A Survey on the Security of Virtual Machines

Doug Hyde, hyde@wustl.edu (A project report written under the guidance of Prof. Raj Jain)

Download 

Abstract

Virtual machines (VM) are rapidly replacing physical machine infrastructures for their abilities to emulate hardware environments, share hardware resources, and utilize a variety of operating systems (OS). VMs provide a better security model than traditional machines by providing an additional layer of hardware abstraction and isolation, effective external monitoring and recording, and on-demand access. However, this new model requires adaptation of existing security methods, which cannot currently keep up with the ease of creating new VMs with a variety of configurations and lifecycles. Attackers have successfully compromised VM infrastructures, allowing them to access other VMs on the same system and even the host. Fortunately, these security concerns are being addressed and users can prevent most intrusions by applying traditional security measures to each VM.

Keywords

Virtual Machine, VM, Virtual Server, Hypervisor, Virtual Machine Monitor, VMM, Virtualization, VM Sprawl, Security, Emulator, Emulation, VM Sprawl, State Restore, External Monitoring, Denial of Service, VM Escape

Table of Contents

1. Introduction

The cost-effective performance of modern day computers combined with the variance of available operating systems (OS), configurations, and capabilities has given rise to virtual machines (VM). A VM is a software-layer abstraction of hardware that allows a process to execute in an emulated environment. This paper deals only with system VMs, which run a guest OS in a virtual hardware environment. Process VMs, such as the Java Virtual Machine, translate generic calls into platform specific calls. VMs allow users to simultaneously run multiple OSs with unique configurations on a single physical machine. They have many applications in software development, web hosting, and cross-platform computing.

Generic VMs are relatively simple. The hypervisor, also called the virtual machine monitor, runs on the host OS and allocates emulated resources to each guest OS. When the guest makes a system call the hypervisor intercepts and translates it into the corresponding system call supported by the host OS. The hypervisor controls each VM's access to the CPU, memory, persistent storage, I/O devices, and the network [Olzak07]. Figure 1 shows the architecture for a standard VM infrastructure on a single physical machine.

