes

PowerVM	IBM	POWER4, POWER5, POWER6, PowerPC 970	POWER4/5/6, PowerPC 970, X86 (PowerVM Lx86)	No host OS	Linux PowerPC, x86; AIX, i5/OS, IBM i	Proprietary
z/VM	IBM	z/Architecture	z/Architecture, z/VM does not run on predecessor mainframes	No host OS, itself (single or multiple levels/versions deep, e.g., VM/ESA running in z/VM 4.4 in z/VM 5.2 in z/VM 5.1.)	Linux on zSeries, z/OS, z/VSE, z/TPF, z/VM, VM/CMS, MUSIC/SP, OpenSolaris for System z, predecessors	Proprietary

Source and for a full list: *Wikipedia*. http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines

- OPL—Open Publication License is a license open publications created by the Open Content Project, which now recommends [1] using one of the Creative Commons licenses.
- Proprietary—Proprietary software is computer software licensed under exclusive legal right of the copyright holder. The licensee is given the right to use the software under certain conditions, while restricted from other uses, such as modification, further distribution, or reverse engineering.
- Open Source—It is a philosophy which allows free redistribution, reuse, reengineering, and access to an end product's design and implementation details (Table 21.1).

What follows are some of the most popular platform virtualization software packages. A more extensive list can be found at http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines.

21.4.2 Network Virtualization

Like storage virtualization, network virtualization pools the resources, like files, folders, storage, and I/O devices, of separate and different networks into one network. This in fact is a network abstraction which isolates network traffic from network physical elements like switches, network ports, routers, and others within those networks, replacing each physical element with virtual representations and being able to duplicate them. This is done by splitting up the available bandwidth into independent channels, within the system. This makes it easy for the network administrator to share and assign network resources out among local network users thus allowing each network user to access all of the pooled network resources from their computer. This perhaps is the greatest benefit for network virtualization. In addition, network virtualization improves the ability of a user to move data into and out of storage resources to meet their current demands.

There are two types of *network virtualization*, the external and internal. External network involves the creation of multiple networks or parts of networks into a single virtual entity using all physical network elements like cabling, network adapters, switches, routers, and so on. Internal virtualization on the other hand is the process of creating one or more logical networks by defining logical switches and network adapters within a virtualized server itself. Note that an internal virtual network can connect two or more virtual machines on a single server and allow data exchanges between the virtual machines via the virtual server without involving the underlying physical network infrastructure, thus creating a virtual system-wide sharing and other network functionality. This is creates a fast and more efficient communication between virtual machines in the network on the virtual server thus minimizing traffic on the physical network. Also it gives a network administrator flexibility to combine virtual network elements in any way to create a network of any size and scope for the organization or create multiple networks that will share

the same physical network infrastructure. Although internal virtualization is fast and eases the job of a network administrator, it creates other problems including workload balancing and migration within the network.

For both external and internal network virtualization to work, it requires network virtualization software on each virtualized server as well as within switches and other network elements that support network virtualization. This integration between hardware and software elements must work well to support network virtualization. At the writing of this chapter, some of the best network virtualization software include Citrix; Vyatta; ConteXtream, Inc.; and others.

Finally, the concept network virtualization is not a new one. For years, we have been working with virtual private networks (VPNs), first by telephone companies before digital networks. With advent of digital network, security professionals have been using the concept VPN for years now. In addition to VPNs, there has also been the concept of virtual local area networks (VLANs), and virtual LAN (VLAN), a group of logically networked devices on one or more LANs configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments, also represents a common variation of network virtualization.

21.4.3 Storage Virtualization

The process of pooling together of resources of many different network storage devices such as hard drives to create what looks like one big storage managed from a single console is referred to as *storage* virtualization. There are several advantages why storage virtualization is good for business. First it hides the complexity of having multiple storage devices in many and different networks into one and simplifying the interface and console operations. Second it reduces the costs of storage reducing the overall storage infrastructure problems. And finally it works well for backups. There are some drawbacks that tend to prevent some from utilizing the technology like being complex to implement therefore requiring external help sometimes.

21.4.4 Application Virtualization

In application virtualization, the software package allows the bytecode of an application package to be portably run on many different computer architectures and operating systems. The virtualization software package achieves this through the use of running an interpreter or just-in-time compilation of the application before it runs on the computer architecture of choice. An example of this is the Java machine virtualization.

21.5 The Benefits of Virtualization

As we discussed in Sect. 21.2, virtualization technology has had a long history. This history has been driven by developers longing for a technology that will come with handsome benefits that will yield a high return on investment. Virtualization technology fits that technology. It is a technology that has brought to the computing community the following benefits [6].

21.5.1 Reduction of Server Sprawl

For a growing business with intensive computing requirement, the demand for servers cannot be underestimated. With business growth, there is a corresponding growth in the number of servers in the business. This can be costly not only in terms of physical storage but also in management, monitoring, and maintenance. One of the best solutions to this problem is server virtualization. Server virtualizations allow the company to scale up the business server infrastructure without purchasing additional pieces of hardware and requiring more space to store them and less technical staff to maintain and manage them.

