Virtual Machine Security Guidelines

Version 1.0

September 2007

http://cisecurity.org

cis-feedback@cisecurity.org

Editor: Joel Kirch

WBB Consulting

TERMS OF USE AGREEMENT

August 2006

TERMS OF USE AGREEMENT

Background.

The Center for Internet Security ("**CIS**") provides benchmarks, scoring tools, software, data, information, suggestions, ideas, and other services and materials from the CIS website or elsewhere ("**Products**") as a public service to Internet users worldwide. Recommendations contained in the Products ("**Recommendations**") result from a consensus-building process that involves many security experts and are generally generic in nature. The Recommendations are intended to provide helpful information to organizations attempting to evaluate or improve the security of their networks, systems, and devices. Proper use of the Recommendations requires careful analysis and adaptation to specific user requirements. The Recommendations are not in any way intended to be a "quick fix" for anyone's information security needs.

No Representations, Warranties, or Covenants.

CIS makes no representations, warranties, or covenants whatsoever as to (i) the positive or negative effect of the Products or the Recommendations on the operation or the security of any particular network, computer system, network device, software, hardware, or any component of any of the foregoing or (ii) the accuracy, reliability, timeliness, or completeness of the Products or the Recommendations. CIS is providing the Products and the Recommendations "as is" and "as available" without representations, warranties, or covenants of any kind.

User Agreements.

By using the Products and/or the Recommendations, I and/or my organization ("**We**") agree and acknowledge that:

1. No network, system, device, hardware, software, or component can be made fully secure;

2. We are using the Products and the Recommendations solely at our own risk;

3. We are not compensating CIS to assume any liabilities associated with our use of the Products or the Recommendations, even risks that result from CIS's negligence or failure to perform;

4. We have the sole responsibility to evaluate the risks and benefits of the Products and Recommendations to us and to adapt the Products and the Recommendations to our particular circumstances and requirements;

5. Neither CIS, nor any CIS Party (defined below) has any responsibility to make any corrections, updates, upgrades, or bug fixes; or to notify us of the need for any such corrections, updates, upgrades, or bug fixes; and

6. Neither CIS nor any CIS Party has or will have any liability to us whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages (including without limitation loss of profits, loss of sales, loss of or damage to reputation, loss of customers, loss of software, data, information or emails, loss of privacy, loss of use of any computer or other equipment, business interruption, wasted management or other staff resources or claims of any

kind against us from third parties) arising out of or in any way connected with our use of or our inability to use any of the Products or Recommendations (even if CIS has been advised of the possibility of such damages), including without limitation any liability associated with infringement of intellectual property, defects, bugs, errors, omissions, viruses, worms, backdoors, Trojan horses or other harmful items.

Grant of Limited Rights.

CIS hereby grants each user the following rights, but only so long as the user complies with all of the terms of these Agreed Terms of Use:

1. Except to the extent that we may have received additional authorization pursuant to a written agreement with CIS, each user may download, install and use each of the Products on a single computer;

2. Each user may print one or more copies of any Product or any component of a Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, provided that all such copies are printed in full and are kept intact, including without limitation the text of this Agreed Terms of Use in its entirety.

Retention of Intellectual Property Rights; Limitations on Distribution.

The Products are protected by copyright and other intellectual property laws and by international treaties. We acknowledge and agree that we are not acquiring title to any intellectual property rights in the Products and that full title and all ownership rights to the Products will remain the exclusive property of CIS or CIS Parties. CIS reserves all rights not expressly granted to users in the preceding section entitled "Grant of limited rights."

Subject to the paragraph entitled "Special Rules" (which includes a waiver, granted to some classes of CIS Members, of certain limitations in this paragraph), and except as we may have otherwise agreed in a written agreement with CIS, we agree that we will not (i) decompile, disassemble, reverse engineer, or otherwise attempt to derive the source code for any software Product that is not already in the form of source code; (ii) distribute, redistribute, encumber, sell, rent, lease, lend, sublicense, or otherwise transfer or exploit rights to any Product or any component of a Product; (iii) post any Product or any component of a Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device, without regard to whether such mechanism or device is internal or external, (iv) remove or alter trademark, logo, copyright or other proprietary notices, legends, symbols or labels in any Product or any component of a Product; (v) remove these Agreed Terms of Use from, or alter these Agreed Terms of Use as they appear in, any Product or any component of a Product; (vi) use any Product or any component of a Product with any derivative works based directly on a Product or any component of a Product; (vii) use any Product or any component of a Product with other products or applications that are directly and specifically dependent on such Product or any component for any part of their functionality, or (viii) represent or claim a particular level of compliance with a CIS Benchmark, scoring tool or other Product. We will not facilitate or otherwise aid other individuals or entities in any of the activities listed in this paragraph.

We hereby agree to indemnify, defend, and hold CIS and all of its officers, directors, members, contributors, employees, authors, developers, agents, affiliates, licensors, information and service providers, software suppliers, hardware suppliers, and all other persons who aided CIS in the creation, development, or maintenance of the Products or Recommendations ("**CIS Parties**") harmless from and against any and all liability, losses, costs, and expenses (including attorneys' fees and court costs) incurred by CIS or any CIS Party in connection with any claim arising out of any violation by us of the preceding paragraph, including without limitation CIS's right, at our expense, to assume the exclusive defense and control of any matter subject to this indemnification, and in such case, we agree to cooperate with CIS in its defense of such claim. We further agree that all CIS Parties are third-party beneficiaries of our undertakings in these Agreed Terms of Use.

Special Rules.

The distribution of the NSA Security Recommendations is subject to the terms of the NSA Legal Notice and the terms contained in the NSA Security Recommendations themselves (http://nsa2.www.conxion.com/cisco/notice.htm).

CIS has created and will from time to time create, special rules for its members and for other persons and organizations with which CIS has a written contractual relationship. Those special rules will override and supersede these Agreed Terms of Use with respect to the users who are covered by the special rules.

