ANSWERING SECURITY QUESTIONS. QUESTIONING SECURITY'S ANSWERS

SAMPLE ISSUE
VIRTUALIZATION

COMPUTER SECURITY ALERT

Computer Security Institute, 600 Harrison Street, San Francisco, California 94107, tel: 415-947-6135

Virtualization:

Security enabler or security threat? Has anyone stopped to ask?

A virtual infrastructure might enable better security, but it definitely demands it.

Virtualization creates as many security management issues as it solves. Plus, paradoxically, while virtualization's greatest security benefit is how it enables resource isolation (e.g. putting each egg in its own basket), virtualization's greatest security risk is how it enables resource consolidation (e.g. putting too many eggs in one basket).

Virtualization may be the buzzingest buzzword of the moment, which makes it exceedingly difficult to decipher the truth through the din. Virtualization technology comes in many forms and suffers from a glut of redundant terms. For the purposes of this article, we'll focus on what are sometimes called "system" virtual machines or "hardware" virtual machines or hitherto just VMs—logical entities that behave just like physical entities (be they servers or workstations), executing complete operating systems.

The virtualization infrastructure

Atop the server/workstation hardware runs the virtualization software product. The product of choice may include a variety of applications, but the heart of the product—the software providing the virtualization layer—is the VMM (which usually stands for "virtual machine monitor," but is occasionally translated to "virtual machine manager"), and the heart of the VMM is the hypervisor (some of which are open-source, like Xen, for example). Some hardware—most notably, AMD-V and Intel VT chips—contains support for virtual machines, but virtualization software can still run atop hardware without this component. (Note one linguistic peccadillo: sometimes the term "hardware virtual machine" or "HVM" is used in the manner described above, in the previous paragraph. Other times, however, this term is used to describe the virtual machine-enabled hardware itself, and not the VM running on top of it.)

Virtualization software comes in two flavors: "hosted" or "type 2" VMMs must run atop a typical operating system; "bare metal" (also called "type 1" or "native") hypervisors can run directly on the hardware, without a standard OS as an intermediary, because bare metal hypervisors are effectively operating systems within themselves. Generally speaking (though there are exceptions), bare metal hypervisors are used for server virtualization and hosted VMMs are used for desktop virtualization.

In either set-up, the hypervisor/VMM can run multiple VMs, each of which has the capability to run its own independent operating system.

In testing and development labs, virtualized workstations are popular. For example, instead of needing multiple physical machines, one MacBook running the Mac OSX operating system could run the Parallels Desktop hosted VMM. Parallels, in turn could run one VM operating on Mac OSX, another VM on Linux, another VM on Windows Vista, another VM on Windows XP with SP2 installed and another operating XP without SP2.

However, virtualization got its start in the data center, as an enticing method of server consolidation. Instead of investing in a hundred hardware servers, an organization might instead buy only one physical server, load a bare metal server hypervisor like VMware ESX onto it, and run a hundred virtual servers.

Consolidation vs. sprawl

Let's get one thing straight: virtualization is *not* a security tool, and no one in the industry is really trying to pass it off as such. Virtualization will not automatically create a more secure infrastructure.

As Alan Shimel at security vendor StillSecure said, "mixing virtualization and security is not exactly like mixing chocolate and peanut butter."

In fact, to believe otherwise goes against conventional security logic. *Hardware is better than software,* but virtualization swaps out hardware for more software. *Simplicity is better than complexity,* but virtualization adds a layer of complexity.

However, if intelligently implemented, with security in mind, virtualization can give your organization's security program a leg up. On this, all but one of the virtualization vendors, virtualization security vendors and security researchers I spoke to agree (though their degree of faith varies widely). The exception is David Lynch of virtualization lifecycle management vendor Embotics. Lynch further posits that some of virtualization's security woes stem

from the fact that when the technology was created it was only intended to be an "operations tool for a fast ROI," but "it has since become a data center architecture."

However, it's precisely this use in the data center that makes virtualization a security-enabler, say some, who argue that virtualization makes solid security more economically feasible. Some say that properly locking down a data center containing 1,000 hardware servers is far pricier than locking down a data center with 50 hardware servers running 20 VMs apiece. So while the first data center security manager may have to cut corners, the virtualized data center security manager can afford to spare no expense on solid security.

"To a large number of customers, virtualization is what they have been looking for, in terms of providing a very cost-effective platform on which to lay additional applications for security features and functions," said David DeValk from virtual security appliance vendor Reflex Security. "I definitely look at it more as a security-enabler."