21.5.2 Conservation of Energy

With less physical servers at the business data center, there is likely to be far less power consumption, thus reducing the overall company IT costs.

21.5.3 Reduced IT Management Costs

Again with a reduced physical server count on the premises and the ability to manage all the virtual infrastructure via one or two consoles, there is corresponding reduction in the IT management requirements and therefore reduced IT management costs.

21.5.4 Better Disaster Recovery Management

The process of preparing for any disaster through routine server backups and recovery is made simpler and faster by the server virtualization because the virtual infrastructure essentially consists of software and files. So backing up of these is a lot easier and far less time-consuming than doing it on several individual machines. Moreover, hardware failures like hard disk failures do not affect virtual machines in the same way they would a physical machine.

21.5.5 Software Development Testing and Verification

If there is any software that is being either developed in-house or outsourced that will run on the business infrastructure, it is easier and cheaper to test it on the virtual infrastructure and verify its compatibility with the infrastructure and all other business processes before deploying it on a live system.

21.5.6 Isolation of Legacy Applications

With virtualization, there is no longer the drive to get rid of any useful software package just because it requires a legacy infrastructure or it is not compatible with newer software versions. Virtualization enables the creation of an isolated server environment where all these legacies can still gainfully function without retarding and constraining the company business.

21.5.7 Cross-Platform Support

Lastly but of great value is the platform flexibility that virtualization brings about that makes it easy to run software packages that would normally otherwise be run on only one specific platform, for example, to run a Windows-based software on a virtual machine running on a Mac physical machine and the other way round.

21.5.8 Minimizing Hardware Costs

One thing that causes more pain in African system management is first acquisition and upgrading of both hardware and software and maintaining these resources in good working conditions. When it comes to maintaining network equipment, this further creates a constant problem. For large institutions and businesses, the costs of keeping all servers and other hardware in top working conditions are always higher than in other parts of the world. Virtualization eases this burden of purchasing more hardware each time a new system is put in place. Why because one server can be used in place of several servers.

21.5.9 Faster Server Provisioning

It is always difficult to have a good estimate of how many servers may be needed especially during those times when there is unseasonal demand. Virtualization gives an answer to being always ready to meet the challenges of unseasonal demands by using its elastic capacity to provide system provisioning and deployment at a moment's notice.

21.5.10 Better Load Balancing

Each virtualization server runs a load balancer—a software that effectively spreads out network traffic among multiple systems, thus avoiding horrible network jams. Network traffic is easily dispersed to multiple systems, virtual or physical by the load balancer.

21.5.11 Reduce the Data Center Footprint

In addition to saving more on energy with smaller energy bills, server consolidation with virtualization will also reduce the overall footprint of the entire data center because data is now on fewer servers, requiring less networking gear, hence a smaller number of racks needed [2].

21.5.12 Increase Uptime

Most server virtualization platforms now offer a number of advanced features such as live migration, storage migration, fault tolerance, high availability, and distributed resource scheduling. These technologies give the virtual machines the ability to quickly recover from unplanned outages. In addition, modern virtualization software has the ability to quickly and easily move a virtual machine from one server to another. There will be more and better capabilities with newer virtualization software [2].

21.5.13 Isolate Applications

Virtualization technology has removed the old requirement of a "one app/one server." This requirement used to cause physical server sprawl, and increased costs, and underutilized servers. This also cuts down on server waste by more fully utilizing the physical server resources and by provisioning virtual machines with the exact amount of CPU, memory, and storage resources that it needs [2].

21.5.14 Extend the Life of Older Applications

Let's be honest—you probably have old legacy applications still running in your environment. These applications probably fit into one or more of these categories: It doesn't run on a modern operating system, it may not run on newer hardware, your IT team is afraid to touch it, and chances are good that the person or company who created it is no longer around to update it.

By virtualizing and encapsulating a legacy application and its environment, we can extend its life, maintain uptime, and finally get rid of that old and costly machines such an application used to run on, thus extending its life [2].

There are of course many other benefits, but we cannot discuss them all here.

21.6 Virtualization Infrastructure Security

To understand virtualization security problems and appreciate the efforts being made to protect any virtualized infrastructure, one has to remember that virtualization technology is based on software. So all security problems and vulnerabilities ever been encountered in any software product have the potential to be in a virtualized infrastructure. This opens up a very broad area of attach for those interested in securing virtualized infrastructures. To narrow the focus, it is important and probably more feasible to concentrate on specific major components of a virtualization infrastructure like the hypervisor, hosts, communication pathways, and probably users. These major focus points can be secured to the best of known security protocols and best practices. More specifically, the focus should be put on the understanding that all virtual infrastructures are based on physical port gateways so if we tighten security on those entry points, we can go a long way in securing the virtual infrastructure. So our first points of interest are those points where certain types of network traffic go within the physical network. We focus on these first because network traffic into and out of the virtual infrastructure goes through these points. The restriction of traffic into and out of the virtual infrastructure through a few of these designated points also offers additional security of the virtual resources from unauthorized users from outside of the virtual infrastructure access gateway ring. Security within the virtual infrastructure is also enhanced by the growing inclusion and migration into the virtual infrastructure of security components that were traditionally hardware-based firewall, VPN, and others, thus ensuring that virtual infrastructure customers can themselves extend the enforcement of security and compliance requirements of their physical network into the virtual environments.