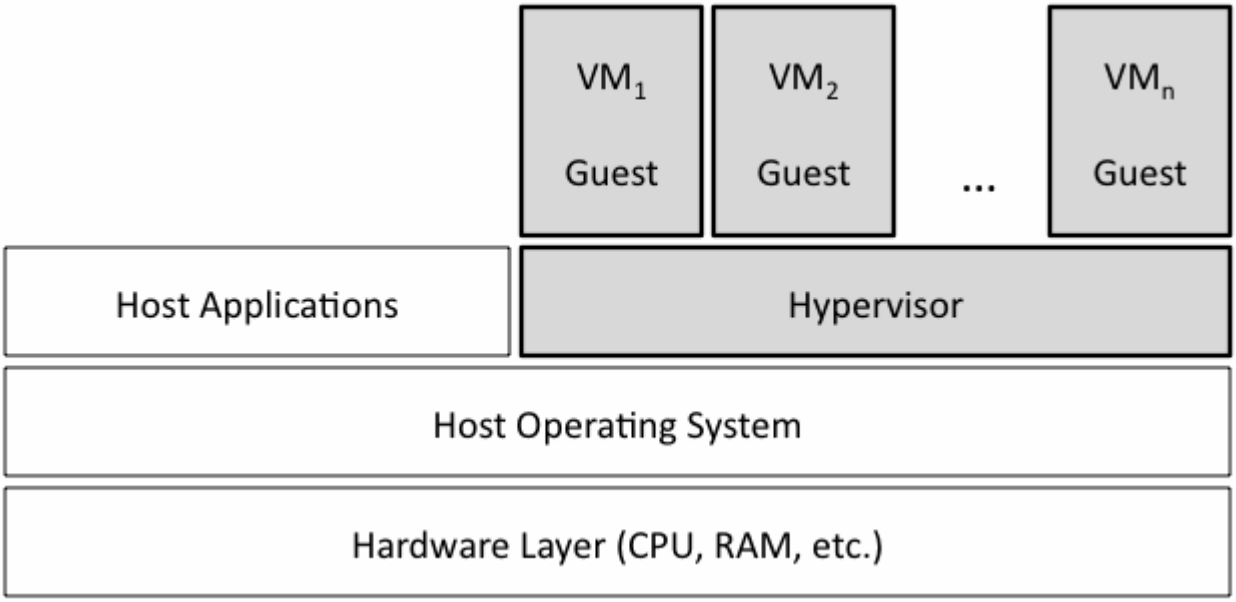

Figure 1. Virtual Machine Architecture

The side effects of virtualization allow VMs to provide security features physical machines do not: isolation, state recording, transience, and mobility. Isolation refers to the encapsulation of each guest OS and abstraction from the hardware, so that each user accesses separate file systems and memory blocks. Ideally a VM compromised by attackers will not affect the host or other VMs on the host. Because each VM is running under the control of the hypervisor, the host OS can easily oversee and record the changes to the system configuration and file data. This allows users to undo system-wide changes to the guest OS, such as a virus attack, file deletion or corruption. Unlike physical servers, which are always on, VMs can be started remotely, allowing them to consume resources only when in use, share hardware and reduce operating costs, since a physical machine is not needed for each installation. Limiting operating time turns out to be a successful technique for limiting intrusion. VMs are often stored as a file, permitting digital transportation to run on another computer.

Unfortunately, VMs are a relatively new technology with security challenges and vulnerabilities. There are three outcomes of an attack that are unfavorable to users: 1) the guest is compromised, 2) multiple guests are compromised, or 3) the host is compromised [Ormandy07]. System administrators can rollback guests to a pre-attack state to restore integrity, but if the host is compromised, attackers have access to hardware and have unlimited freedom. Since it is easy to restore an infected VM, many users have not secured them with basic virus protection. An estimated *60% of virtual machines in production are less secure than their physical counterparts*, due to neglecting to use traditional security measures, which would prevent most attacks to VMs [Brodkin08]. This problem stems from another, called VM sprawl. VMs are extremely easy to make, relative to a physical machine, causing their growth to frequently far exceed the administrators' ability to secure each unique guest.

In addition to local attacks, there are three other classes of attacks on VMs. The attacker may utilize a compromised VM to communicate with other VMs on the same physical host, a violation of the isolation principal of VMs. The second class of attacks are on the hypervisor, which can potentially give the attacker access to the host OS and hardware. Finally, denial of service (DoS) attacks can be particularly successful on VMs because they have the potential to consume resources from all VMs on the host.

VMs are revolutionizing server infrastructure. They allow emulation of many isolated OSs, reducing hardware costs, while providing features such as state restore, transience, and mobility. The three components of a typical VM setup are: the host OS which interacts with the hardware, the hypervisor which allocates resources and manages the VM, and a guest OS run without access to the host OS or physical hardware. VMs are subject to unique attacks in addition to attacks that physical machines face, but can be deterred using similar security methods as applied to normal systems.

---

2. Security Benefits

VMs are rapidly gaining popularity due to their ability to emulate computing environments, isolate users, restore previous states, and support remote initialization. All of these features have positive security side effects. The hardware abstraction and isolation of VM bounds the scope of attack and makes it much more difficult for the attacker to access unauthorized data and resources on the physical machine. VM state restore allows users to return to a state prior to attack or data loss, providing an easy method of malware removal and data preservation. By allowing users to start and stop VMs remotely, attackers have small time windows in which their preparation and attack must take place. This is a surprisingly effective security measure. Since hypervisors run outside the VM, they have the potential to monitor for malware. For these reasons, VM infrastructures have the potential to be safer and more secure than physical server infrastructures.

2.1 Abstraction

VMs abstract the hardware layer and each VM is allocated its own strictly bounded resources. This layer of abstraction provides additional security. When an attacker gains access to the hardware layer, they have full control over the computer. OSs restrict hardware access by abstracting the hardware details, which is why you can run the same OS on two machines with different hardware configurations. In other words, the OS interfaces directly with the hardware so that programmers and hackers cannot. VMs create a complete hardware and OS abstraction. A program run locally on a physical machine knows what OS it is running on. However, as Figure 2 shows, the guest OS and its processes running inside a VM do not know the host OS or hardware configuration at all. The guest is not even aware it is running in a virtualized environment. Since the attacker does not know details of the host environment, manipulating and compromising the machine is much more difficult.