Others—including many CSI members—quite vehemently argue the oppoiste. They say that although a virtualized data center saves money on the hardware itself and on the the cost of maintaining an uninterruptible power supply (UPS), it doesn't save money on security. Security software must still be applied to each VM, input/output from each VM must still be monitored, etc. Further, virtualization poses unique security risks may require a larger investment in security—more on this below. Obviously there's some cost–benefit analysis necessary here. (See page 15 to view the results of a survey of CSI members' opinions on virtualization, and highlights from a members-only conference call on virtualization.)

VMware Not Threatened by Hyper-V

As part of Server 2008, Microsoft will include its own virtualization product: Hyper-V, still in beta. When asked if VMware—the indisputable market leader in server virtualization products—thought Hyper-V would take business away from VMware, a representative said, "We see Hyper-V being used by Windows customers as a way to get introduced to virtualization." The representative pointed out that Hyper-V lacks the ability to do live migration and has a far greater footprint—4 GB compared to VMWare's 32 MB for ESXi.

Another benefit of virtualization is that, should one VM become infected or corrupted in some way, it is simpler to roll back to a "golden image" or a "known good" state. So it is useful to business continuity efforts and makes it potentially quicker and easier to recover from infections.

The trouble is, while a server VM behaves like a real server and while a desktop VM behaves like a real desktop, VMs can be

moved and copied with the same ease and speed you move or copy a file folder. It makes things too easy.

Avoiding the obvious theft-related risks for the moment, examine some of the subtler headaches this convenience may cause. For example, I travel quite a bit for business, and I have foolishly left something behind in every single hotel room in which I've spent a night. I tremble at the thought that one day my laptop could be the thing I forget. It would certainly be convenient if I could quickly make a few copies of my entire hard drive before I leave for a trip (one at the office, one in my suitcase, perhaps. It'd also make life easier if I could quickly make a few copies of my department's corporate server—so that if the hotel Internet access is too expensive or if my VPN is giving me trouble, I can still remotely access the important files I need from the server.

When one VM can so suddenly become three or four, the risk of VM sprawl is very real, and can defeat the original purpose of hardware consolidation.

When one VM can so suddenly become three or four, the risk of VM sprawl is very real, and can defeat the original purpose of hardware consolidation.

Further, if you cannot properly inventory all of your VMs, then you cannot properly scan them, patch them, secure them, audit them, quarantine them or destroy them. So while virtualization can make it easier to recover from a viral infection by reintroducing a known-good instance of the VM, it can also make it easier to become reinfected by reintroducing an infected VM that was lost in the shuffle. In their paper "When Virtual is Harder Than Real: Security Challenges in Virtual Machine-Based Computing Environments," Stanford University's Tal Garfinkel and Mendel Rosenblum wrote, "New and potentially vulnerable VMs are created on an ongoing basis, due to copying, sharing, etc. As a result, worm infections tend to persist at a low level indefinitely, periodically flaring up again when conditions are right."

Of course we've until now ignored perhaps the most obvious concern. Already we worry about careless end users losing portable data storage devices and sinister bad guys slurping sensitive data onto an iPod or a USB stick and walking out the door with it. However, when an entire server (or even multiple servers) can be slurped up with the same ease and speed of a folder of Excel documents, the need for strong port/device control mechanisms and other endpoint security measures becomes imperative.

What might have been considered merely "advisable best practices" in a physical environment become "essential" in a virtualized environment.

Eggs in baskets

Perhaps the greatest way of using virtualization to create a more secure environment is by using it to isolate resources.

For example, clearly the human resources department at your organization requires both access to a database containing employee PII and access to a Web-based recruitment service. Equally clear is that the confidential database would be more secure if it were never exposed to Internet-borne malware. So perhaps the HR department uses one VM for Web browsing and another VM—one which is never connected to the Internet, nor even has a Web browser—that is devoted exclusively to the task of running the PII database software.

"If you leave everything the same and then add a hypervisor, then yes, virtualization would have an additive effect on the attack surface," said Kurt Roemer, security strategist at Citrix. "But the power of virtualization is really being able to go through and compartmentalize. You can reduce the attack surface significantly, but to do so you do need to rearchitect."

Granular isolation of resources and access to those resources can be a very effective part of a security program. The thing is, everything hinges on the strength of the isolation, and not everyone trusts virtualization technology's ability to hold those isolation boundaries.

