National Institute of
Standards and Technology
U.S. Department of Commerce

Special Publication 800-125

Guide to Security for Full Virtualization Technologies

Recommendations of the National Institute of Standards and Technology

Karen Scarfone
Murugiah Souppaya
Paul Hoffman

NIST Special Publication 800-125

Guide to Security for Full Virtualization Technologies

Recommendations of the National Institute of Standards and Technology

Karen Scarfone
Murugiah Souppaya
Paul Hoffman

COMPUTER SECURITY

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

January 2011

U.S. Department of Commerce

Gary Locke, Secretary

National Institute of Standards and Technology

Patrick D. Gallagher, Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-125
Natl. Inst. Stand. Technol. Spec. Publ. 800-125, 35 pages (January 2010)

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

Acknowledgments

The authors, Karen Scarfone of G2, Inc., Murugiah Souppaya of the National Institute of Standards and Technology (NIST), and Paul Hoffman of the VPN Consortium, wish to thank their colleagues who reviewed drafts of this document and contributed to its technical content. The authors gratefully acknowledge and appreciate the contributions from individuals and organizations whose comments improved the overall quality of this publication.

Trademark Information

Table of Contents

List of Appendices

Executive Summary

Virtualization is the simulation of the software and/or hardware upon which other software runs. This simulated environment is called a *virtual machine (VM)*. There are many forms of virtualization, distinguished primarily by computing architecture layer. This publication focuses on the form of virtualization known as full virtualization. In *full virtualization*, one or more OSs and the applications they contain are run on top of virtual hardware. Each instance of an OS and its applications runs in a separate VM called a *guest operating system*. The guest OSs on a host are managed by the *hypervisor.* which controls the flow of instructions between the guest OSs and the physical hardware, such as CPU, disk storage, memory, and network interface cards. The hypervisor can partition the system's resources and isolate the guest OSs so that each has access to only its own resources, as well as possible access to shared resources such as files on the host OS. Also, each guest OS can be completely encapsulated, making it portable. Some hypervisors run on top of another OS, which is known as the *host operating system*.

The recent increase in the use of full virtualization products and services has been driven by many benefits. One of the most common reasons for adopting full virtualization is operational efficiency: organizations can use their existing hardware (and new hardware purchases) more efficiently by putting more load on each computer. In general, servers using full virtualization can use more of the computer's processing and memory resources than servers running a single OS instance and a single set of services. A second common use of full virtualization is for desktop virtualization, where a single PC is running more than one OS instance. Desktop virtualization can provide support for applications that only run on a particular OS. It allows changes to be made to an OS and subsequently revert to the original if needed, such as to eliminate changes that negatively affect security. Desktop virtualization also supports better control of OSs to ensure that they meet the organization's security requirements.

Full virtualization has some negative security implications. Virtualization adds layers of technology, which can increase the security management burden by necessitating additional security controls. Also, combining many systems onto a single physical computer can cause a larger impact if a security compromise occurs. Further, some virtualization systems make it easy to share information between the systems; this convenience can turn out to be an attack vector if it is not carefully controlled. In some cases, virtualized environments are quite dynamic, which makes creating and maintaining the necessary security boundaries more complex.

This publication discusses the security concerns associated with full virtualization technologies for server and desktop virtualization, and provides recommendations for addressing these concerns. Most existing recommended security practices remain applicable in virtual environments. The practices described in this document build on and assume the implementation of practices described in other NIST publications.

To improve the security of server and desktop full virtualization technologies, organizations should implement the following recommendations:

Secure all elements of a full virtualization solution and maintain their security.

The security of a full virtualization solution is heavily dependent on the individual security of each of its components, from the hypervisor and host OS (if applicable) to guest OSs, applications, and storage. Organizations should secure all of these elements and maintain their security based on sound security practices, such as keeping software up-to-date with security patches, using secure configuration baselines, and using host-based firewalls, antivirus software, or other appropriate mechanisms to detect and stop attacks. In general, organizations should have the same security controls in place for virtualized operating systems as they have for the same operating systems running directly on hardware. The same is true for applications running on guest OSs: if the organization has a security policy for an application, it should apply the same regardless of whether the application is running on an OS within a hypervisor or on an OS running on hardware.

Restrict and protect administrator access to the virtualization solution.

The security of the entire virtual infrastructure relies on the security of the virtualization management system that controls the hypervisor and allows the operator to start guest OSs, create new guest OS images, and perform other administrative actions. Because of the security implications of these actions, access to the virtualization management system should be restricted to authorized administrators only. Some virtualization products offer multiple ways to manage hypervisors, so organizations should secure each management interface, whether locally or remotely accessible. For remote administration, the confidentiality of communications should be protected, such as through use of FIPS-approved cryptographic algorithms and modules.

Ensure that the hypervisor is properly secured.

Securing a hypervisor involves actions that are standard for any type of software, such as installing updates as they become available. Other recommended actions that are specific to hypervisors include disabling unused virtual hardware; disabling unneeded hypervisor services such as clipboard- or file-sharing; and considering using the hypervisor's capabilities to monitor the security of each guest OS running within it, as well as the security of activity occurring between guest OSs. The hypervisor itself also needs to be carefully monitored for signs of compromise. It is also important to provide physical access controls for the hardware on which the hypervisor runs. For example, hosted hypervisors are typically controlled by management software that can be used by anyone with access to the keyboard and mouse. Even bare metal hypervisors require physical security: someone who can reboot the host computer that the hypervisor is running on might be able to alter some of the security settings for the hypervisor.

Carefully plan the security for a full virtualization solution before installing, configuring, and deploying it.

Planning helps ensure that the virtual environment is as secure as possible and in compliance with all relevant organizational policies. Security should be considered from the initial planning stage at the beginning of the systems development life cycle to maximize security and minimize costs. It is much more difficult and expensive to address security after deployment and implementation.

1. Introduction

1.1 Authority

The National Institute of Standards and Technology (NIST) developed this document in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), "Securing Agency Information Systems," as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided in A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright, though attribution is desired.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority, nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

1.2 Purpose and Scope

The purpose of the guide is to discuss the security concerns associated with full virtualization technologies for server and desktop virtualization, and to provide recommendations for addressing these concerns. All forms of virtualization other than server and desktop full virtualization are outside the scope of this document.

Most existing recommended security practices remain applicable in virtual environments. The practices described in this document build on and assume the implementation of practices described in other NIST publications.

1.3 Audience

The intended audience for this document is system and security administrators, security program managers, information system security officers, and others who have responsibilities for or are otherwise interested in the security of server or desktop full virtualization technologies.

This document assumes that readers have some operating system, networking, and security expertise. Because of the constantly changing nature of full virtualization technologies, readers are encouraged to take advantage of other resources (including those listed in this document) for more current and detailed information.

1.4 Document Structure

The remainder of this document is organized into the following major sections:

- Section 2 presents an introduction to full virtualization technologies and describes server and desktop full virtualization.

- Section 3 discusses the security characteristics of full virtualization technologies.
- Section 4 provides security recommendations for virtualization components.
- Section 5 highlights security considerations of particular interest throughout the lifecycle of full virtualization solutions.

The document also contains appendices with supporting material:

- Appendix A defines terms used in this document.
- Appendix B contains a list of acronyms and abbreviations used in this document.

2. Introduction to Full Virtualization

Virtualization is the simulation of the software and/or hardware upon which other software runs. This simulated environment is called a *virtual machine (VM)*. There are many forms of virtualization, distinguished primarily by computing architecture layer. For example, *application virtualization* provides a virtual implementation of the application programming interface (API) that a running application expects to use, allowing applications developed for one platform to run on another without modifying the application itself. The Java Virtual Machine (JVM) is an example of application virtualization; it acts as an intermediary between the Java application code and the operating system (OS). Another form of virtualization, known as *operating system virtualization*, provides a virtual implementation of the OS interface that can be used to run applications written for the same OS as the host, with each application in a separate VM container.

Application virtualization and operating system virtualization are outside the scope of this publication. This publication focuses on the form of virtualization known as full virtualization. In *full virtualization*, one or more OSs and the applications they contain are run on top of virtual hardware. Each instance of an OS and its applications runs in a separate VM called a *guest operating system*. The guest OSs on a host are managed by the *hypervisor*, also called the *virtual machine monitor (VMM)*, which controls the flow of instructions between the guest OSs and the physical hardware, such as CPU, disk storage, memory, and network interface cards. The hypervisor can partition the system's resources and isolate the guest OSs so that each has access to only its own resources, as well as possible access to shared resources such as files on the host OS. Also, each guest OS can be completely encapsulated, making it portable. Some hypervisors run on top of another OS, which is known as the *host operating system*.