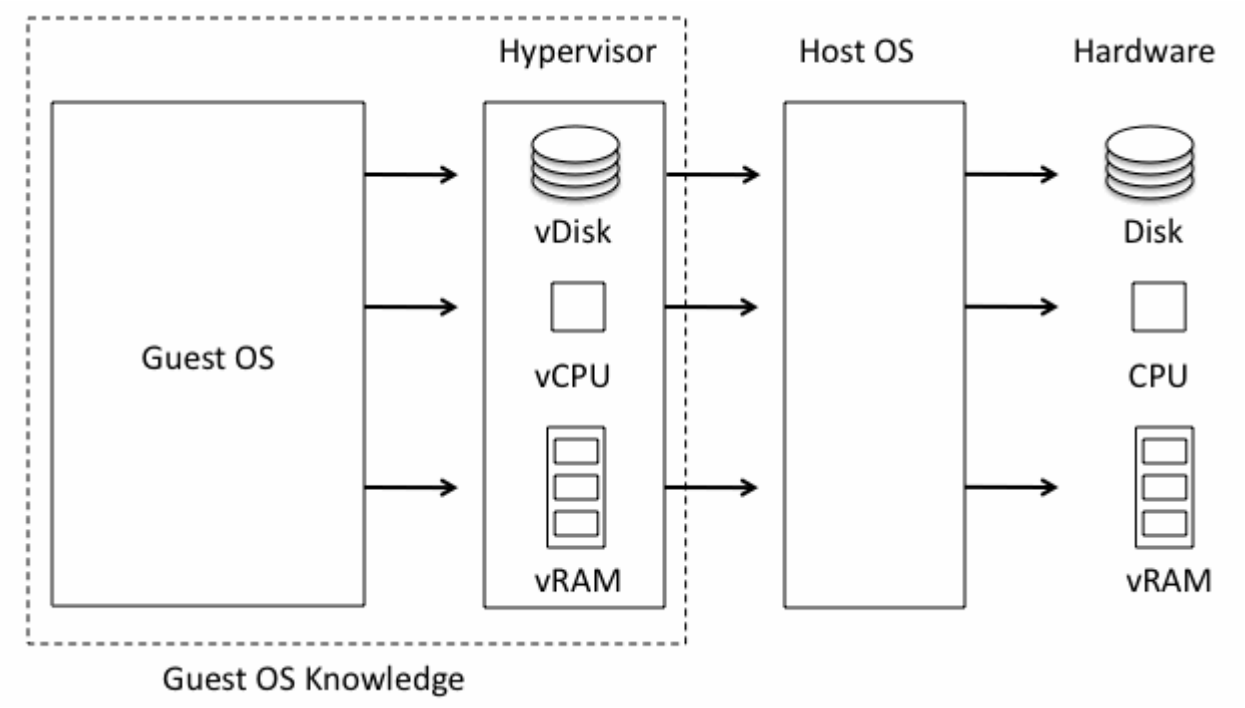

Figure 2. Abstraction of Physical Resources

Hypervisors are much simpler than traditional OSs, and are therefore much easier to secure [Higgins07]. They are *nothing more than a microkernel with a hardware compatibility layer* and a smaller code base correlates to simplicity, making it easier for software developers to minimize bugs and vulnerabilities [Garfinkel05]. The primary job of the hypervisor is to allocate and interface physical resources to each VM. Each VM encapsulates the guest and prevents a malicious guest from accessing resources it does not own. This means that attackers should not be able to compromise the physical machine, only one VM at a time.

2.2 Isolation

The hypervisors segment physical resources into isolated entities and allow each guest OS to run independently. An attack on the VM should not affect any of the other VMs on the server or the host OS. This is unlike a multi-user OS, where all users can be affected by an attack. In a VM infrastructure, each user can either access a given VM or they cannot. Inside a multi-user OS, a complex and thus vulnerable file system dictates which files each user can read, write, or execute. The isolation and abstraction of VMs provides additional security over a traditional multi-user computer. When a VM is compromised, the hypervisor can remove that VM or restore it to a state prior to attack.

2.3 State Restore

VMs are touted for their ability to restore to a previous state. The contents of the virtual disk for each VM are usually stored as a file on the host. Most VMs take a snapshot of the contents of the virtual disk when changes are made or on a time interval. Besides being a convenience, state restore helps to ensure data integrity and provides perfect virus removal. Services on physical machines, such as backup and Window's System Restore attempt to provide this feature, but are generally not as complete as they do not combine all system settings and files into a comprehensive state. Unfortunately, state restore complicates security processes as it can reintroduce un-patched states, creating inconsistencies on the server infrastructure [Garfinkel05].

2.4 Transience

One often-overlooked security feature of VMs is their ability to be started remotely, which allows them to be turned on and made available only when needed. Physical servers are usually always on, even when they are not in use. Minimizing how much time a given computer is online is the best deterrent against malicious attacks, since an offline server cannot be accessed. For example, if a worm infects one computer on the network, only the online VMs can be infected [Garfinkel05]. Since a VM can be used on-demand, it should be in use at all times, which means that it is being directly monitored. A user is much more likely to detect intrusion while they are using a computer than when they are not.