"I think that virtualization, by itself, actually weakens isolation boundaries," said Jon Oberheide, the University of Michigan researcher who presented a talk about exploiting VM migration at the BlackHat Federal conference in February. "Instead of having a hardware platform protecting your machine state, you now have a software layer, which is more vulnerable to attacks. As we see more software involved in enforcing isolation barriers, we see more vulnerabilities like the one discovered by Core Labs."

The vulnerability to which Oberheide referred was a path traversal vulnerability in the shared folders implementation of VMware's desktop (not server, so not ESX) virtualization products—VMware Workstation, VMware Player and VMware ACE. Core Security Technologies' CoreLabs notified VMware of the vulnerability in October, and publicly disclosed the vulnerability Feb. 25. From the CoreLabs advisory:

> VMware's shared folders allow users to transfer data between a virtualized system (guest) and the non-virtualized host system that contains it. This form of data transfer is available

> to users of the Guest system through read-and-write access to file system folders shared by both Guest and Host systems. To maintain effective isolation between Guest and Host systems, these mechanisms should limit access from the Guest only to the Host system's folders that are selected for sharing with the virtualized guests.
>
> A vulnerability was found in VMware's shared folders mechanism that grants users of a Guest system read-and-write access to any portion of the Host's file system, including the system folder and other security-sensitive files. Exploitation of this vulnerability allows attackers to break out of an isolated Guest system to compromise the underlying Host system that controls it.

Ivan Arce, chief technology officer of Core Security Technologies, described the vulnerability as being trivial to exploit. It is not, however, remotely exploitable.

So, in the theoretical example above, an attacker's chances of successfully using a Web-based attack on the HR department's Web-browsing VM to obtain access to that sensitive database located on another VM are much slimmer than his chances would be if he were attacking a typical non-virtualized environment. The attacker would first have to find a way to break out of the compromised VM, access the hypervisor and then access the offline VM containing the database. Normally one would consider this a tough job, but a vulnerability like the one disclosed by Core would make it quite easy.

Security professionals are already nauseated at the thought of an attacker gaining root access to one machine. Yet an attacker could gain access to potentially a hundred times more data and resources if he could break free of a VM and get access to the hypervisor, thereby earning root access to all the VMs being run by that hypervisor.

The stakes are far higher.

On the other hand, there are benefits to having fewer, smaller points of access—in other words, a smaller attack surface—because not only is it cheaper, but it's easier to properly secure one or a few small points than it is to properly secure many. So, the security of any virtual infrastructure relies heavily upon having a small, secured hypervisor.

It's possible here to get stuck in some semantics. The hypervisor isn't really the attacker's end target, and the attacker probably won't go after the hypervisor directly. Most likely the ne'er-do-well would actually launch the attack against a VM. The first goal of the attack would be to break free of the VM and access the hypervisor. The hypervisor itself, however, is really just a small piece of

code that creates the virtualization layer. Its value is not in itself, but in the access it provides to all the other VMs. It's these other VMs that are the end targets.

"So," said DeValk, "the more that you can do to rip out all of the attack surface in the VM, the more you can reduce the chances that an attacker can use the VM to launch an attack against a hypervisor."

The hypervisor itself, however, is really just a small piece of code that creates the virtualization layer. Its value is not in itself, but in the access it provides to all the other VMs. It's these other VMs that are the end targets.

This is an important distinction to make, because it drives home the point that the hypervisor is not the only important piece of the virtualization infrastructure. Nonetheless, the security of the hypervisor is paramount, particularly if it's to be an effective method of isolation.

"If we could indeed effectively 'rip out all of the attack surface in the VM' (in this case a standard OS like Windows), then we would not need a virtualization layer at all," said Joanna Rutkowska, founder of Invisible Things Lab and creator of the Blue Pill. "The problem we have is that the industry has kept failing at providing security for general purpose OSes for years, and this is why we would like to now use a virtualization layer to provide isolation. And for this to work, the hypervisor should be immune to attacks, even if one or more of its VMs get compromised. Otherwise it doesn't really make much sense."

So how to make the hypervisor secure? First of all, the code must be as free of vulnerabilities as possible. Thus it should be as small a piece of code as possible, because the fewer lines of code, the fewer opportunities to introduce a vulnerability and the more likely that any vulnerabilities will be found. The pursuit of the smallest-possible hypervisor is the subject of much research.

"We work very closely with the major hypervisor providers," said DeValk, "and I can say they spend a tremendous amount of time not only making the code as small as possible, but limiting the functionality of what the code can do to the bare, bare minimum."