In full virtualization the hypervisor provides most of the same hardware interfaces as those provided by the hardware's physical platform. This means that the OSs and applications running within full virtualization do not need to be modified for virtualization to work if the OSs and applications are compatible with the underlying hardware. An interesting twist on full virtualization is *paravirtualization*, which is a method for the hypervisor to offer interfaces to the guest OS that the guest OS can use instead of the normal hardware interfaces. If a guest OS can use paravirtualized interfaces, they offer significantly faster access for resources such as hard drives and networks. Different types of paravirtualization are offered by different hypervisor systems.

This section provides an overview of full virtualization as a foundation for the rest of the publication. It also explains the two common use cases for full virtualization: server virtualization and desktop virtualization.

2.1 Motivations for Full Virtualization

The recent increase in the use of full virtualization products and services has been driven by many benefits. One of the most common reasons for adopting full virtualization is operational efficiency: organizations can use their existing hardware (and new hardware purchases) more efficiently by putting more load on each computer. In general, servers using full virtualization can use more of the computer's processing and memory resources than servers running a single OS instance and a single set of services. Recent advances in CPU architectures have made full virtualization faster than it was just a few years ago, and similar advances are expected to continue to be made both by CPU vendors and virtualization software vendors. Also, CPU architecture changes have made full virtualization more secure by strengthening hypervisor restrictions on resources.

A second common use of full virtualization is for desktop virtualization, where a single PC is running more than one OS instance. There are several reasons for deploying desktop virtualization. It can provide support for applications that only run on a particular OS. It allows changes to be made to an OS and subsequently revert to the original if needed, such as to eliminate changes that negatively affect security. Desktop virtualization also supports better control of OSs to ensure that they meet the organization's security requirements. This control can be asserted by creating a high-assurance platform that constantly updates the guest OS to have the exact versions of the programs that it is authorized to have, and no other programs.

A more recent use of desktop virtualization is to enable the use of applications that only run on an older version of an OS when the user's desktop is running a newer version. In such a situation, desktop virtualization is useful for continuity of applications as the OSs advance faster than the applications that run on them. As more applications become web-based, desktop virtualization can become even more important: a web application that only runs on an older version of a particular browser can be run in a virtualized system that has the older version of that browser, while the user's main environment is running the newer (usually more secure) version of the browser. For use cases such as this, many organizations use application virtualization instead of desktop virtualization.

Full virtualization has some negative security implications. Virtualization adds layers of technology, which can increase the security management burden by necessitating additional security controls. Also, combining many systems onto a single physical computer can cause a larger impact if a security compromise occurs. Further, some virtualization systems make it easy to share information between the systems; this convenience can turn out to be an attack vector if it is not carefully controlled. In some cases, virtualized environments are quite dynamic, which makes creating and maintaining the necessary security boundaries more complex.

2.2 Types of Full Virtualization

There are two forms of full virtualization. Figure 2-1 compares their high-level architectures. In *bare metal virtualization*, also known as *native virtualization*, the hypervisor runs directly on the underlying hardware, without a host OS; the hypervisor can even be built into the computer's firmware. In the other form of full virtualization, known as *hosted virtualization*, the hypervisor runs on top of the host OS; the host OS can be almost any common operating system such as Windows, Linux, or MacOS. Hosted virtualization architectures usually also have an additional layer of software (the *virtualization application*) running in the guest OS that provides utilities to control the virtualization while in the guest OS, such as the ability to share files with the host OS. Hosted virtualization architectures also allow users to run applications such as web browsers and email clients alongside the hosted virtualization application, unlike bare metal architectures, which can only run applications within virtualized systems.

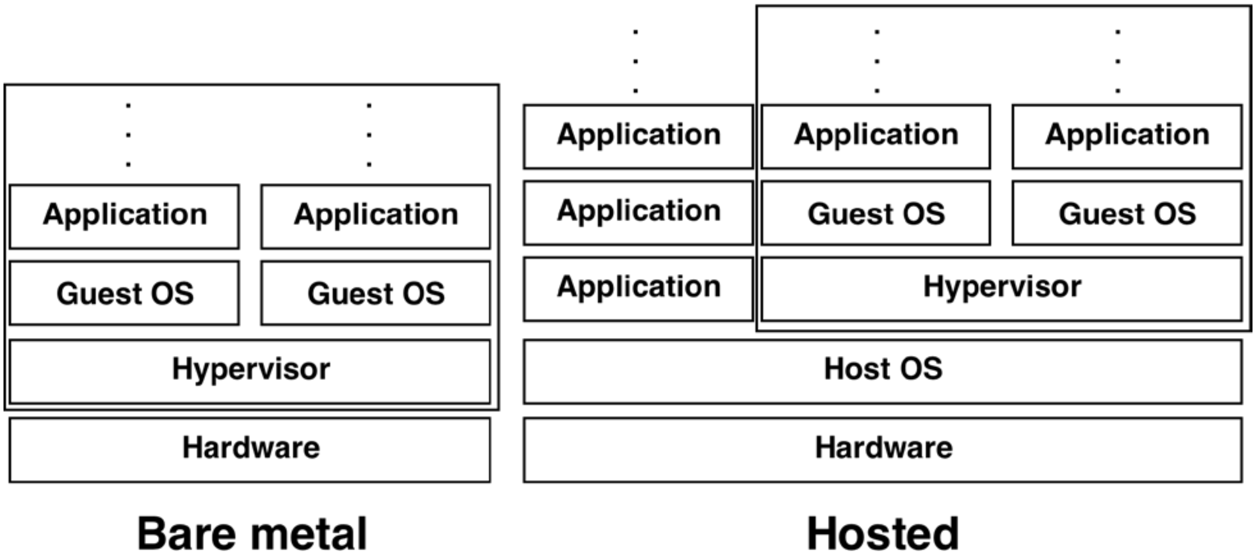

Figure 2-1. Full Virtualization Architectures

Servers are most often virtualized on computers using bare metal virtualization. Desktops are most often virtualized on computers with hosted virtualization. In both bare metal and hosted virtualization, each guest OS appears to have its own hardware, like a regular computer. This includes:

- CPU
- Memory
- Storage (hard disk, and possibly floppy and CD-ROM drives)
- Storage controllers
- Ethernet controllers
- Display and sound devices
- Keyboard and mouse

Many virtualization environments offer additional virtual hardware, such as USB controllers, parallel ports for printing, and serial ports. Some hypervisors allow paravirtualization of some hardware interfaces, most commonly the storage controller and Ethernet controllers. Some hypervisors also provide direct memory access (DMA) to high-speed storage controllers and Ethernet controllers, if such features are supported in the hardware CPU on which the hypervisor is running. DMA access from guest OSs can significantly increase the speed of disk and network access, although this type of acceleration prevents some useful virtualization features such as snapshots and moving guest OSs while they are running.

Deciding between bare metal and hosted virtualization—whether or not to have a host OS—is an important operational and security decision. Adding a hypervisor on top of a host OS adds more complexity and more vulnerabilities to the host. However, a hypervisor is much simpler and smaller than a host OS, so it provides a smaller target. Choosing bare metal virtualization by replacing a host OS with a hypervisor may improve security, depending on how well-secured the hypervisor is, while adding a hypervisor on top of a host OS tends to increase risk. Organizations should balance security and functionality when deciding whether or not a host OS should be used under a server or desktop virtualization solution. They should also take into account that bare metal hypervisors run on a much more limited range of hardware than hosted hypervisors; for example, bare metal hypervisors often work on only a limited number of Ethernet controllers and graphics cards.

Hardware emulation (sometimes called *hardware translation*) is a type of hosted virtualization. The primary difference is that in hardware emulation, the hypervisor provides different hardware interfaces from those provided by the physical hardware. Because the hypervisor in hardware emulation can simulate all of the hardware required by the guest OS, it can run unmodified OSs designed for platforms different from the host platform. For example, early versions of VirtualPC allowed users to run the Microsoft Windows OS on the PowerPC processor supported by the Apple MacOS platform. Similarly, Apple supplies the Rosetta software with its Intel Mac OS X platform, allowing programs designed for the PowerPC version of Mac OS X to run on the Intel Mac platform.

2.3 Virtualizing Hardware

For full virtualization to be effective, the virtualized hardware presented to the guest OS must resemble physical hardware extremely closely. In addition, virtualization systems must offer additional features for the virtualized hardware to help it integrate well with the physical hardware in an organization's network. This section discusses virtualized networking and storage, as well as how a guest OS is encapsulated.