2.5 External Monitoring

Since VMs run on a subset of hardware resources, it is possible observe VM resource usage and detect malicious software from outside the VM. Physical installations of OSs usually rely on installed virus protection. However, sophisticated attackers have found many ways to disable the virus protection, giving them access to an unprotected OS. However, VMs can be monitored by either the hypervisor, or an authorized dedicated VM that can view software activity. The later is the preferred method since it limits the hypervisor's role, helping to keep the hypervisor as simple and secure as possible. The hypervisor simply gives the dedicated VM permission to view resources allocated to the monitored VM. Figure 3 shows an example of a single dedicated VM that is only used to monitor other active VMs. *These monitors are used in intrusion detection systems (IDS), integrity checking, honeypot systems, and forensic analysis, among others* [Payne07]. Malicious code should not have access outside of the VM, so if the VM is being monitored by an outside process, then the VM can be shut down or disabled in an attack is detected by the dedicated monitor and hypervisor. The paradox of physical machines is that if they have a virus, they cannot reliably detect whether they have a virus. VMs do not have this problem.

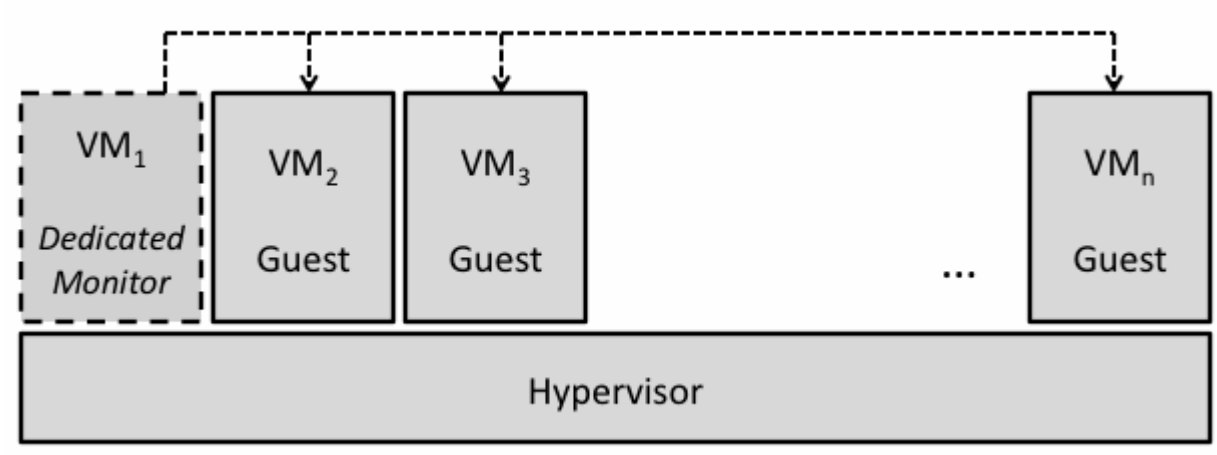

Figure 3. Dedicated External Monitor (VM1)

---

3. Security Maintenance

A VM infrastructure is drastically different than a physical machine infrastructure and as a result, IT processes require significant adaptation. All of the benefits are VMs over physical machines are inherently differences that require new security measures. Though VMs can be restored easily, it is important to protect them with traditional virus protection to avoid more serious attacks on the host. VMs can be created rapidly, each with a unique configuration, complicating the process of securing all machines exponentially. The dynamic moves and state restore of VMs interferes with traditional patching and testing cycles, as a guest can be rolled-back to a previous vulnerable state [Higgins07]. Security IT must keep record of all VMs to ensure that all VMs are monitored and maintained.

3.1 VM Sprawl

The single biggest vulnerability of VMs is due to the ease in which users can create many VMs, which become very difficult to secure, monitor, and maintain. VMs can be created and deployed in a matter of seconds, which is significantly smaller than the time to ensure that all VMs are up-to-date and secure. Traditional security methods need to be applied to each VM because the guest OS accesses the network directly [McLaughlin07]. A compromised VM is a potential entry point for attackers to the hypervisor and host [Brodkin08]. There are significant manual processes each time a VM is made. Most current security products are not designed for efficient use in securing many VMs on the same physical server, but this is a problem many software companies are working on. In addition to creating more work for security administrators, VM sprawl wastes resources and creates more entry points for attackers [Garfinkel05]. An obsolete VM becomes an easy entry point for attackers, who could potentially access the hypervisor, other VMs, and the host OS.