Next, layer the security architecture atop the virtualization layer. Sometimes this is discussed as "adding security to the hypervisor," but this is not exactly the case. Adding security tech into the hypervisor itself would increase the size of the hypervisor, not shrink it. So instead, separate

security products are laid atop or wrapped around the hypervisor—the virtual infrastructure requires the same tools of the trade you use in traditional non-virtualized environments.

The trouble is, although traditional security products will run, unchanged, in a virtual environment, they are not optimized for the virtual environment. (Further, according to Oberheide, the existing security devices, such as network intrusion detection systems, do not have visibility into the interactions between VMs.)

To this end, VMware recently announced VMsafe. VMsafe is not a security tool itself, but rather a set of APIs written to help security vendors create products optimized for the virtual environment.

VMware's Nand Mulchandani explains that if you're a hypervisor, "you're literally sitting under the OS and managing the hardware and the OS together. You have information on everything the CPU is doing, and you can touch and manipulate it. What we're doing with VMsafe—which is very much a first step—is taking this very rich information and handing it up to the security technologies."

A different tack on securing the virtual environment is to cut one's potential losses simply by not putting all your virtual eggs in the basket of one hypervisor. This is where the idea of nested hypervisors comes in. By having more hypervisors—one, inside another, inside another—it is true that you have more points of access, more complexity, and thus a wider attack surface. However, if an attacker gets access to one hypervisor, he only gets access to a comparatively small group of VMs located on that piece of hardware, as opposed to all the VMs located on that piece of hardware.

Plus it furthers the pursuit of a tiny hypervisor.

"You really want to have the smallest trusted computing space possible," said Oberheide. "In nested hypervisors, the first layer is a very small hypervisor. You slowly build layers of functionality upon that hypervisor, but you keep your smallest amount of code running on the hardware directly. You gain a lot of security that way."

There are, however, ways in which nested virtualization could be abused for malicious purposes. More on this below.

Blue Pill

Of course no discussion on virtualization security would be complete without mentioning the much-discussed Blue Pill and its creator, Joanna Rutkowska. The BlackHat Briefings conference in August 2006 was abuzz about Rutkowska's presentation in which she discussed Blue Pill, a

nearly undetectable rootkit. Note well: Blue Pill does *not* exploit vulnerabilities within virtualization technology. It is proof-of-concept code that shows how to *abuse* legitimate functionality within virtualization technology.

"Blue Pill," as Rutkowska put it to me, "just demonstrates how the hardware virtualization technology, like AMD's SVM and Intel's VTx (yes, Blue Pill runs on both architectures) can be abused to create malware. Sort of like we could abuse a kitchen knife in order to kill somebody."

In the September 2006 *Alert,* Robert Richardson discussed Rutkowska's BlackHat presentation (for which she focused on AMD's SVM technology) in greater depth. From his story:

> The idea behind SVM is that a program running in normal fashion on the CPU can set up a special control buffer, invoke a special new machine instruction, and the CPU will invoke a new instance of the machine. When that virtual instance returns, it returns to the instruction immediately following the instruction that invoked the virtual machine.
>
> The trick to Rutkowska's approach is to set up the control buffer such that calling the special instruction causes the normal mode system to become the virtual machine and the invoked system to become the controlling instance. Thus the 'normal' machine is now virtualized, analogous to 'the world' in *The Matrix,* which once really was the real world but now is a virtual world.

The name "Blue Pill" alludes to the blue pill the character of Neo in *The Matrix* was offered, but declined, for the red pill.

Since the malware resides on the normal machine, it cannot be found from inside the VM. Blue Pill tricks the VM so that it does not know it is a virtual emulation as opposed to actual physical hardware, and if it is a truly perfect emulation, the end user or administrator will never suspect that anything is amiss.

Rutkowska said that the virus writers could abuse the techniques of nested virtualization to create even more insidious malware.

"Although it could not decrease the security of any system, nested virtualization could be used by malware to virtualize the other hypervisor (the good one)," said Rutkowska. "Imagine Neo, who decided to swallow the Red Pill, eventually awakes on the Nebuchadnezzar ship (after getting out of the Matrix) to find out later that this whole outside-the-Matrix world is just…another Matrix."

Critics leapt to discredit Rutkowska's work, saying that Blue Pill *is* detectable, being that timing techniques can be used to detect the fact that the machine is running in virtual mode. In an interview with Virtualization.info, Anthony Liguori, a software engineer at IBM's Linux Technology Center, explained "Software emulation implies that [certain instructions] take much longer to complete when executed under a VMM than on normal hardware. This fact is what can be used to detect the presence of a VMM."