CIS hereby grants each CIS Security Consulting or Software Vendor Member and each CIS Organizational User Member, but only so long as such Member remains in good standing with CIS and complies with all of the terms of these Agreed Terms of Use, the right to distribute the Products and Recommendations within such Member's own organization, whether by manual or electronic means. Each such Member acknowledges and agrees that the foregoing grant is subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Choice of Law; Jurisdiction; Venue

We acknowledge and agree that these Agreed Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland, that any action at law or in equity arising out of or relating to these Agreed Terms of Use shall be filed only in the courts located in the State of Maryland, that we hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action. If any of these Agreed Terms of Use shall be determined to be unlawful, void, or for any reason unenforceable, then such terms shall be deemed severable and shall not affect the validity and enforceability of any remaining provisions.

Terms of Use Agreement Version 2.1 – 02/20/04

Table of Contents

INTRODUCTION

This white paper addresses security concerns that apply generally to Virtual Machine technologies. The recommendations contained within are vendor neutral and should apply to most virtualization deployments. Recommendations are based on a variety of public sources and input from members of the Center for Internet Security (CIS).

Scope & Audience

This document is intended for system administrators, but should be read by anyone involved with or interested in installing and/or configuring Virtual Machines. In the context of this document, a system administrator is defined as someone who can create and manage accounts and groups, understands how operating systems perform access control, understands how to set account policies and user rights, is familiar with how to set up auditing and read audit logs, and can configure other similar system-related functionality.

Configuration Guide for a Secure Virtual Machine Infrastructure

This document focuses on the security aspects of virtual machine technologies and implementations. While these topics cannot be completely separated from the standard security issues of operating a physical computer or configuration of the involved operating systems, this document focuses only on issues unique to virtual machine deployments. We do not cover all of the steps needed to harden the individual operating systems. Other Center for Internet Security documents provide the necessary guidance to secure other aspects of a computing infrastructure.

The first release that will be based on this whitepaper will be the addendum for VMware ESX Server (separate document). It will cover the specific steps needed to apply the general concepts discussed in this document to an installation of VMware ESX Server. The benchmark is meant to be used as a foundation (not a framework) to allow future addenda for other Virtual Machine platforms and vendors to be created using the relevant concepts discussed here.

Virtual Machines

Virtualization refers to one of the following approaches that divides up a physical computer into partly or fully isolated "virtual machines" commonly called "VMs" or "guests". For the operating systems and programs running within these guests, it appears that they are running on their own physical computer. In actuality, they may share the physical hardware of the machine, which may include processor(s), memory, disks, and networking hardware. The use of virtual machines offers two primary benefits: resource sharing and isolation.

In a non-virtual environment, all of the resources on the physical computer are permanently dedicated to the applications running on that computer. If the system has 2 GB of memory but the running applications only need 1 GB, the remainder is either unused or underused.

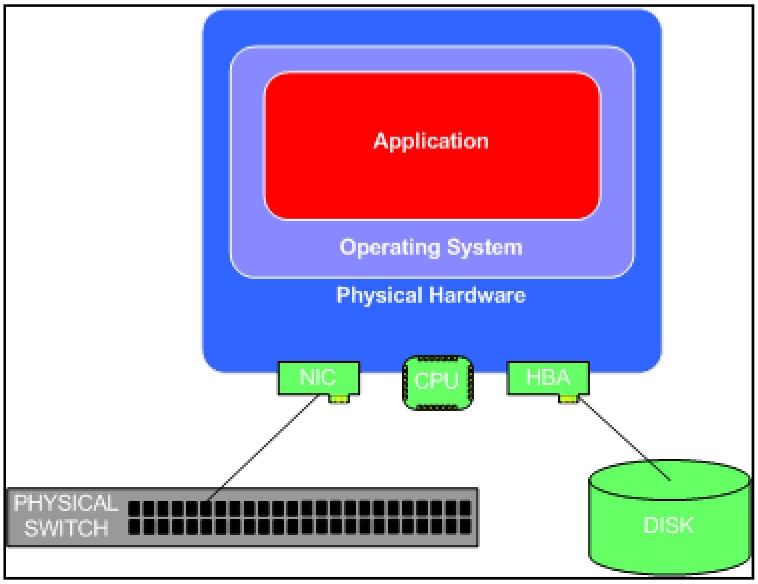

Illustration 1: A non-virtual environment running a single operating system.

On a system with one or more virtual machines, resources including processors, memory, disk, and network devices can be allocated on demand, in some cases without rebooting the virtual machine. Virtual machine environments also provide isolation. In a non-virtual system, all the running programs can see each other, and if given sufficient rights, communicate with each other.