2.3.1 Virtualized Networking

Full virtualization hypervisors can provide networking capabilities, allowing the individual guest OSs to communicate with one another while simultaneously limiting access to the external physical network. The network interfaces that the guest OSs see may be virtual, physical, or both. Typical hypervisors offer three primary forms of network access:

- **Network Bridging.** The guest OS is given direct access to the host's network interface cards (NIC) independent of the host OS.
- **Network Address Translation (NAT).** The guest OS is given a virtual NIC that is connected to a simulated NAT inside the hypervisor. As in a traditional NAT, all outbound network traffic is sent through the virtual NIC to the host OS for forwarding, usually to a physical NIC on the host system.
- **Host Only Networking.** The guest OS is given a virtual NIC that does not directly route to a physical NIC. In this scenario, guest OSs can be configured to communicate with one another and, potentially, with the host OS.

When a number of guest OSs exist on a single host, the hypervisor can provide a virtual network for these guest OSs. The hypervisor may implement virtual switches, hubs, and other network devices. Using a hypervisor's networking for communications between guests on a single host has the advantage of greatly increased speed because the packets never hit physical networking devices. Internal host-only networking can be done in many ways by the hypervisor. In some systems, the internal network looks like a virtual switch. Others use virtual LAN (VLAN) standards to allow better control of how the guest systems are connected. Most hypervisors also provide internal network address and port translation (NAPT) that acts like a virtual router with NAT.

Networks that are internal to a hypervisor's networking structure can pose an operational disadvantage, however. Many networks rely on tools that watch traffic as it flows across routers and switches; these tools cannot view traffic as it moves in a hypervisor's network. There are some hypervisors that allow network monitoring, but this capability is generally not as robust as the tools that many organizations have come to expect for significant monitoring of physical networks. Some hypervisors provide APIs that allow a privileged VM to have full visibility to the network traffic. Unfortunately, these APIs may also provide additional ways for attackers to attempt to monitor network communications. Another concern with network monitoring through a hypervisor is the potential for performance degradation or denial of service conditions to occur for the hypervisor because of high volumes of traffic.

The security implications of networks internal to a hypervisor should not be minimized. For example, assume that an organization has two computers, one that acts as a public-facing web server and another that is an internal database server. The organization also monitors the switch that connects the two computers, watching for traffic that would indicate an attack on the database. If both of those servers were moved onto a single hypervisor, and the hypervisor's virtual network was used for communications between the servers for increased efficiency, the ability to monitor all the traffic between the two systems would be lost unless the hypervisor itself can perform this monitoring that meets the organization's security policies.

To get around this loss of visibility, some organizations purposely expose network traffic between virtualized hosts to the physical network already in place in the organization. This requires the system on which the hypervisor is running to have multiple network interfaces, and this may significantly slow network communications as compared to a virtual-only network, but the advantage is that the organization does not need to change its security policies to gain the cost advantages of virtualization. Organizations should consider the tradeoffs between traffic being hidden within a hypervisor and the extra overhead and risk of exposing that traffic but being able to control it using the same tools already used for controlling other network traffic. See Section 3.5 for additional information.

2.3.2 Virtualized Storage

Hypervisor systems have many ways of simulating disk storage for guest OSs. All hypervisors, at a minimum, have virtual hard drives, while some of them also have more advanced virtual storage options. In addition, some hypervisors can use advanced storage interfaces on the host system, such as network-attached storage (NAS) and storage area networks (SAN) to present different storage options to the guest OSs. This section describes those options.

All hypervisors in common use present the guest OSs with virtual hard drives though the use of disk images. A *disk image* is a file on the host that looks to the guest OS like an entire disk drive. Whatever the guest OS writes onto the virtual hard drive goes into the disk image. With hosted virtualization, the disk image appears in the host OS as a file or a folder, and it can be handled like other files and folders.

Most virtualization systems also allow a guest OS to access physical hard drives as if they were connected to the guest OS directly. This is different than using disk images in that a disk image is a virtual representation of a real drive. Direct access is common for floppy and CD-ROM drives that are attached to the host OS, so that a guest OS can, for example, install new software from a CD-ROM that is inserted in the host computer. Some hypervisors also allow a guest OS to connect to an entire physical hard drive. The main advantage of using physical hard drives is that accessing them is much faster than accessing disk images.

Many computers can access NAS and SAN systems. Some hypervisors can present these systems to the guest OSs as a NAS or SAN, while others can make those systems appear as virtual drives. This is an active area of development in the virtualization market, and new types of storage virtualization are being added to hypervisors frequently.

The security implications of using virtual storage are essentially the same as using real storage. Access to the various types of storage that a guest OS has access to should be controlled as it would be if the storage were being used by a full computer. Of course, using disk backups as part of a security strategy is just as important with virtual computers as it is with non-virtual computers, so organizations should incorporate backups of virtualized storage into their backup policies. In addition, access to the virtual storage can be controlled at the host and VM level. Existing authentication and authorization mechanisms is leveraged to restrict user access to the file and object resources according to the organization policy.

2.3.3 Guest OS Images

A full virtualization hypervisor encapsulates all of the components of a guest OS, including its applications and the virtual resources they use, into a single logical entity. An *image* is a file or a directory that contains, at a minimum, this encapsulated information. Images are stored on hard drives, and can be transferred to other systems the same way that any file can (note, however, that images are often many gigabytes in size). Some virtualization systems use a virtualization image metadata standard called the Open Virtualization Format (OVF)[1] that supports interoperability for image metadata and components across virtualization solutions.

A *snapshot* is a record of the state of a running image, generally captured as the differences between an image and the current state. For example, a snapshot would record changes within virtual storage, virtual memory, network connections, and other state-related data. Snapshots allow the guest OS to be suspended and subsequently resumed without having to shut down or reboot the guest OS. Many, but not all, virtualization systems can take snapshots.

On some hypervisors, snapshots of the guest OS can even be resumed on a different host. While a number of issues may be introduced to handle real-time migration, including the transfer delay and any differences that may exist between the two physical servers (e.g., IP address, number of processors or hard disk space), most live-migration solutions provide mechanisms to resolve these issues. Should the target system use the same virtualization product, many of these issues will not arise. However, live migration across heterogeneous hypervisors may introduce potential configuration errors that may affect the security of the guest OS.

2.4 Full Virtualization Use Cases

Full virtualization solutions have two major use cases: server virtualization and desktop virtualization. These are described below.

2.4.1 Server Virtualization

Virtualizing a server can provide some security benefits. Running a server within a hypervisor provides a sandbox, which can limit the impact of a compromise, and the hypervisor might provide a smaller attack surface than a host operating system would, reducing the possibility of expanding a successful compromise outside the guest OS. However, server virtualization does not prevent attackers from compromising the server through vulnerabilities in the server application or the guest OS, nor does it prevent attackers from directly compromising the host OS (if present), such as attacking the host OS's network services from another host on the same subnet. Most importantly, virtualizing multiple servers on the same host tends to negatively affect security because of the logical proximity of the servers and the potential impact of a single compromise affecting all the servers on a host.

The discussions below address common reasons for using single server and multiple server virtualization.

[1] The specification for OVF version is published by the Distributed Management Task Force, Inc. (DMTF).

2.4.1.1 Single Server Virtualization

A common use case for single server virtualization is supporting a service that only runs on a legacy OS that cannot be properly secured on its own. For example, common security controls may not be available for the legacy OS. If the service and legacy OS are run as a guest OS, the hypervisor or host OS may be able to monitor the guest OS's actions using various security controls that the legacy OS itself cannot. Also, an additional layer of authentication and auditing could be added, such as at the host OS level. Such monitoring can be built into the organization's security policies.

2.4.1.2 Multiple Server Virtualization

For many years, organizations have typically deployed each important service to its own dedicated host so as to better isolate each server from the others and prevent a compromise of one server or host from granting control to other servers. However, having many hosts is costly (space, power consumption, maintenance, hardware, etc.), so organizations have been adopting server virtualization so that they can host multiple services on a single host, with each service in a separate guest OS to enforce security requirements. When a service experiences higher-than-normal use, the hypervisor can coordinate requests among guest OSs and other hypervisors to ensure that resources are properly distributed. Services that are rarely used can be kept in a saved state by the guest OS and loaded by the guest OS on demand. This frees up resources for other guest OSs. Also, new servers can be deployed without the need to configure and deploy as much new dedicated hardware.

Another benefit for data centers is that a hypervisor can present resources to the guest OSs as a single entity, such as showing multiple hard drives as a single storage entity. This allows individual resources to be added or removed from the system transparently, without modifying individual guest OSs. A single read-only image can be shared among many servers. Additionally, there are benefits for configuration management, because each guest OS can be derived from a parent guest OS, providing a consistent security baseline and reducing the time and effort required to configure and patch individual servers. Organizations taking advantage of virtualization may benefit from improved incident handling; servers can be reverted to an uninfected state quickly, while the complete state (including RAM) of the compromised guest OS can be saved in a snapshot for later examination.