At the 2007 BlackHat Briefings, security researchers Thomas Ptacek, Nate Lawson and Peter Ferrie demonstrated various techniques that they claimed could detect Blue Pill.

However, later that day, Rutkowska and fellow Invisible Things Lab researcher Alexander Tereshkin presented a BlackHat session demonstrating how Ptackek, Lawson and Ferrie "failed to prove their claims." Rutkowska's counterargument to Ferrie, Lawson, Ptacek and others (explained well in an Aug. 22 interview with SecurityFocus) is that while these methods can indeed detect that virtualization is enabled, they cannot determine whether a VM was created for a legitimate purpose or a malicious purpose.

She said that these days, many companies aren't actively running virtual environments; so for those organizations, the mere fact that virtualization is enabled might be suspicious enough to effectively indicate that the machine's been Blue-Pilled. However, as more organizations begin adopting virtualization for legitimate purposes, admins will expect virtualization to be enabled. If virtualization-enabled is the normal state, then merely detecting virtualization will not be enough to detect virtualization-based malware.

"Distinguishing legitimate virtualization from malicious virtualization sounds logically as being the next step," said Rutkowska in the SecurityFocus interview, "but I somewhat don't see any good methods we could use to do that effectively."

Managing the virtual infrastructure

Perhaps there is no technological solution to Blue Pill; no characteristic activity to scan for. There may however be management mitigations.

For example, Blue Pill couldn't hide itself behind a VM if it wasn't able to and/or authorized to create the VM in the first place. This might be accomplished by adopting the secure virtualization layer outlined and recommended by Garfinkel and Rosenblum. From their paper:

> The heart of a virtualization layer is a high-assurance virtual machine monitor [hypervisor]. On top of it would run a secure distributed storage system, and components replacing security and management functions traditionally done in the guest OS.
>
> Enforcing policies such as limiting VM mobility and connectivity requires that the virtualization layer on a particular machine be trusted by the infrastructure. Virtualization layer integrity could be verified either through normal authentication and access controls, or through dedicated attestation hardware.

(Attestation, by the way, essentially means the ability to validate, at the time of loading, that the only software being loaded is "trusted"; in other words, the only software being loaded is the software that is supposed to be loading. Attestation relies upon the Trusted Platform Module (TPM). Attestation isn't entirely a cure-all, because even if the software is "trusted," it may still be vulnerable. Those vulnerabilities could then be exploited by malware that could run its own code before being detected by the TPM chip upon reboot.)

> Policy at this layer could limit replication of sensitive VMs and control movement of VMs in and out of a management infrastructure. Document control style policies could prevent certain VMs from being placed onto removable media, limit which physical hosts a VM could reside on and limit access to VMs containing sensitive data to within a certain time frame.
>
> User and machine identities at this layer could be used to reintroduce a notion of ownership, responsibility and machine history. Tracking information such as the number of machines in an organization and their usage patterns could also help to gauge the impact of potential threats.

The secure virtualization layer described by Garfinkel and Rosenblum addresses most of the security and management troubles described above. However, the technology needed to create

and maintain such an infrastructure is not quite ready. Rosenblum (a co-founder of VMware) and Garfinkel note that some of the problems presented in their paper "are beginning to be addressed by VMware ACE."

To virtualize or not to virtualize

"Despite all the talk of 'net security gains' and 'cost-effective business models,' virtualization at a fundamental and technical level weakens security," said Oberheide. "In the theoretical best case where the software hypervisor is perfect and 100 percent secure, the [non-virtualized model and virtualized model] are, at best, equivalent in security."

"Virtualization is not a magic bullet," said Roemer. "It is something that you need to implement carefully, as you would any mission-critical technology."

"All in all, there are potential security benefits in using virtualization," said Rutkowska, "but the design and implementation of the whole virtualization system must be carefully reviewed, as otherwise we could have serious problems."

"I think this debate is very temporal," said Mulchandani. "We absolutely feel that, in the long term, virtualization can be a security-enabling technology. But VMware has focused most of our attention on data center operators, and frankly, the data center operators have not engaged the security operators. The engagement is just beginning."

I want to say "wade carefully into virtual waters," but perhaps that's not the right analogy to make. Gingerly testing the virtual waters may open you up to the risks without giving you the security benefits. The wisest course of action may be to wait until 1) the virtualization security tools are more mature and 2) your organization is prepared to adequately invest in the superior security the virtual environment demands. Only then should you strap on your SCUBA gear and take the plunge. —*Sara Peters*