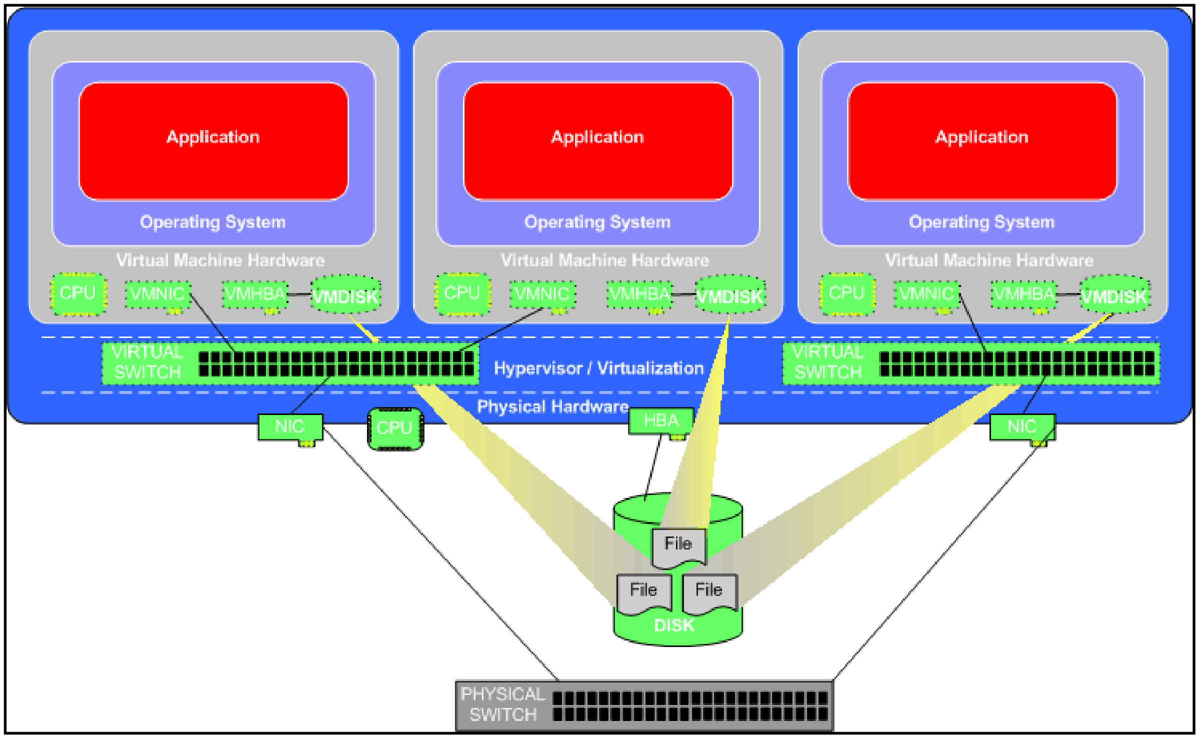

Illustration 2: A virtual environment running three operating systems (guests).

Virtual machines provide what appear to be independent coexisting computers, when in fact the programs running inside these multiple simulated machines are all indirectly running on the physical host. While the extent of the isolation depends on the underlying underlying virtualization technology, as a general rule operations within a virtual machine guest should not be allowed to affect the operation of another guest or the host platform, without specific configuration to permit such interaction. Additionally, this isolation should be strong enough to contain crashes of applications or entire virtual machines to the affected guest without affecting any other component of the virtual environment.

Data Isolation

One of the key issues that will separate virtual computing from physical computing will be the issue of data isolation. The ability of a virtual machine to isolate data from the other guests is a key factor in determining the deployment and implementation in an environment.

John Scott Robin and Cynthia E. Irvine wrote a white paper in 2000 titled, “Analysis of the Intel Pentium’s Ability to Support a Secure Virtual Machine Monitor.” In this paper they concluded the following:

"After defining a strategy to “virtualize” the Pentium architecture, an analysis was conducted to determine whether a Pentium-based secure virtual machine monitor is able to securely isolate classified from unclassified virtual machines could be built. We conclude that current VMM products for the Intel architecture should not be used as a secure virtual machine monitor." - http://www.cs.nps.navy.mil/people/faculty/irvine/publications/2000/VMM-usenix00-0611.pdf

In contrast, recent papers reviewing more current hardware support the ability of modern processors to provide appropriate isolation. It is imperative that the underlying processor be known when making decisions about multi-classification VM guests created on a single host system. A security best practice is to take the position of treating the virtual machine platform with strongest security controls needed to protect the most sensitive data in the guest operating systems, regardless of the hardware architecture (Intel, AMD, Bochs, etc) or virtualization technology.

This document will address data isolation in terms of protecting data of the same classification on a particular host. Protecting dissimilar classifications on a single host may be done using appropriately certified hardware and operating systems, but is outside the scope of this benchmark document.

For further reading see:

- www.intel.com/technology/magazine/computing/intel-virtualization-0405.pdf
- http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/41632A_Virtualization_WP.pdf
- http://www.cs.nps.navy.mil/people/faculty/irvine/publications/2000/VMM-usenix00-0611.pdf
- http://web.mit.edu/Saltzer/www/publications/protection/
- http://research.microsoft.com/~yuqunc/papers/ngscb.pdf

Full Virtualization

In this approach, the VM layer simulates a stand-alone instance of a computer, all the way down to the IO ports, DMA channels, and Interrupts. Because these low-level structures need attention even when nothing is happening in the virtual machine, this approach has some small but steady overhead for each running VM instance.

In true full virtualization all CPU operations are reproduced by the virtual processor. The overhead for handling all CPU instructions, however, makes true full virtualization of x86 processors impractical, if not impossible. Instead, virtual machines that provide a robust enough representation of the underlying hardware to allow guest operating systems to run without modification can be considered to provide “full virtualization”.

Additionally, device I/O in a fully virtualized environment is enabled by emulating common devices within the virtual machine monitor. Interactions with these virtual devices are then translated and provided to the actual physical devices through the host OS driver or hypervisor driver. Some vendors are currently working to build “virtualization aware” devices that provide VM specific functionality and isolation mechanisms.

Paravirtualization

Unlike full-virtualization, paravirtualization requires some modification of the guest OS to operate. Such guests are considered to be “aware” or “enlightened”, due to their knowledge that they are operating in a virtual environment. By blurring the line between the guest OS and the hypervisor, proponents of paravirtualization claim increased performance capabilities through a reduced number of context switches and guest-hypervisor collaboration.

Device interaction in paravirtualized environments also relies on native device drivers operating in a host kernel. Representations to the guest include a split device driver in which each guest is provided with a front-end interface to a back-end generic driver in the host. The host is then responsible for coordination with the native device driver.

Hardware Supported Virtualization

In light of the move towards virtualization as a mainstream technology, both Intel and AMD have released processors with specific hardware support for virtualization. Intel’s product, known as Virtualization Technology (VT), and AMD’s product AMD-V, (also known as “Pacifica”), are beginning to be leveraged by most major virtualization software vendors.

Building on the traditional x86 ring model, hardware supported virtualization creates the ability to establish a trusted “root mode” and an untrusted “non-root mode”, each with their own rings 0-3. In hardware supported virtualization architectures, the hypervisor resides in root mode and all guests are relegated to operation in the non-root mode. Special virtualization instructions, called hypercalls, allow the guest operating systems to call out to the hypervisor for resource allocation, device interaction, or processing requests.

The instruction sets provided by Intel and AMD are not compatible, but provide essentially the same functionality.

System Call Proxy

With this approach, the VM layer intercepts requests (called "System Calls") created by applications heading to the kernel. By modifying the request and/or the kernel's response, the VM layer can present an illusion that the application is running in it's own private space.

Because VM layer work only needs to be done when system calls are made, it has essentially no overhead when the VM is idle, but does have overhead for each system call.

Examples of this include User-Mode Linux and LynxOS.