However, there can be substantial security risks in consolidating multiple services within a single hypervisor. For example, a critical service is usually placed on its own dedicated host so that the host can be secured specifically for that service and so that a compromise of any other service would not impact the critical service. By placing a critical service on a host with other services, both of those goals are impacted. It is particularly risky to place multiple services on a host if they have significantly different security needs. For example, suppose that one service is considered critical and is secured very strongly, while another service on the same host is considered low-impact and is secured relatively weakly. An attacker wanting to compromise the critical service could compromise the low-impact service and use the fact that it is local on the virtual network to attempt to access the critical service or to compromise the hypervisor and thus gain access to the critical service. Organizations that have policies relating to allocation of computer resources should consider virtualization in such policies.

When multiple server virtualization is used for running servers on many hosts and for moving servers from host to host based on changing resource needs, it can be called cloud computing. *Cloud computing* is "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."[2] Full virtualization is a core enabling technology for cloud infrastructure as a service architectures. NIST describes cloud computing and related topics in detail at http://csrc.nist.gov/groups/SNS/cloud-computing/. Further discussion of cloud computing is outside the scope of this document.

2.4.2 Desktop Virtualization

One of the most common reasons for using desktop virtualization is to allow a user to run applications for different OSs on a single host. Without virtualization, this would be accomplished by using multiple devices, each with a different OS, or by configuring a single device to boot using multiple OSs and using one OS (and application) at a time. Desktop virtualization allows users to access both OSs simultaneously on one computer.

Another common use for desktop virtualization is allowing organizations to more tightly control the environment of its users. The organization stores a known-good image that contains the OS and all the applications needed for the user. The user loads this image using hosted virtualization and does all their work within this image, not on the host OS, then exits the guest OS. Later the user restarts the guest OS, causing any previous changes to the guest OS to be lost. The advantage of this quit-and-restart strategy is that malicious changes that have been introduced to the OS or applications are erased by quitting.

With desktop virtualization, the user's data is normally stored on the host or on the network; otherwise, it would be lost each time the user quits from the virtualization system. This aspect of desktop virtualization can be the most frustrating and complicated for users; it is quite easy to store new documents on what appears to be the correct place, only to discover later that the document was lost because it was stored in the wrong place. Some desktop virtualization systems have methods to deal with making sure that users' data is correctly stored before quitting, but these programs are not foolproof. For example, many Windows programs store valuable information in the Windows Registry, and such data is often hard to find and back up correctly. Organizations should test the data backup facilities of any desktop virtualization system they use for all software that the desktop users are expected to use.

[2] Mell, Peter and Grance, Tim, "The NIST Definition of Cloud Computing", Version 15, October 7, 2009, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

3. Virtualization Security Overview

Migrating computing resources to a virtualized environment has little or no effect on most of the resources' vulnerabilities and threats. For example, if a service has inherent vulnerabilities and that service is moved from a non-virtualized server to a virtualized server, the service is still just as vulnerable to exploitation. However, the use of virtualization may help reduce the impact of such exploitation—but virtualization may also provide additional attack vectors, thus increasing the likelihood of successful attacks. Many of the features of virtualization offer both benefits and disadvantages to security.

This section describes these security implications. Section 3.1 discusses the isolation of guest OSs from each other and the underlying hypervisor and host OS. Section 3.2 explains the purpose of and mechanisms for guest OS monitoring. Section 3.3 discusses image and snapshot management.

3.1 Guest OS Isolation

The hypervisor is responsible for managing guest OS access to hardware (e.g., CPU, memory, storage). The hypervisor *partitions* these resources so that each guest OS can access its own resources but cannot encroach on the other guest OSs' resources or any resources not allocated for virtualization use. This prevents unauthorized access to resources and also helps prevent one guest OS from injecting malware into another, such as infecting a guest OS's files or placing malware code into another guest OS's memory. Separately, partitioning can also reduce the threat of denial of service conditions caused by excess resource consumption in other guest OSs on the same hypervisor.

Resources may be partitioned physically or logically. In *physical partitioning*, the hypervisor assigns separate physical resources to each guest OS, such as disk partitions, disk drives, and network interface cards. *Logical partitioning* may divide resources on a single host or across multiple hosts as in a pool of resources with the same security impact level categorization, allowing multiple guest OSs to share the same physical resources, such as processors and RAM, with the hypervisor mediating access to the resources. Physical partitioning sets hard limits on resources for each guest OS because unused capacity from one resource may not be accessed by any other guest OS. However, having physical separation for resources may provide stronger security and improved performance than logical partitioning. Many virtualization systems can do both physical and logical partitioning. Some organizations have policies about which application data can physically reside on drives with the data of other applications, and such policies should take into account physical and logical partitioning in hypervisors.

Having separate partitions for resource is an important part of isolating guest OSs. Isolation also involves limiting guest OS communications and the access that each guest OS has to the other guest OSs, to the hypervisor, and to the host OS (if present). Hypervisors can theoretically support a level of logical isolation nearly equivalent to physical isolation, mediating all communications from each guest OS to have full control over each guest OS's actions. Hypervisors can permit interactions between guest OSs as needed, such as allowing two desktop OSs to share a file system. Hypervisors can also dynamically alter isolation for each guest OS as needed—for example, enabling and disabling networking at specific times. Isolation has obvious security benefits, but it can also increase the reliability of a host by preventing actions in one guest OS from directly affecting another. For example, if one guest OS crashes because of an application fault or an attack, the other guest OSs on that host are unlikely to be affected. Isolating each guest OS from the others and restricting what resources they can access and what privileges they have is also known as *sandboxing*.

Another motivation for isolating guest OSs from each other and the underlying hypervisor and host OS is the mitigation of *side-channel attacks*. These attacks exploit the physical properties of hardware to reveal information about usage patterns for memory access, CPU use, and other resources. A common goal of these attacks is to reveal cryptographic keys. These attacks are considered difficult, usually requiring direct physical access to the host.

Attackers may attempt to break out of a guest OS so that they can access the hypervisor, other guest OSs, or the underlying host OS. Breaking out of a guest OS is also known as *escape*. If an attacker can successfully escape a guest OS and gain access to the hypervisor, the attacker might be able to compromise the hypervisor and gain control over all of its guest OSs. So the hypervisor provides a single point of security failure for all the guest OSs; a single breach of the hypervisor places all the guest OSs at high risk.

Guest OSs are often not completely isolated from each other and from the host OS because that would prevent necessary functionality. For example, many hosted virtualization solutions provide mechanisms called *guest tools* through which a guest OS can access files, directories, the copy/paste buffer, and other resources on the host OS or another guest OS. These communication mechanisms can inadvertently serve as an attack vector, such as transmitting malware or permitting an attacker to gain access to particular resources. Bare metal virtualization software does not offer such sharing capabilities.

3.2 Guest OS Monitoring

The hypervisor is fully aware of the current state of each guest OS it controls. As such, the hypervisor may have the ability to monitor each guest OS as it is running, which is known as *introspection*. Introspection can provide full auditing capabilities that may otherwise be unavailable. Monitoring capabilities provided through introspection can include network traffic, memory, processes, and other elements of a guest OS. For many virtualization products, the hypervisor can incorporate additional security controls or interface with external security controls and provide information to them that was gathered through introspection. Examples include firewalling, intrusion detection, and access control. Many products also allow the security policy being enforced through hypervisor-based security controls to be moved as a guest OS is migrated from one physical host to another.

Network traffic monitoring is particularly important when networking is being performed between two guest OSs on the host or between a guest OS and the host OS. Under typical network configurations, this traffic does not pass through network-based security controls, so host-based security controls should be used to monitor the traffic instead.

3.3 Image and Snapshot Management

Creating guest machine images and snapshots does not affect the vulnerabilities within them, such as the vulnerabilities in the guest OSs, services, and applications. However, images and snapshots do affect security in several ways, some positive and some negative, and they also affect IT operations.

Note that one of the biggest security issues with images and snapshots is that they contain sensitive data (such as passwords, personal data, and so on) just like a physical hard drive. Because it is easier to move around an image or snapshot than a hard drive, it is more important to think about the security of the data in that image or snapshot. Snapshots can be more risky than images because snapshots contain the contents of RAM memory at the time that the snapshot was taken, and this might include sensitive information that was not even stored on the drive itself.

An operating system and applications can be installed, configured, secured, and tested in a single image and that image then distributed to many hosts. This can save considerable time, providing additional time for the contents of the image to be secured more effectively, and also improve the consistency and strength of security across hosts. However, because images can be distributed and stored easily, they need to be carefully protected against unauthorized access, modification, and replacement. Some organizations need to have a small number of known-good images of guest OSs that differ, for example, based on the application software that is installed.