3.2 Unique Configurations

Virtualization is often used for emulating a variety of OS configurations. For example, software developers wishing to test their software on both current and older OSs with and with out certain updates to ensure the product will work for all clients. The traditional infrastructure security model is to work closely with one securely configuration that can be used on all computers. By allowing different OSs, this manual work becomes increasingly more complicated. Compounding to the problem is the fact that some VMs may intentionally not have all patches to ensure software works with and without those patches. Unlike shared physical servers, where each user is given an account with limited rights and permissions, VM infrastructures often give each user an administrator account to the guest OSs. This makes it more difficult for system administrators to ensure that each VM is secure, since the users often have the permissions to remove security measures.

3.3 State Restore

The ability of a VM to restore to a previous state is often considered a security benefit to protect against data loss. However, returning to an unpatched or compromised state is a great danger. When a new security update is released, physical machines are patched and remain patched. A VM may also get the security patch, but if for some reason the user needs to rollback to a previous state, then the guest is no longer patched [Garfinkel05]. The biggest challenge is for system administrators to record when patches are made and evaluate which patches need to be applied again when a VM is restored to a previous state.

An even bigger concern is returning to a contaminated state. Often machines are infected with viruses and are not detected until updates are made to virus protection software. If a user returns to a state prior to virus removal, the virus may or may not exist on the system, since the origin of the virus is unknown. This scenario is illustrated in figure 4.

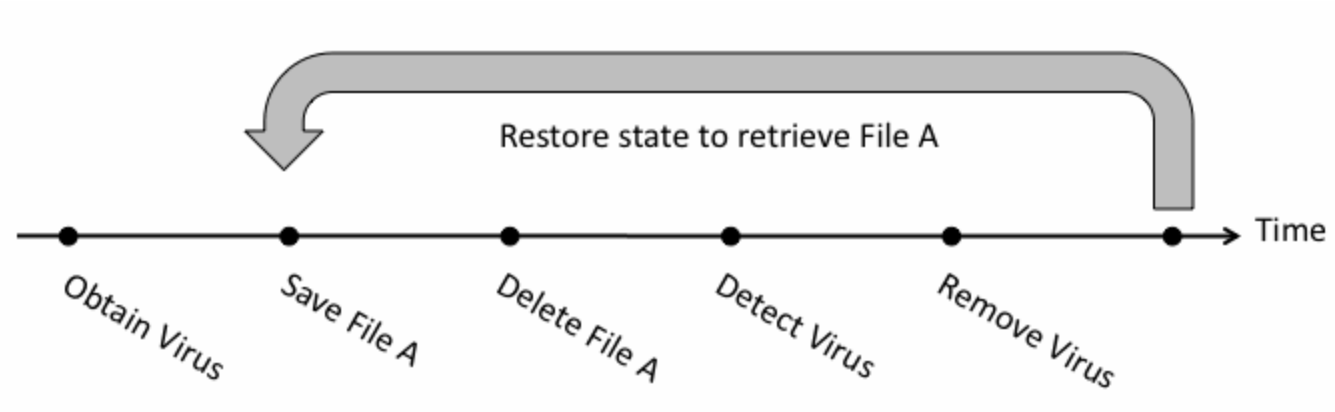

Figure 4. State Restore to Malicious State

One security doctrine *for building secure systems is minimizing the amount of time that sensitive data remains in a system* [Garfinkel05]. However, the state restore feature of VMs violates this principle since all information that was ever on the guest remains on the host indefinitely. If an attacker compromised the hypervisor and gained access to the state information for each virtual machine, he or she could access all information that was ever contained on the VM.

3.4 Transience

The VM model *gives rise to a phenomenon in which large numbers of machines appear and disappear from the network sporadically*, known as VM transience [Garfinkel05]. VM transience limits the window in which attackers can attempt to compromise the system, but it also makes security maintenance and audits more challenging. When a worm attacks a traditional server infrastructure, all vulnerable machines are rapidly infected. Then system administrators evaluate which machines are infected and correct the problem.

However, a virtual machine could become infected, go offline before detection, come online at a later time, and re-infect all vulnerable machines. There is a tradeoff between security and spontaneity. Traditional security update cycles require that all machines are online simultaneously for patching, virus removal, audits, and configuration changes. Transience itself not a security vulnerability, but it complicates security processes for system administrators, potentially opening up virtual machines to many different vulnerabilities.