Threats

While Virtualization is still evolving, so are the threats. History has shown that great features can open unwanted security vulnerabilities. Unfortunately, features are often implemented before the associated security consequences are fully understood. Some of the items labeled as threats here could also be considered fantastic benefits. The goal of listing these items is to raise the awareness of the potential downside, while encouraging people to think through the implementations with respect to security.

Communication Between VMs or Between VMs and Host

Virtual machines can be used to serve multiple needs:

- Sharing a physical computer between multiple organizations or companies
- Using a single physical computer for low security and high security applications
- Consolidation of services onto fewer physical computers
- Providing a common hardware platform to host multiple operating systems

The first three applications assume isolation exists between the virtual machines. In the first case, the companies should be able to assume that applications on the other VMs cannot access theirs. In the second and third cases, a break-in on one virtual machine should not provide access to the others. However, in the fourth case the goal of the virtualization may be to facilitate communication between multiple guests and the host (see section 6.1.2).

Although the majority of security concerns associated with virtual machines are similar, if not identical, to those on physical platforms, VMs do have unique potential weaknesses. Some of these are identified below:

- Technologies like a shared clipboard allow data to be transferred between VMs and the host. This useful functionality can also provide a gateway for transferring data between cooperating malicious programs in VMs of different security realms or to exfiltrate data to/from the host operating system.
- In one VM technology, the operating system kernel that provides the VM layer has the ability to log keystrokes and screen updates passed across virtual terminals in the virtual machine. The

keystrokes and screen updates are logged to files out on the host, allowing monitoring of even encrypted terminal connections inside the VM.

- Some VMs do not focus on isolation at all, giving the guests unfettered access of the host's resources, such as the file system. Such solutions tend to focus on running applications designed for one operating system on another operating system, and eschew the isolation that many VM users expect. VM users with significant security and isolation needs should discuss with their vendor to determine a proper approach toward isolation.

VM Escape

Virtual Machines allow us to share the resources of the host computer and provide isolation between VMs and their host. In an ideal world, a program running inside a virtual machine would not be able to monitor, affect or communicate with another program on the host or another VM. Unfortunately, architectural limitations, the VM vendor's approach to isolation, or bugs in the virtualization software may result in the ability to compromise isolation.

In the worst case, a program running inside a virtual machine would be able to completely bypass the VM layer, getting full access to the host system. The term for this is "VM escape". Because of the host's privileged position, the result is a complete breakdown in the security model of the system. This problem may be compounded significantly by improperly configured host/guest interaction.

VM Monitoring from the Host

It is not generally considered a bug or limitation when one can initiate monitoring, change, or communication with a VM application *from the host*. The host is considered to be in a control position. This is why the host environment needs to be even more strictly secured than the individual VMs it manages.

Depending on the VM technology used, the host can influence the VMs in the following ways:

- Start, stop, pause, and restart VMs.
- Monitor and configure resources available to the VMs, including: CPU, memory, disk, and network usage of VMs.
- Adjust the number of CPUs, amount of memory, amount and number of virtual disks, and number of virtual network interfaces available to a VM.
- Monitor the applications running inside the VM.
- View, copy, and possibly modify, data stored on the VM's virtual disks.

As all network packets coming from or going to a VM pass through the host, the host is generally able to monitor network traffic for its VMs. This is not specific to VM setups, but is included as a general reminder. A similar situation exists in a co-hosting facility where one would also have the ability to monitor network traffic for its hosted machines.

VM Monitoring from Another VM

Because isolation is considered a primary characteristic of VM technology, it is generally considered to be a security flaw when one VM can monitor another without specific configurations to do so. The memory protections built into most modern CPUs can be enforced by a hypervisor that makes use of these memory protection capabilities. The hypervisor is responsible for memory isolation. If implemented properly, the memory protections should prohibit one VM from seeing the memory used by another. Since VMs should not be able to have direct access to the host file systems, VMs should not be able to directly access each other's virtual disks at rest on the host.

For network traffic, there may be an issue with isolation depending on how the network connections are set up with the VMs. If there is a dedicated physical channel for each host-VM link, then guest VMs should not be able to sniff each other's packets.

However, if the VM platform uses a "virtual hub" or "virtual switch" to connect all the VMs with the host, VM guests may be able to simply sniff, or use ARP (Address Resolution Protocol) poisoning to redirect packets to sniff packets going to or from another VM, respectively. Again, this is not peculiar to VM implementations; the exact same approach would work on a network segment shared at a co-hosting facility.

In either situation, authentication of network traffic is considered reasonable mitigation. It may also be possible to enforce limits on what Ethernet MAC address is used on a VM's virtual network interface.

Denial of Service

Because the VMs and the host share CPU, memory, disk, and network resources, virtual machines may be able to cause some form of denial of service attack against another VM.

The best way to prevent this from happening is to limit the resources a VM can access. Many virtualization technologies provide mechanisms to limit the allocation of resources to individual virtual machines. Proper configuration of the host virtualization technology can prevent guests from consuming an excessive amount of resources on the host, thereby preventing a denial of service attack.

External Modification of a VM

The ability to trust a VM in an infrastructure is critical to allow it access in an environment.

For example, we define a User VM that has the ability to access the employee database via a protected application. The user is locked in a captive account to the application. By captive we mean that the user cannot access the database outside of the application.   Hence the VM is trusted in the environment with access to the database. If the VM can be modified, so that the user has access to the system but is not captive, the trust model is broken.

This protection can be afforded by digitally signing the VM and validating the signature prior to execution. The signing key should never be placed anywhere it can be compromised, and special care should be taken to re-sign the VM after any external patches are made. In order to take advantage of this construct the hypervisor would need to be modified to validate the signature.

External Modification of the Hypervisor

While the VM may be able to be made 'self protecting' or 'protected' this does not help with a rogue or badly behaved hypervisor. Hence care should be made to protect the hypervisor from unauthorized modification; or in lieu of that the VM may need to be able to validate the hypervisor.

SOFTWARE INSTALLATION

Most of the installation requirements, steps, and considerations are specific to individual software packages. The notes that follow are generic recommendations; please consult the addendum for your VM platform for more specific details.