As the use of server and desktop virtualization grows within an organization, the management of images can become a significant challenge. Some virtualization products offer management solutions that can examine stored images and update them as needed, such as applying patches and making security configuration changes, but other products offer no way of applying updates other than loading each image. For these products, the longer an image is stored without running it, the more vulnerabilities it is likely to contain when it is loaded again. It may be necessary to track all images and ensure that each non-archival image is periodically updated. Tracking images may also be a significant problem, particularly if users and administrators are able to create their own images. These images may also not be secured properly, especially if they are not based on a security baseline (e.g., the one provided by a different pre-secured image). This could increase the risk of compromise.

Another potential problem with increasing the use of virtualization in particular is the proliferation of images, also known as *sprawl*. It is easy to create a new image—it can often be done in just a few minutes, albeit without any consideration of security—so unnecessary images may be created and run. Each additional image running is another potential point of compromise for an attacker. Also, each additional image is another image that has to have its security maintained. Therefore, organizations should minimize the creation, storage, and use of unnecessary images. Organizations should consider implementing formal image management processes that govern image creation, security, distribution, storage, use, retirement, and destruction, particularly for server virtualization. Similar consideration should be given to snapshot management. In some cases, organizations have policies to not allow storage of snapshots because of the risk of malware from infected systems being stored in snapshots and later reloaded.

Image management can provide significant security and operational benefits to an organization. For example, if the contents of an image become compromised, corrupted, or otherwise damaged, the image can quickly be replaced with a known good image. Also, snapshots can serve as backups, permitting the rapid recovery of information added to the guest OS since the original image was deployed. One of the drawbacks associated with this type of backup is that incremental or differential backups of the system may not be feasible unless those backups are supported by the hypervisor. If a modification is made to the guest OS after a snapshot has been captured, the original snapshot will not include the modification, and a new snapshot will need to be applied. Because of this, snapshot management needs to be considered as part of image management.

If an image has been compromised, its encapsulated nature means that it can easily be preserved for forensic purposes. Also, a guest OS can be suspended quickly, which causes a snapshot to be recorded that captures the entire state of a compromised guest OS, including the complete contents of RAM, then stops the guest OS to prevent the compromise from spreading to other guest OSs or hosts. In traditional environments, it is more difficult to capture the complete contents of RAM during or after an attack. Often, multiple steps must be performed before the data can be captured, potentially leading to the loss of important information.

Image files can be monitored to detect unauthorized changes to the image files; this can be done by calculating cryptographic checksums for each file as it is stored, then recalculating these checksums periodically and investigating the source of any discrepancies. Image files can also be scanned to detect rootkits and other malware that, when running, conceal themselves from security software present within the guest OS.

In some virtualization systems, guest OSs can be moved from one host computer to another when needed, such as when a host needs to be rebooted or shut off for maintenance work, when a partial host failure or an attack against the hypervisor or OS is detected, when there is a strong expectation of an impending attack. This can lower the pressure to perform upgrades and replacements quickly, thus reducing inconvenience to system administrators and providing more time for testing the changes. For server virtualization, the hypervisor may be able to move its guest OSs to other host computers automatically; on some VM systems, this can happen when the virtual machines are still running, and do not require a shutdown or suspend of the guest operating system. For desktop virtualization, manual actions are generally needed. A storage network is dedicated to perform these migrations or the transport channel is fully authenticated and encrypted to preserve the integrity of the VMs and prevent information leakage.

For virtualization involving multiple physical servers, guest OS migration supports load balancing by allowing dynamic control over which host each virtualized server is running on at any given time. For example, if a particular host is being heavy utilized, nearly to the point of resource exhaustion, one or more of its guest OSs could be transferred to hosts with lower utilization. This prevents denial of service conditions, but is most often used simply to improve performance of the guest OSs.

A potential drawback of using guest OS migration is that if a guest OS has been compromised or contains malicious code, but this malicious activity has not been detected, the guest OS could be migrated to another host and could compromise that host. The same problem occurs when converting a compromised physical system to a virtual machine.

The use of images can improve software testing practices. An organization can test an application on multiple OSs without needing separate hardware for each OS. Additional physical test machines may still be necessary to test hardware compatibility. Organizations should not depend solely on the results of tests provided in a virtual environment. The virtualization environment may provide some functionality or protections that do not exist on the target environment, potentially resulting in inaccurate results. In particular, due to the overhead required for virtualization, load testing in a virtual environment may not provide the same results as load testing in a physical environment. In addition to ensuring that the test guest OS image is configured the same as the target environment, organizations should ensure that additional tests will be performed on physical hardware.

Organizations can maintain known-good copies of each guest OS in a single place, allowing testers to take advantage of a “fresh” copy of the guest OS for each test that restores the system to the desired baseline. This allows testers to ensure that the test environment’s configuration matches that of the production environment and that the effects of performing one test do not inadvertently affect the results of a subsequent test. Also, through virtualization, testers can have access to multiple configurations and platforms to test applications, software updates or patches in a secure, confined environment. By properly configuring the guest OS, any configuration available on a production system can be replicated. In some situations, testing can be performed on an exact copy of the production guest OS. In all these cases, images can be used to good effect. Images holding an entire guest OS can be replicated for each fresh copy, and many organizations keep their images on shared storage so that many departments can access them easily.

4. Security Recommendations for Virtualization Components

The security of a full virtualization solution is heavily dependent on the individual security of each of its components, including the hypervisor, host computer, and host OS (if applicable), guest OSs, applications, and storage. Organizations should secure all of these elements and maintain their security based on sound security practices, such as restricting access to administrative interfaces, keeping software up-to-date with security patches, using secure configuration baselines, performing monitoring and analysis of logs at all layers of the solution, and using host-based firewalls, antivirus software, or other appropriate mechanisms to detect and stop attacks.

Following these recommendations alone is not sufficient to secure a virtualization solution. Virtualization can be used in many ways, so the appropriate security controls for each situation vary. This section discusses common threats against virtualization solutions and provides recommendations for countering these threats. With this information, organizations will be able to apply the risk management framework outlined in NIST SP 800-37 Revision 1, *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* to more accurately assess the risks associated with virtualization.

In general, organizations should have the same security controls in place for the virtualized operating systems as they have for the same operating systems running directly on hardware. The same is true for applications running on guest OSs: if the organization has a security policy for an application, it should apply the same regardless of whether the application is running on an OS within a hypervisor or on an OS running on hardware.

4.1 Hypervisor Security

The programs that control the hypervisor should be secured using methods similar to those used to protect other software running on desktops and servers. The security of the entire virtual infrastructure relies on the security of the *virtualization management system* that controls the hypervisor and allows the operator to start guest OSs, create new guest OS images, and perform other actions. Because of the security implications of these actions, access to the virtualization management system should be restricted to authorized administrators only. Some virtualization management systems allow different level of access to different users, such as giving some users read-only access to the administrative interface of a guest OS, other users control over particular guest OSs, and yet other users complete administrative control. Most hypervisor software currently only uses passwords for access control; this may be too weak for some organizations' security policies and may require the use of compensating controls, such as a separate authentication system used for restricting access to the host on which the virtualization management system is installed.

Hypervisors can be managed in different ways, with some hypervisors allowing management through multiple methods. It is important to secure each hypervisor management interface, both locally and remotely accessible. The capability for remote administration can usually be enabled or disabled in the virtualization management system. If remote administration is enabled in a hypervisor, access to all remote administration interfaces should be restricted by a firewall. Also, hypervisor management communications should be protected. One option is to have a dedicated management network that is separate from all other networks and that can only be accessed by authorized administrators. Management communications carried on untrusted networks must be encrypted using FIPS-approved methods, provided by either the virtualization solution or a third-party solution, such as a virtual private network (VPN) that encapsulates the management traffic.

Because of the hypervisor's level of access to and control over the guest OSs, limiting access to the hypervisor is critical to the security of the entire system. The access options vary based on hypervisor type. Most bare metal hypervisors have access controls to the system. Typically, the access method is just username and password, but some bare metal hypervisors offer additional controls such as hardware token-based authentication to grant access to the hypervisor's management interface. On some systems, there are different levels of authorization, such as allowing some users to view logs but not be able to change any settings or interact directly with the guest OSs. These view-only user accounts allow auditors and others to have sufficient access to meet their needs without reducing overall security.