Perhaps the greatest threat presented by virtualization to computer networks is the fact that using one physical computer, one can access many virtual infrastructures, a feat that is not so feasible in the physical networks. According to Gruman quoting Simard [7], "graphics cards and network cards today are really miniature computers that see everything in all the VMs." They could be used as spies across all the VMs, letting a single PC spy on multiple networks.

21.6.1 Hypervisor Security

In Sect. 21.3.3 we defined a virtualization hypervisor as a virtual machine manager software program that allows multiple operating systems to share a single physical hardware host. Besides its traditional role of creating and managing VMs, the hypervisor is also responsible for the security between the virtual machines. However, the security provided to the virtual infrastructure is not enough. One has to remember again that the hypervisor is still a software package that is prone to all software threats and vulnerabilities as usual.

21.6.2 Securing Communications Between Desktop and Virtual Infrastructure

This is an old problem with probably similar security threats and vulnerabilities and the same protocols and best practices with communications between two or more physical network infrastructures. In this particular case, we are focusing on the pathways between the desktop and the virtual infrastructure. Securing these pathways is essential in order to prevent eavesdropping, data leakage, and man-in-the-middle attacks. Best practices today for securing these pathways include SSH, SSL, and IPsec [8].

21.6.3 Security of Communication Between Virtual Machines

In a virtual infrastructure, every host has a virtual switch. This virtual switching component manages and directs all inter-VM traffic that go via the host. This virtual switch creates a potential threat to all virtual machines connected to this host. Although this is the case, standard protocols and best practices enjoyed in physical network router infrastructure for network monitoring and intrusion detection can still be deployed and successfully used in the virtual switching environment.

21.6.4 Threats and Vulnerabilities Originating from a VM

We have been talking only about threats and vulnerabilities that are pumped upstream from the workstations, the hypervisor, and the host machines into the virtual machines. There is also a high potential for threats and vulnerabilities originating from the individual virtual machines and spreading downstream to the hypervisor, the hosts, and the desktops. The good news is that most of these problems can be handled by current best practices including protocols and vendor patches.

Exercises

1. What is a virtual switching element?
2. Why should a small business opt to virtualize its computing resources?
3. In recent years, there has been a phenomenal growth in the business use of computing virtualization technology. What are the biggest challenges you see to the technology in its future growth?
4. Although there has been tremendous growth in the virtualization of computing resources, there are still many skeptics of the technology. List their concerns. Suggest ways to overcome those concerns.
5. Discuss the differences between desktop and server virtualization. Which one of the two is most likely to benefit a small business?
6. Discuss the differences between virtualization and emulation by giving examples.

Advanced Exercises

1. Discuss the connection between virtualization and cloud computing.
2. In the chapter we discussed the pros of virtualization, discuss the cons of virtualization.
3. Compare and contrast the two most popular virtualization software packages.
4. From the knowledge you have acquired in this chapter about virtualization, discuss the future of virtualization as a business model.
5. Compare and contrast the security concerns in a virtual network infrastructure and a physical network infrastructure.
6. Virtual PC from Microsoft Corp. is a free virtualization software that can start you going for a free VMs on Windows XP or Windows 2003 server. Download Virtual PC and create a few VMs on your Windows.
7. Sun xVM VirtualBox is also a free virtualization software. And it is open source best for small networks. Download Sun xVM and set up a couple of VMs.
8. Try out the following:
   (a) Citrix Xen
   (b) Linux KVM
9. QEMU is a free emulation software that runs on a limited number of architectures including x86 and x86-64. Try QUEM.

References

1. VMware.com
2. Mullins R (2012) Virtualization tops CIO priorities in 2012: IDC savings from server consolidation will go to new IT innovations, IDC says, InformationWeek. January 11, 2012. http://www.informationweek.com/news/storage/virtualization/232400150
3. http://www.infobarrel.com/History_of_Virtualization#ixzz1l9armMAL
4. History of virtualization http://www.everythingvm.com/content/history-virtualization
5. Workstation Virtualization Featuring VMware Workstation 7.0/7.1. http://mv4t.com/Virtualization_VMware-Workstation.php
6. Wikipedia: http://en.wikipedia.org/wiki/Comparison_of_platform_virtual_machines)
7. Gruman G (2008) Virtualization's secret security threats: virtualization can be both a blessing and a curse, serving up improved security while at the same time hiding dangers. InfoWorld, March 13, 2008, http://www.infoworld.com/d/security-central/virtualizations-secret-security-threats-159?page=0,0
8. Shackleford D (2010) An introduction to virtualization security, SANS – Tuesday, 9 March 2010. http://www.net-security.org/article.php?id=1397&p=2
9. Wikipedia. http://en.wikipedia.org/wiki
10. Kartheek. Virtualization technology. http://katireddykartheek.blogspot.com/