Hardware Requirements

Your platform-specific addendum will cover specific resource requirements. In this general document, we offer some generic recommendations for resources.

The physical host computer requires sufficient processing capacity, memory, disk capacity and bandwidth to accommodate all of the applications running in VMs as if they were running on separate systems. Because of relatively small inefficiencies in the sharing process, it is generally necessary to plan for a modest amount of additional processing power and memory to cover the VM platform overhead.

In resource-limited environments, it may be possible to *under*-allocate one or more resources on the logic that individual VMs generally hit resource spikes at different times. For example, if 3 VM's and their applications each needed no more than 500mhz of processor time at peak, one could argue that the physical computer might be able to get away with only a 1 GHz CPU. The problem with this argument is that, while unlikely, it is possible that all of the VMs could find themselves under a denial-of-service attack (through resource starvation), placing all at peak usage simultaneously. Because resource sharing between VMs is somewhat coarse, it is possible that one or more VMs (or guest applications) could be starved for CPU, memory, disk, or network bandwidth. Sizing host systems for VMs is as much and art as a science in the presence of unpredictable workloads, but a this general formula can be used:

$$\text{Hardware Requirements} = H + G^1 + G^2 + G^3 ... + G^N + O$$

where H = Host OS + virtualization software, G = Guest OS + Applications, and O = Overhead.

Although requirements will vary between the various VMs the following initial hardware considerations should be reviewed before deciding on a particular VM vendor:

- CPU – the number of processors, cores, cache, and speed.
- Random-Access Memory – the amount of memory necessary depends on the combination of the amount of memory required by the host operating system, the amount required by each guest operating systems, plus any overhead requirements for each guest.
- Hard Drives – capacity, rotational speed, access times, and buffer size.
- Network interfaces – single or multiple network interfaces as required by VM function.
- Physical ports – some VMs cannot share physical ports between guest operating systems.

Boot Time Disk Requirements

One virtual machine specific resource issue is disk bandwidth at boot time. During normal operation, it may be the case that a single disk provides enough read and write bandwidth to handle multiple running VMs. At boot time, however, that disk would need to provide what is generally a sustained read spike from each of the VMs as libraries, daemons, and other files are loaded off each virtual disk simultaneously.

There are a few ways to address this:

- Stagger boot times so that the most critical VMs are loaded first. As each VM finishes booting, load the next, and so on. If it is not easy to tell when a VM has finished booting, a simple time delay between VM starts (perhaps 2-5 minutes) can approximate this.
- Give each VM its own dedicated physical disk.
- Start all simultaneously, with the understanding that the boot process will take longer on each.

Virtualization Assistance Tools

Virtualization Assistance Tools can offer features such as: performance enhancements, drivers, and data passing capabilities. Some current vulnerabilities exist with regard to data isolation principles and some of these tools, particularly data passed through strings and clipboard features. Refer to the specific Virtual Machine addendum or hardening guide for details.

Other Virtualization Assistance Tools offer can offer a wide range of services between the host and guests. These services can offer time synchronization, host-to-guest monitoring, and other scripts.

Refer to the specific Virtual Machine addendum or hardening guide for details.

Limit Physical Access to Host

Attackers with physical access to the hardware foundations of the virtual infrastructure possess an attack capability equivalent to similar attacks on non-virtual hardware. These include:

- Use OS-specific keystrokes to kill processes, monitor resource usage, or shutdown the machine, commonly without needing a valid login account and password.
- Reboot the machine, booting to external media with known root password.
- File stealing using external media (floppy, CD/DVD-RW, USB/flash drives, etc).
- Capture traffic coming into or out of the network interfaces.
- Remove one or more disks, mounting them in a machine with a known administrator or root password, potentially providing access to the entire contents of the host and guest VMs.
- Simply remove the entire machine.

The risks involved in providing physical access are not different from those in a non-virtual machine environment, but the exposure increases as more virtual machines are hosted on a single accessible computer. For this reason, it may be prudent to lock down the physical hosts more carefully as the value of the guests increase:

- Require card or guard access to the room with the machines.
- Use locks to anchor the machines to the building, and/or lock the cases to prevent removal of the hard drives.
- Remove floppy and CD drives after initial setup.
- In the BIOS, disable booting from any device except the primary hard drive. Also, password protect the BIOS so the boot choice cannot be changed.
- Control all external ports through host and guest system configuration or third party applications.

Harden Base Operating System

The operating system used to host virtual environments needs more scrutiny than the guests because of the host's role as manager for the guest VMs. A compromise on one of the guests should, barring VM platform vulnerabilities, not have any effect on the host or other VMs. A host compromise, however, would give the attacker complete access to all the services and data hosted on all the VMs.

For this reason, the host needs additional security similar to that applied to a security device such as a firewall or intrusion detection system:

- The host should have only as many accounts as needed to manage the VMs. Passwords should be long, hard to guess, changed frequently, and only provided to staff that must have access.
- The host should ideally have no network accessible services whatsoever. Any services that need to remain running should require authentication before any actions can be taken, and should be firewalled on the host itself, or at a local firewall, to restrict access to the management machines that must have access.
- Unneeded programs and services should be disabled or removed entirely. This provides a security and resource benefit, as disabled daemons free up memory.
- The host should be patched regularly. Because an errant patch on the host OS could potentially disable multiple guest VMs, patches destined for the host should be tested on a non-production test machine before being applied to production systems.

Configuration Maximums

Most VM platforms allow the administrator to set limits for each VM for one or more of processors, memory, disk space, and virtual network interfaces. When initially configuring your VMs, set appropriate limits for each of these so that no one VM can monopolize the resources on the system.

If two or more VMs share a processor or disk (or even a physical network card in extreme cases), there may be some increased latency in CPU or disk access. While most services will gracefully handle a slight increase in latency, some time-sensitive applications may not. Examples include burning a CD or DVD, multi-track audio capture or video capture or playback, and time-sensitive direct access to hardware devices on the host. Activities like these are not generally good candidates for operation inside a VM.

Firewalling Virtual Machine Layer Service Ports

In addition to the ports normally open inside an operating system for its servers and clients, the virtual machine layer may open its own ports (using the IP address(es) of the host operating system). These ports may allow others to connect remotely to the virtual machine layer to view or configure virtual machines, share drives, or perform other tasks.