In contrast to bare metal solutions, hosted virtualization products rarely have hypervisor access controls: anyone who can launch an application on the host OS can run the hypervisor. The only access control is whether or not someone can log into the host OS. Because of this wide disparity in security, organizations should have security policies about which guest OSs can be run from bare metal hypervisors and which can be run from hosted virtualization hypervisors. Further, organizations running bare metal hypervisors should have policies specifying who can and cannot access various features of the hypervisor.

The following are security recommendations for the hypervisor itself:

- Install all updates to the hypervisor as they are released by the vendor. Most hypervisors have features that will check for updates automatically and install the updates when found. Centralized patch management solutions can also be used to administer updates.

- Restrict administrative access to the management interfaces of the hypervisor. Protect all management communication channels using a dedicated management network or the management network communications is authenticated and encrypted using FIPS 140-2 validated cryptographic modules.

- Synchronize the virtualized infrastructure to a trusted authoritative time server.

- Disconnect unused physical hardware from the host system. For example, a removable disk drive might be occasionally used for backups, but it should be disconnected when not actively being used for backup or restores. Disconnect unused NICs from any network.

- Disable all hypervisor services such as clipboard- or file-sharing between the guest OS and the host OS unless they are needed. Each of these services can provide a possible attack vector. File sharing can also be an attack vector on systems where more than one guest OS share the same folder with the host OS.

- Consider using introspection capabilities to monitor the security of each guest OS. If a guest OS is compromised, its security controls may be disabled or reconfigured so as to suppress any signs of compromise. Having security services in the hypervisor permits security monitoring even when the guest OS is compromised.

- Consider using introspection capabilities to monitor the security of activity occurring between guest OSs. This is particularly important for communications that in a non-virtualized environment were carried over networks and monitored by network security controls (such as network firewalls, security appliances, and network IDPS sensors).

- Carefully monitor the hypervisor itself for signs of compromise. This includes using self-integrity monitoring capabilities that hypervisors may provide, as well as monitoring and analyzing hypervisor logs on an ongoing basis.

Of course, it is also important to provide physical access controls for the hardware on which the virtualization system runs. For example, hosted hypervisors are typically controlled by management software that can be used by anyone with access to the keyboard and mouse. Even bare metal hypervisors require physical security: someone who can reboot the host computer that the hypervisor is running on could alter some of the security settings for the hypervisor. It is also important to secure the external resources that the hypervisor uses, particularly data on hard drives and other storage devices.

There are additional recommendations for hosted virtualization solutions for server virtualization. Hosted virtualization exposes the system to more threats because of the presence of a host OS. To increase the security of the host OS, minimize the number of applications other than the hypervisor that are ever run on the system. All unneeded applications should be removed. Those that remain should be restricted as much as possible to prevent malware from being inadvertently installed on the system. For example, a web browser is often used to download updates to the hypervisor, and also to read instructions and bulletins about the hypervisor. If the computer is intended to be exclusively used to run the hosted hypervisor, the web browser should have as many settings as possible adjusted to their highest security level.

Because hosted virtualization systems are run under host OSs, the security of every guest OS relies on the security of the host OS. This means that there should be tight access controls to the host OS to prevent someone from gaining access through the host OS to the virtualization system and possibly changing its settings or modifying the guest OSs.

There has been some concern in the security community about designing hypervisors so that they cannot be detected by attackers. The motivation for this is to provide an additional layer of security that is invisible to the attacker, thus preventing successful attacks against the hypervisor and the host OS below it. However, hypervisors have various characteristics that permit attackers to detect their presence. Detection techniques include checking for hypervisor artifacts in processes, file system, registry, or memory; checking for hypervisor-specific processor instructions or capabilities; and checking for hypervisor-specific virtual hardware devices. These detection techniques are hypervisor implementation-dependent. Although hypervisor detection can be deterred by a vendor modifying the hypervisor's implementation or hiding its identifiable software artifacts, it is not possible to completely hide all characteristics. When planning their virtualization security, organizations should not assume that attackers will not be able to detect the presence of a hypervisor or the product type and version.

4.2 Guest OS Security

A guest OS running in a virtualized environment acts almost identically to the OS running on real hardware. All of the security considerations that apply to OSs running on real hardware also apply to guest OSs; however, there are some additional security considerations for guest OSs.

To run in a virtual machine, a guest OS needs to use video, sound, keyboard, mouse, and network hardware drivers that are specific to the hypervisor. There are no specific security issues with such drivers unless they have programming bugs that are not present in the normal drivers.

More importantly, many hosted virtualization systems allow guest OSs to share information with the host OS through shared disks or folders, which are normally created by emulating networked disks. In either case, if one guest OS has been compromised by malware, it might spread the malware through the shared disk or folder. This is a security vulnerability that does not exist on regular OSs unless they have shared network storage. Organizations that have security policies that cover network shared storage should apply those policies to shared disks in virtualization systems.

Many hosted virtualization systems also allow guest OSs to share information with the host OS through clipboard sharing. That is, copying information to the clipboard in the host OS allows that information to be pasted in the guest OS, and vice versa. Similarly, putting information on the clipboard in one guest OS makes the same information show up on the clipboard in other guest OSs running on the same hypervisor. This is a handy feature for users, but it is also a vector for attacks between the guest OS and host OS. Because of this, organizations should have policies regarding the use of shared clipboards.

The following are security recommendations for the guest OS itself:

- Follow the recommended practices for managing the physical OS, e.g., time synchronization, log management, authentication, remote access, etc.
- Install all updates to the guest OS promptly. All modern OSs have features that will automatically check for updates and install them.
- Back up the virtual drives used by the guest OS on a regular basis, using the same policy for backups as is used for non-virtualized computers in the organization.
- In each guest OS, disconnect unused virtual hardware. This is particularly important for virtual drives (usually virtual CDs and floppy drives), but is also important for virtual network adapters other than the primary network interface and serial and/or parallel ports.
- Use separate authentication solutions for each guest OS unless there is a particular reason for two guest OSs to share credentials.
- Ensure that virtual devices for the guest OS are associated only with the appropriate physical devices on the host system, such as the mappings between virtual and physical NICs.

If a guest OS on a hosted virtualization system is compromised, that guest OS can potentially infect other systems on the same hypervisor. The most likely way this can happen is that both systems are sharing disks or clipboards. If such sharing is turned on in two or more guest OSs, and one guest OS is compromised, the administrator of the virtualization system needs to decide how to deal with the potential compromise of other guest OSs. Two strategies for dealing with this situation are:

- Assume that all guest OSs on the same hardware have been compromised. Revert each guest OS to a known-good image that was saved before the compromise.
- Investigate each guest OS for compromise, just as one would during normal scanning for malware. If malware is found, follow the organization's normal security policy.

The first method assumes that guest OSs are different than "regular" systems, while the second assumes that the organization's current security policy is sufficient and should be applied to all systems in the same manner.

4.3 Virtualized Infrastructure Security

Virtualization provides simulation of hardware such as storage and network interfaces. This infrastructure is as important to the security of a virtualized guest OS as real hardware infrastructure is to an operating system running on a physical computer. Many virtualization systems have features to provide access control to the virtual hardware, particularly storage and networking. Access to virtual hardware should be strictly limited to the guest OSs that will use it. For example, if a virtual hard drive will be shared between two guest OSs, only those two OSs should have access to the virtual hard drive. Some virtual hardware is meant to be widely shared. For example, a disk image that represents an installation CD may be shared among many guest OSs; still, access to that image should be read-only, and no guest image should have write access to it.

Hypervisor systems that connect multiple guest OSs together on a virtual network present issues for organizations whose policies require that all networks be monitored in specified fashions. For example, an organization might have a network security policy that says that all network switches connecting multiple servers must be managed and that traffic between the servers be monitored for suspicious activity. However, network switches in most virtual systems do not have such a capability. Some virtual switches support virtual LAN (VLAN) and firewall capabilites to provide separation and isolation of the VM network traffic. In some environments, additional security appliances can be implemted to inspect, control, shape, and monitor the VM network communications in a centralized location.

Hypervisors sometimes offer virtual storage networks and virtual interfaces to existing hardware storage networks. These features offer the same security problems as virtual networks, namely that organizations whose security policies require monitoring those connections cannot use the same methods for virtual storage as they do for physical storage. Using physical interfaces to existing networked storage can eliminate this problem, but also reduces some of the flexibility that hypervisors offer.

4.4 Desktop Virtualization Security

A major difference in security between server and desktop virtualization is the ability to control the images. In a server environment, the ability to create and manage images is usually limited to administrators. But in desktop environments, end users often have the ability to create, modify, duplicate, and delete images. The virtualization software itself may also be fully controlled by the user. It may not be possible for the organization to ensure that the guest OS images meet the organization's security policy requirements, such as being patched regularly.