Access to these ports should be strictly limited to the machines that have authorization to manage the virtual infrastructure. At a minimum,a firewall in the host operating system or a separate firewall protecting the host should block access to those ports by default, only allowing access to those ports to a small number of management machines.

Ideally, no remote access would be permitted to either the host or hypervisor. This configuration provides insulation of the core trusted processes from the potentially compromised environment. Realistically, however, enterprise-level deployments will mandate some remote management capability. Wherever practical, the management network communications should take place on an independent network interface (admin interface) on the host, used only for this purpose. Additionally, if a separate physical administrative infrastructure (admin network) can be created, this would further serve to benefit the protections of the host. Optimally, only this physically separated administrative infrastructure would be used for management functions, such as creating new VMs or changing existing images.

Use Encryption For Communication

The use of encryption for secure communications should be used whenever possible. The use of HTTPS, TLS, SSH or encrypted VPNs from guest to host or from management devices to hosts should be employed.

Encrypted links provide not only encryption to hide the requests and responses between the management machines and hosts, but also provide packet authentication to prevent spoofed source address attacks, connection hijacking, route hijacking, and man-in-the-middle attacks.

LOGICAL ACCESS CONTROLS

Virtualization Server Authentication

Virtualization servers will need to be configured in a similar manner to their physical server counterparts. Issues such as, encryption / masking of stored passwords, lockouts (timeouts) for successive failed login attempts, should be considered. Refer to the specific Virtual Machine addendum or hardening guide for details.

Disabling Features (Including Screensavers and Suspend)

In a single-operating system computer, there are a number of background or low-priority tasks that run during off hours or when the system is otherwise idle. While most properly resourced host environments will not be affected by these problems, limited resource hosts may. These include:

- Screensavers
- Defragmenters
- Search tools, such as those that index file names or contents for the entire drive.
- Virus and malware scanners
- File integrity checkers
- Log rotation and analysis tools
- System update, or "patching" tools

There are a few problems with running these in virtual machines.

The first is idle detection. In a non-virtual environment, a program may choose to monitor keyboard activity, network activity, disk load, and/or processor load in order to decide when to run. In a virtual machine setup, even a program attempting to run when the system is idle won't have knowledge of the state of the other VMs needed to make an accurate decision about whether the entire physical system is truly idle. A processor-intensive screensaver, even a well-intentioned one, may very well monopolize a processor when it's needed by another VM.

An application that attempts to give priority to other running applications on the system may instruct the operating system kernel to lower it's processor, disk, or network priority. While this does indeed give higher priority to other applications inside that virtual machine, it does not affect relative priorities between virtual machines; the application that thought it was lowering it's priority uses just as much CPU time as if it had not tried at all.

The final problem is scheduling. Operating systems and applications tend to schedule their low-priority jobs at a fixed time in the middle of the night or on weekends. The problem is that with similar operating systems in the virtual machines, all of the VMs will start their low priority jobs at the same time. This kicks off a rush for resources that can drastically affect any normal-priority tasks on the system.

There are a few answers to this issue:

- Disable any low priority jobs that are not needed. Screensavers, in particular, rarely serve a legitimate use in a virtual machine, at least in respect to preventing monitor burn-in. If a screensaver is used to lock the desktop after a period of inactivity, it should be chosen to be one that uses little processor time.

- Stagger the start time for any low-priority tasks that still need to run. As an example, on systems with the bash command shell, the command:

  ```
  sleep $[ $RANDOM / 32 ]
  ```

  placed at the beginning of the list of low-priority tasks to be run will pause for a random number of seconds between 0 and 1023, a pause of between 0 minutes and 17 minutes. While there will still be some contention, that will be reduced. Other platforms may have similar approaches for sleeping for a random amount of time.

If a random sleep is not available, change the start times for these jobs in each virtual machine so the tasks have minimal chance of overlap.

File Sharing Between Host and Guests

Many virtual machine environments support file sharing between host and guests, allowing for a convenient way to link the two systems together and move files between them in a relatively seamless fashion. However, such file sharing introduces security risk. At a minimum, a compromised guest could access the host file system and alter those directories that are explicitly configured for sharing with a guest. But, even if no directories have been shared in the host, the mere activation of file sharing introduces extra functionality into the virtual machine environment. This functionality is attackable and may have security flaws that could lead to the exploitation of the host machine or hypervisor. Therefore, unless there is a business need that explicitly requires VM file sharing, this functionality should be disabled.

Time Synchronization

In a non-virtual environment, the clock circuitry in PCs is notoriously inaccurate, gaining or losing seconds per day. Operating system, BIOS, power management, and other choices can exacerbate the problem, causing significant clock drift.

In a virtual environment, the problem can become even worse. The timer ticks that the virtual machines depend on for their timekeeping may be delayed, delivered in a batch, or missed completely. The combined virtual machine clock drift and normal clock drift can result in tasks running significantly early or late, logs that no longer record accurate times for forensic use, and security issues such as time-of-day login restrictions that don't function correctly.

There are two primary approaches to clock synchronization in a virtual machine environment. First, if the virtual machine layer allows the guest operating system to synchronize its time directly (as opposed to using the above timer ticks) from the host, this is the simplest way and should be used.

If the virtual machine layer does not have this ability, the NTP daemon (Unix-like operating systems) or service (Windows) should be enabled and configured to synchronize with an existing time server, preferably one that is on or close to your network.

In either case, use NTP because it is critical that all system clocks are synchronized (time stamps, logging, scheduling, forensics, etc). At a minimum, NTP should be configured and run on the host operating system to mitigate these issues. Guest VMs should use the same NTP server as the host, or may use the host itself as an NTP server, if it is being appropriately synced.

Use Hardening Guide For Guest OSes

Use a hardening guide such as one from the Center for Internet Security, SANS Institute, Defense Information Systems Agency (DISA), National Security Agency (NSA), or vendor supplied to configure guest operating systems.

Disconnect Unused Devices

The virtualization layer allows individual virtual machines to directly or indirectly control physical devices out on the host such as floppy and CD-ROM drives. This capability can be configured for each virtual machine and can usually be changed for a VM while it is running.