Organizations considering the use of desktop virtualization should determine which scenarios require the enforcement of security by managed virtualization solutions and which scenarios do not require centralized management. For example, if a teleworker is using desktop virtualization to run programs that the security policy would allow them to run from, say, a lightly-protected home computer, then that system probably does not need to be as tightly managed as one that accesses internal databases or websites, and therefore would only be allowed from computers with more stringent security controls. Organizations often manage virtual machines like they are real computers; another option is to treat them as appliances that expire (or are forced out of service) after a period of time and replaced by more up-to-date appliances.

Desktop virtualization can be used to improve security by providing a well-secured guest OS image for the desktop environment. A number of virtualization vendors provide solutions that will allow organizations to deploy a managed desktop guest OS on unmanaged computers. For example, telecommuting employees may install a hypervisor on their home computer and access the organization's intranet through a specific guest OS image, or a remote access server might deliver a clean guest OS image every time a user initiates a remote access session. Some solutions even permit users to boot their home computers from removable media containing a hypervisor and guest OS image; this can provide a bare metal full virtualization solution that does not run the host OS on the home computer. Guest OS images on read-only media are not a panacea, however. Guest OSs are often updated, which means that the old read-only media would need to be destroyed and new media created and distributed. Because of this, some organizations might be tempted to use rewritable media instead, but that could lead to the media being infected with malware.

Organizations typically take advantage of these types of desktop virtualization solutions to reduce the security concerns associated with connecting unmanaged systems to internal resources, as well as to lessen dependence on distributing managed computers to individuals and trying to ensure that unmanaged computers meet security requirements. An often overlooked concern about checking personally owned computers is that scans and other checks may affect privacy. For the data and resources the guest OS accesses, it can provide some protection from threats in the host OS—for example, by establishing a VPN to the organization and by encrypting stored data. However, it cannot fully protect against threats in the host OS unless the host OS is bypassed altogether (e.g., by booting the computer to run a hypervisor from removable media).

Another benefit of using managed guest OS images is that they can be updated by the organization as needed without requiring user intervention. However, image distribution can be problematic because a single guest OS image can be many gigabytes in size, making it difficult to download. Organizations may choose to lessen the frequency of full image updates by configuring images to patch and update their operating systems and applications automatically. Organizations that manage guest OSs for multiple users should also be particularly careful that any changes made by one user do not propagate back to the main image and then appear in the images used by other users.

Another common use of desktop virtualization is supporting an application that only runs on a legacy OS that cannot be properly secured on its own. In this case, the hypervisor or host OS may be able to monitor the guest OS's actions using various compensating controls that the legacy OS cannot run. Legacy applications may have vulnerabilities that may be exposed if it is granted network access. Legacy applications (and the OSs on which they run) may also be more susceptible to insider attack due to a lack of stringent auditing mechanisms. An additional layer of authentication and auditing could be added, such as at the host OS level. Virtualization can also be used to strengthen network communications involving the legacy applications—options for accessing an application include accessing a standalone guest OS via a console and using a remote connection protocol to the guest OS. In cases where the legacy application must be provided full network access, care must be taken to ensure that the data it receives is not malicious and, when policy requires, that the information is encrypted and signed.

5. Secure Virtualization Planning and Deployment

A critical aspect of deploying a secure virtualization solution is careful planning prior to installation, configuration, and deployment. This helps ensure that the virtual environment is as secure as possible and in compliance with all relevant organizational policies. Many virtualization security and performance problems can be traced to a lack of planning or management controls. Security should be considered from the initial planning stage at the beginning of the systems development life cycle to maximize security and minimize costs. It is much more difficult and expensive to address security after deployment and implementation. The guidelines found in Section 4 and other general resources on security practices should aid organizations in making appropriate decisions for deploying virtualization solutions.

This section brings together the concepts presented in the previous sections of the guide and explains how they should be incorporated throughout the entire life cycle of virtualization solutions, involving everything from policy to operations. This section references a five-phase life cycle model to help organizations determine at what point in their virtualization deployments a recommendation may be relevant. This model is based on one introduced in NIST SP 800-64, *Security Considerations in the Information System Development Life Cycle*. Organizations may follow a project management methodology or life cycle model that does not directly map to the phases in the model presented here, but the types of tasks in the methodology and their sequencing are probably similar. The phases of the life cycle are as follows:

- **Phase 1: Initiation.** This phase includes the tasks that an organization should perform before it starts to design a virtualization solution. These include identifying needs for virtualization, providing an overall vision for how virtualization solutions would support the mission of the organization, creating a high-level strategy for implementing virtualization solutions, developing virtualization policy, identifying platforms and applications that can be virtualized, and specifying business and functional requirements for the solution.

- **Phase 2: Planning and Design.** In this phase, personnel specify the technical characteristics of the virtualization solution and related components. These include the authentication methods and the cryptographic mechanisms used to protect communications. At the end of this phase, solution components are procured.

- **Phase 3: Implementation.** In this phase, equipment is configured to meet operational and security requirements, installed and tested as a prototype, and then activated on a production network. Implementation includes altering the configuration of other security controls and technologies, such as security event logging, network management, and authentication server integration.

- **Phase 4: Operations and Maintenance.** This phase includes security-related tasks that an organization should perform on an ongoing basis once the virtualization solution is operational, including log review, attack detection, and incident response.

- **Phase 5: Disposition.** This phase encompasses tasks that occur when a virtualization solution is being retired, including preserving information to meet legal requirements, sanitizing media, and disposing of equipment properly.

This section highlights security considerations of particular interest for virtualization solutions. These considerations are not intended to be comprehensive, nor is there any implication that security elements not listed here are unimportant or unnecessary. In addition to following the security recommendations presented in this publication, organizations implementing full virtualization solutions should also follow the recommendations from NIST SP 800-53, *Recommended Security Controls for Federal Information*

Systems and Organizations, which defines minimum recommended management, operational, and technical controls for information systems based on impact categories.

5.1 Initiation

The initiation phase involves many preparatory actions, such as identifying current and future needs, and specifying requirements for performance, functionality, and security. A critical part of the initiation phase is the development of a virtualization security policy. The section lists elements that a virtualization security policy should contain and, where relevant, describes some of the factors that should be considered when making the decisions behind each element. A virtualization security policy should define which forms of virtualization the organization permits and which types of applications and data are permitted to be used with each form of virtualization. It should also cover how the organization's virtualization solutions are administered and how their policies are updated. Note that some virtualization security policies will inherently prevent certain types of virtualization from being selected during the initiation process.

In addition to the considerations described in this section for virtualization security policies, organizations should also consider how other security policies may be affected by virtualization and adjust these policies as needed to take virtualization into consideration.

Organizations should be aware of how their use of virtualization may affect the security categorization of the physical system. The security categories associated with Federal information system based on three security objectives: confidentiality, integrity and availability. These security categories are described in NIST FIPS 199, *Standards for Security Categorization of Federal Information and Information Systems.* The security categorization of a particular information system depends on the potential impact associated with a loss of confidentiality, integrity or availability. If a system hosts guest OSs with different impact levels, the system should be secured in accordance with the highest of those levels. The organization's virtualization security policy should define how combining multiple guest OSs on a single system affects the system's security requirements, both positively and negatively, and which combinations of guest OSs are permitted or prohibited. Organizations may also choose to reduce risk by prohibiting combinations that include resources accessing particular types of information, such as highly sensitive personally identifiable information (PII).[3]

Every year, there are many changes in virtualization capabilities, the security controls available to organizations, the types of threats against different types of virtualization, and so on. Therefore, organizations should periodically reassess their policies for virtualization. Organizations should also be aware of the emergence of new types of virtualization solutions and of major changes to existing virtualization technologies, and ensure that policies are updated accordingly as needed.

5.2 Planning and Design

Once the organization has established a virtualization security policy, identified virtualization needs, and completed other preparatory activities, the next step is to determine which types of virtualization technologies should be used and to design a solution to deploy. There are many considerations for designing a solution, most of which are generally applicable to any IT technology. This section focuses on the technical security considerations that are most important for designing virtualization solutions. Major considerations include the following:

[3] For more information on protecting PII, see NIST SP 800-122, *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII).*

- **Architecture.** Designing the architecture includes the placement of the virtualization solution and the selection of virtualization software. It also includes placement and selection of storage, network topology, bandwidth availability, management systems, etc.
- **Authentication.** This involves determining which layers of the virtualization solution (e.g., application/server, guest OS, hypervisor, host OS) need separate authentication mechanisms and selecting, implementing and maintaining those mechanisms.
- **Cryptography.** Decisions related to cryptography include selecting the algorithms for encryption and integrity protection of virtualization communications, and setting the key strength for algorithms that support multiple key lengths. These decisions affect many aspects of virtualized systems, including the type of authentication used at different levels of management and the protection of virtual machines when they are stored on disk.

Testing should be performed to make sure that servers, storage, and the network infrastructure can accommodate virtualization. For example, virtualization may require more powerful hardware platforms that currently exist in the organization, and moving disparate servers into a single location may cause network bandwidth problems. Also consider whether the networked environment would benefit from application streaming.

The security aspects of the virtualization solution design should be documented in the system security plan. The organization should also consider how incidents involving the virtualization solutions should be handled and document those plans as well.[4] An incident response team can save a snapshot of the guest OS to capture the contents of memory, the hard disk, and state information. Many forensics tools can directly examine the contents of a snapshot, allowing full forensic analysis of the image itself. The incident response team may also be able to run a copy of the snapshot within a separate environment to determine what the effects of the attack were. Organizations should not assume that incidents are contained by virtualization; an attacker may have escaped the guest OS, potentially compromising the hypervisor and other guest OSs, so it may be necessary to shut down or disconnect the host quickly to achieve containment.

Running OSs and applications in virtualized environments sometimes has effects on software licensing. This is not directly a security issue, but losing a license for running software can cause loss of data or access to data. An example of a problem is a license that is tied to a NIC address or CPU characteristic; moving the software to a virtualized environment could force relicensing. Many products have changed their licensing terms to address virtualization, but others have not. Before placing software in a virtualized environment, organizations should ensure that such use would be compatible with existing licensing agreements and, if not, discuss with their software vendor how to minimize the possible loss of data or access to their data.

5.3 Implementation

After the virtualization solution has been designed, the next step is to implement and test a prototype of the design before putting the solution into production. Aspects of the solution that should be evaluated include the following:

- **Physical to Virtual Conversion.** Existing servers and desktops may need to be migrated to guest OSs. Most hypervisors provide tools for doing this quickly and automatically. It is important to verify that the tools are compatible with the OSs running on the physical and the virtual machines. There

[4] For more information on incident handling, see NIST SP 800-61 Revision 1, *Computer Security Incident Handling Guide*.

may also be problems with loading the guest OS (e.g., boot device errors) or communicating properly with the hypervisor.

- **Introspection.** Determine whether the virtualization solution gives the necessary capability to monitor security events occurring within the guest OSs. If not, deploying additional security monitoring, such as with security devices, may need to be added and tested before the overall system is deployed.
- **Authentication.** Authentication is required at each of the appropriate solution layers and cannot be readily compromised or circumvented. If token-based authentication is needed in virtualized systems, test whether or not the tokens work in the emulated hardware.
- **Connectivity.** Users can connect to all of the resources that they are permitted to and cannot connect to any other resources. Each traffic flow is protected, if necessary, in accordance with the organization's established requirements.
- **Applications.** The virtualization solution does not interfere with the use of applications or servers within the guest OSs.
- **Networking.** The internal networking offered by the virtualization solution can be configured to conform to the organization's security policy. This includes the ability to monitor communications between guest OSs and to block particular types of traffic.
- **Management.** Administrators can configure and manage the solution effectively and securely. This includes all components, including hypervisors and images.
- **Performance.** The solution provides adequate performance during normal and peak usage. This is particularly important for server virtualization and may require the use of simulated traffic generators to mimic the actual characteristics of expected traffic as closely as possible. Test all appropriate load characteristics, such as CPU, network, and storage load.
- **Security of the Implementation.** The virtualization implementation itself may contain vulnerabilities and weaknesses that attackers could exploit. Organizations with high security needs may choose to perform extensive vulnerability assessments against the virtualization components. At a minimum, all components should be updated with the latest patches and configured following sound security practices.

5.4 Operations and Maintenance

Operational processes that are particularly important for maintaining virtualization security, and thus should be performed regularly, include the following:

- **Administration.** Being sure that only authorized administrators have physical access to the hypervisor hardware and logical access to the hypervisor software and host OS.
- **Staying Current.** Check for upgrades and patches to the hypervisor, each guest OS, the host OS (if relevant), and all application software running on each guest OS and the host OS. Acquire, test, and deploy the updates to all systems in the virtualized environment.
- **Time Synchronization.** Ensure that each virtualization component has its clock synched to a common time source so that its timestamps will match those generated by other systems. In some virtualization systems, guest OSs can be synchronized with the host OS. At a minimum, use the guest OS's built-in time synchronization to synchronize with a reliable time server.

- **Control.** Reconfigure access control features as needed based on factors such as policy changes, technology changes, audit findings, and new security needs

- **Logging.** Document anomalies detected within the virtualized environment. Such anomalies might indicate malicious activity or deviations from policy and procedures. Anomalies should be reported to other systems' administrators as appropriate.

Organizations should periodically perform assessments to confirm that the organization's virtualization policies, processes, and procedures are being followed properly. Assessment activities may be passive, such as reviewing logs, or active, such as performing vulnerability scans and penetration testing. Assessments need to be made at all levels of the virtualized infrastructure, including the host and guest OSs, the hypervisor, and shared storage media. More information on technical assessments is available from NIST SP 800-115, *Technical Guide to Information Security Testing and Assessment*.

5.5 Disposition

Before a device using virtualization permanently leaves an organization (such as when a leased server's lease expires or when an obsolete PC is being recycled), the organization should remove any sensitive data from the host. Data may also need to be wiped if an organization provides "loaner" devices to teleworkers, particularly for travel. The task of scrubbing all sensitive data from storage devices is often surprisingly difficult because of all the places where such data resides. See NIST SP 800-88, *Guidelines for Media Sanitization*, for additional information and recommendations on removing data from devices. Note that sensitive data may be found nearly anywhere on a device because of the nature of virtualization. An organization should strongly consider erasing all storage devices completely.

Appendix A—Glossary

Selected terms used in the publication are defined below.

Application virtusalization: A virtual implementation of the application programming interface (API) that a running application expects to use.

Bare metal virtualization: A form of full virtualization where the hypervisor runs directly on the underlying hardware, without a host operating system.

Disk image: A virtual representation of a real disk drive.

Escape: The act of breaking out of a guest OS to gain access to the hypervisor, other guest OSs, or the underlying host OS.

Full virtualization: A form of virtualization where one or more operating systems and the applications they contain are run on top of virtualized hardware.

Guest operating system: A virtual machine that runs an instance of an OS and its applications.

Guest tools: Mechanisms within hosted virtualization solutions that allow a guest OS to access files, directories, the copy/paste buffer, and other resources on the host OS or another guest OS.

Host operating system: In a hosted virtualization solution, the OS that the hypervisor runs on top of.

Hosted virtualization: A form of full virtualization where the hypervisor runs on top of a host OS.

Hypervisor: The virtualization component that manages the guest OSs on a host and controls the flow of instructions between the guest OSs and the physical hardware.

Image: A file or directory that contains, at a minimum, the encapsulated components of a guest OS.

Logical partitioning: The hypervisor allowing multiple guest OSs to share the same physical resources.

Native virtualization: See "bare metal virtualization".

Operating system virtualization: A virtual implementation of the OS interface that can be used to run applications written for the same OS.

Paravirtualization: A method for a hypervisor to offer interfaces to a guest OS that the guest OS can use instead of the normal hardware interfaces.

Partitioning: Managing guest operating system access to hardware so that each guest OS can access its own resources but cannot encroach on the other guest OSs' resources or any resources not allocated for virtualization use.

Physical partitioning: The hypervisor assigning separate physical resources to each guest OS.

Sandboxing: Isolating each guest OS from the others and restricting what resources they can access and what privileges they have.

Snapshot: A record of the state of a running image, generally captured as the differences between an image and the current state.

Sprawl: The proliferation of images.

Virtual machine (VM): A simulated environment created by virtualization.

Virtual machine monitor (VMM): See “hypervisor”.

Virtualization: The simulation of the software and/or hardware upon which other software runs.

Appendix B—Acronyms and Abbreviations

This appendix contains a list of selected acronyms and abbreviations used in the guide.

API	Application Programming Interface
CPU	Central Processing Unit
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
IDPS	Intrusion Detection and Prevention System
IP	Internet Protocol
IT	Information Technology
ITL	Information Technology Laboratory
JVM	Java Virtual Machine
LAN	Local Area Network
MAC	Media Access Control
NAPT	Network Address and Port Translation
NAS	Network Attached Storage
NAT	Network Address Translation
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
OS	Operating System
OVF	Open Virtualization Format
PC	Personal Computer
PII	Personally Identifiable Information
RAM	Random Access Memory
SAN	Storage Area Network
SP	Special Publication
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Monitor
VPN	Virtual Private Network