You're encouraged to disable this connection to host devices for all VMs by default:

- If more than one VM requests access to a host device during boot, the remaining VMs may block until the first has released the device, needlessly delaying the boot process.
- Media inserted in the drive may contain malware or unwanted code; if the operating system inside the VM automatically mounts and executes code from the disk (the "Autorun" or similar feature), this may be loaded into the VM.

A better approach is to enable host access to devices only when needed.

Remote Management Approaches

Unlike desktop and laptop systems which can reasonably be expected to have a dedicated monitor and keyboard, Virtual Machines are similar to server class or rack-mount systems, where a dedicated, easily-accessible monitor and keyboard are not likely to be available. Depending on how the VM layer operates, the host and VMs may share the sole connected monitor.

For this reason, the system administrator needs to consider alternate management approaches. The addenda to this document will go into more detail about specific approaches that cover specific VM technologies, but we'll cover some generic approaches that should apply to most VM approaches.

Each of the following approaches requires a functional network interface inside the virtual machine and the use of some kind of software service or daemon running inside the VM.

Because these services are network accessible, they need to be secured just as much inside VMs as on regular servers:

- Access should be limited to a small set of IP addresses of authorized management systems.
- Access should require a username and sufficiently strong password, with a supporting password policy. In environments with stronger security requirements, two-factor authentication, public-private keypairs, and/or one-time passwords may need to be used.
- Communication to the management tools should be both encrypted and authenticated. If the tool in question does not have built-in encryption and authentication, it may be that the communication channel can be routed through some form of VPN or SSH tunnel.

SSH

This cross-platform terminal program allows one to connect from a desktop or laptop machine to a VM. This terminal connection allows one to run both text-based and graphical applications remotely; the screen updates for both show up on the original desktop or laptop monitor. SSH uses the X-Windows protocol to carry graphical applications, so both the client and server need some X-Windows support (included in some operating systems, can be added to others).

Because the SSH client and server can be running different operating systems, this ends up being a relatively flexible and secure approach to running whatever native management tools the administrator would already have used in a non-virtual environment.

SSH (to include: scp, sftp or any other application based on SSH) should configure SSH to disable the version 1 protocol. Also, disable SSH login for root or administrator and force users to login with individual user accounts and then su to root or administrator or utilize some other form of role based access control.

For additional security or for accounts with elevated privileges, implement two-factor authentication (ex. DSA/RSA) to help prevent “man-in-the-middle” attacks. Do not forget to enable a login banner for law enforcement purposes.

Virtual Network Computing (VNC)

VNC, Virtual Network Computing (the name doesn't have any direct connection to the term "virtual machine"), is a software approach that provides a remote desktop. Applications running inside the Virtual Machine send their screen updates to the VNC desktop. An administrator can connect up to a VM VNC desktop, run some applications, and then disconnect, possibly even leaving some applications running in the background. He/she can later reconnect to continue working with those tools.

Like SSH, VNC provides a cross-platform approach to running tools native to the VM operating environment. However, unlike SSH, VNC does not necessarily provide encrypted communications (as recommended in section 4.2).

Encryption for VNC solutions may be provided by vendor configuration settings, vendor plugins, or third party plugins. Exposing VNC connections, whether encrypted or not, to the Internet or other external networks is not recommended. External connections should be avoided or controlled through additional protections such as a Virtual Private Network (the name doesn't have any direct connection to the term "virtual machine") connection or SSH tunneling.

Some virtualization vendors provide VNC support as a function of the hypervisor. The associated infrastructure used to provide this information to users may introduce its own security or performance concerns. These must be evaluated on a case-by-case basis.

Web Management

Some management tools offer multiple operating modes. In addition to a native graphical application, the tool may provide an HTTP-based management interface for managing the operation or configuration of a system or service. Once a network interface is in place, these interfaces provide an easy approach to remote management. As mentioned previously (in section 4.2), encrypted communications should be used. For vendors that do not provide the capabilities of utilizing these remote management services across an HTTPS connection, VPN or SSH tunneling should be implemented.

Patching and Vulnerabilities

Creating and maintaining virtual environments adds a new complexity to patch management. Now, not only do you have additional **host** machines to worry about, you now have the added complexity of **guest** operating systems being turned on, moved and shutdown with little or no notice. This creates a challenge not only in verifying the appropriate patch levels but also with maintenance windows. I.E. Patching the **host** may mean shutting down multiple **guests**. It is critical that the **guest** and **host** operating systems maintain the latest security patches to avoid exposing your environment to unnecessary risk of not just the individual target but the entire virtual environment.

Subscribing to vendor or project mailing lists that focus on security can be a great method of keeping systems updated. Administrators must be careful to use patches and updates that have been modified specifically for their products because in many cases patches must be modified by the virtualization vendor or project.

Of course, patches should be tested before applying to a production environment. Since updates and patches can force systems or services to be restarted, a strategy should be developed to minimize disruption to virtual machines.

Auditing

Due to the dynamic nature of the virtual guest operating systems, a centralized logging server (either on the host or on a separate machine) can act as a tool to aid a system administrator in determining if guests have gone offline. As discussed previously, offline guests are more susceptible to becoming out of sync with patches, updates and signatures.

Good virtual machine events to log can include: power status (on, off, suspend, resume), changing a hardware configuration, and login attempts for accounts with elevated privileges. Additionally, the host should log changes to the virtual machines to include: copying, moving, or deleting from the host.

Host and Network Defenses

The host operating system, generally, has the ability to route and inspect all network traffic traversing its interfaces to and from the guests OSes. This situation creates an excellent choke point for running network based protections such as firewalls and intrusion monitoring systems. These solutions can be utilized to augment host-based firewalls and intrusion detection software and provide additional levels to the defense-in-depth strategy.

The advantage to running network defenses through virtual machines on the host OS is that these defenses cannot be directly affected by malicious softwares or intrusions on any of the guest OS systems barring the threat of “VM Escape” described above (in section 2.3.2). This advantage, however, does come with the cost of resource and bandwidth consumption which should also be taken into consideration.

File Integrity Checking

File integrity checking can be a valuable aid to administrators in verifying that unauthorized modifications to files have not occurred. Although file integrity checking for a guest OS depends on the level of risk associated with its function, file integrity of the host OS is recommended. The files associated with VM configuration should be taken into consideration when addressing file integrity checking on the host OS. Storing the hashes “offline” (on a CD-ROM or other read-only media, for example) should be done to ensure that a malicious user has not re-hashed the modified files.

Strong Passwords

Strong passwords should be used for both hosts and guests. Refer to the specific Virtual Machine addendum or hardening guide for details.

GRUB, BIOS Passwords

Passwords should be used for BIOS and boot loaders for both hosts and guests. Refer to the specific Virtual Machine addendum or hardening guide for details.

Disk Partitioning

Disk partitioning on the host can prevent inadvertent denials of service from guests filling up their resources, and can allow role-based access controls to be placed individually on each virtual machine guest partition (or the partition that holds all of the virtual machines).

Warning Banners

Warning banners should be used for both hosts and guests. Refer to the specific Virtual Machine addendum or hardening guide for details.

Backups

Image backups are recommended for all virtual machines. This fosters methods of error recovery that include:

- recovering a server entire from a catastrophic event using the image
- restoring individual files simply by mounting the backup image

However, care should be taken with the security of the backup data stream as well as security of the final backup image on disk and/or tape. If possible, the data stream of the backup should be encrypted to prevent the "theft" of a server image by capturing the packets in the backup. More importantly in most cases, the data at rest should have appropriate ACLs to restrict copying or mounting images to authorized support personnel. In higher security situations, network level protections such as VLANs and ACLs should also be implemented to protect both the data in transit and data at rest. In extreme circumstances, encryption of disk directories or partitions as well of tapes will be implemented.

These measures protect both the server image entire, as well as the data in the files that comprise the image. As with physical servers, security of tapes or other backup media is also key. Tapes can be protected using physical security as well as tape encryption, depending on requirements. The backup process itself should use a dedicated backup account, with appropriate restrictions (no shell for instance). Under no circumstances should the root account be used for backups. In instances where disk and tape encryption is utilized for backup and recovery, additional consideration must be made for the escrow of the keys associated with the encryption.

References

- "Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor.", Robin and Irvine, 2000, http://www.cs.nps.navy.mil/people/faculty/irvine/publications/2000/VMM-usenix00-0611.pdf

- "The Protection of Information in Computer Systems", Saltzer and Schroeder, http://web.mit.edu/Saltzer/www/publications/protection/

- "NGSCB: A Trusted Open System", Peinado, Chen, England, Manferdelli (Microsoft Corporation), http://research.microsoft.com/~yuqunc/papers/ngscb.pdf

- Intel - Virtualization Technology (VT): http://www.intel.com/technology/platform-technology/virtualization/index.htm

- AMD – AMD-V or "Pacifica": http://www.amd.com/us-en/Processors/ProductInformation/0,,30_118_8826_14287,00.html

- User-Mode Linux: http://user-mode-linux.sourceforge.net/

- LynxOS: http://www.lynuxworks.com/products/whitepapers/compatibility.php3

- "An Introduction to Virtualization", Singh http://www.kernelthread.com/publications/virtualization/

- VMware: http://www.vmware.com

- Xen: http://www.xensource.com

- Virtual Server: http://www.microsoft.com/windowsserversystem/virtualserver

- "Security Design of the VMWare Infrastructure 3 Architecture": http://www.vmware.com/vmtn/resources/727

- "VMWare Infrastructure 3 Security Hardening": http://www.vmware.com/vmtn/resources/726

- Debunking Blue Pill myth: http://www.virtualization.info/2006/08/debunking-blue-pill-myth.html
- "Blue Pill" is quasi-illiterate gibberish": http://x86vmm.blogspot.com/2006/08/blue-pill-is-quasi-illiterate.html

- “A Comparison of Software and Hardware Techniques for x86 Virtualization”, Adams & Agesen: http://www.vmware.com/pdf/asplos235_adams.pdf

- “The Promise of Virtualization: Data Center Technologies for Today and Tomorrow”, Novell:
  http://www.novell.com/collateral/4622040/4622040.pdf

- “Security Management in a VMware Virtual Infrastructure Environment”, Arrasjid & Mills: http://download3.vmware.com/vmworld/2005/sln138.pdf

- VMware Security Response Policy, http://www.vmware.com/support/policies/security_response.html

- “Configuration Maximums for VMware Infrastructure 3”, http://www.vmware.com/pdf/vi3_301_201_config_max.pdf

- “Virtualization (& paravirtualization), [x86] Background, Risks, Controls, Audit Steps”, Hoesing: http://members.cox.net/m-d-hoesing/CACS_Virtualization_V2.ppt

- DRAFT Virtual Computing STIG Version 1, Release 0: http://iase.disa.mil/stigs/draft-stigs/Virtual-Computing-STIG-V1R01.doc

- “Securing Virtualized Infrastructure: From Static Security to Virtual Shields”, 2007, Antonopoulos: http://hackreport.net/wp-content/uploads/2007/03/nemertes-issue-paper-securing-virtualized-infrastructure.pdf

- “Virtual Machines as a Special Class of Operating Systems”: http://www.softpanorama.org/VM/index.shtml

Credits

John Banghart, CIS – Project Lead

Dave Shackleford, CIS – Project Lead

Chris Farrow, Configuresoft – Project Initiator

Joel Kirch, WBB Consulting – Technical Lead & Primary Author

Bill Stearns, Intelguardians – Technical Co-Lead

Special thanks to all of the active CIS Virtual Machine Mailing list participants:

Michael Angelo
Charu Chaubal
Michael Hoesing
Kirk Larsen
Steven Nelson
Dennis Moreau
Iben Rodriguez
Greg Shipley
Jared Skinner
Ed Skoudis
Doug Staz
Rob VandenBrink
Brian Waite
Don Webber
Joe Wulf
and many more.

This work was supported in part by the Department of Homeland Security (DHS) Science and Technology (S&T) Directorate and the Air Force Research Laboratory (AFRL) under contract FA8750-05-C. Points of view expressed in this document are those of the authors, and do not represent the official position of the DHS S&T Directorate or the Air Force Research Laboratory.