---

4. Security Vulnerabilities

There is nothing about virtual computing that is inherently unsecure; it is just *a new security attack vector* [Higgins07]. The virtual machine layer is more secure than any OS, due to its simplicity and strict access control. Compromising the hypervisor could give attackers access to all virtual machines controlled by it and possibly the host, which makes the hypervisor a compelling target. Unauthorized communication between guests is a violation of the isolation principle, but can potentially take place through shared memory.

Like physical machines, VMs are vulnerable to theft and denial of service attacks. The contents of the virtual disk for each virtual machine are usually stored as a file, which can be run by hypervisors on other machines, allowing attackers to copy the virtual disk and gain unrestricted access to the digital contents of the virtual machine. Since VMs share resources from the physical machine, VM infrastructures were particularly vulnerable to denial of service attacks, which could starve resources from all VMs on the physical machine. Fortunately, this problem is easily fixed by limiting resource consumption per each VM. Newer products solve many of these problems, however they remain concerns that hypervisors much continue to consider in development.

4.1 Mobility

Virtual machines are inherently not physical, which means their theft can take place without physical theft of the host machine. The contents of the virtual disk for each VM are stored as a file by most hypervisors, which allows VMs to be copied and run from other physical machines. While this is a convenience feature, it is also a security threat. Attackers can copy the VM over the network or to a portable storage media and access data on their own machine without physically stealing a hard drive [Garfinkel05]. Once attackers have direct access to the virtual disk, they have unlimited time to defeat all security mechanisms, such as passwords, using offline attacks. Since the attacker is accessing a copy of the VM rather than the original, the VM will not show any records of intrusion.

The second danger of virtual disks is that the attacker could corrupt or externally modify the file while the VM is offline [Garfinkel05]. This means the integrity of an offline VM may be compromised if the host is not securely protected. This is not usually an issue with physical servers because the machine must be running to be accessible by network. Modern encryption and integrity algorithms can be applied by the host hypervisor so that access and integrity are ensured.

4.2 Hypervisor Intrusion

The hypervisor provides the abstraction and resource allocation between the host and guests. Attackers' ultimate goal is to compromise the hypervisor to gain with the ability *to execute arbitrary code on the host with the privileges of the [hypervisor] process* [Ormandy07]. The hypervisor is a program, running on the host, so if it is compromised, all VMs it controls and the host itself are accessible to the attacker. The hypervisor converts instructions for the guest OS into instructions for the host OS, however, if the guest is compromised, then the instructions sent to the hypervisor may be irrational.

Researchers ran two tools, crashme and iofuzz, which generate random I/O port activity inside the virtual machines, until the hypervisor or VM crashed [Ormandy07]. The details of these attacks are not included in this paper because they were unique to each VM implementation, and all flaws detected we fixed before the paper was published. They were able to compromise the hypervisor and access the host using any major hypervisor available to the public. However, hypervisors are generally more secure than OSs, since the hypervisor is a relatively small and simple program, which helps to limit low-level vulnerabilities that may be present in many programs [Higgins07]. Ultimately, if the guest OS is not secured security flaws have the potential to sneak through the hypervisor layer.

4.3 Hypervisor Modification

It does not matter how secure the original hypervisor is if it can be externally modified to use the attacker's software. One attack of this form is known as Virtual Machine Based Root Kits (VMBR) [King06]. In this attack, the hypervisor's system calls to the host OS are changed to run malicious code instead. Fortunately, there are several methods for preventing hypervisor modification. The host can use a trusted platform module (TPM) to create a trusted relationship with the hypervisor [Olzak07]. Alternatively, the guest can verify the integrity of the hypervisor when it boots. A third solution is to use an embedded hypervisor. An embedded hypervisor runs without a host OS and inherently cannot be modified [McLaughlin07].

4.4 Communication

VM Communication refers "guest-to-guest" attacks, in which attackers use one VM to access or control other VMs on the same hypervisor. These attacks can happen with out without compromising the hypervisor layer. A malicious VM can potentially access other VMs through shared memory, network connections, and other

shared resources [Olzak07]. For example, if a malicious VM determines where another VM's allocated memory lies, then it could read or write to that location and interfere with the other's operation. Figure 5 shows an attack from VM1 on VM2 and VM3. The attacker may or may not be authorized to access VM1, but in this example has accessed two unauthorized VMs.

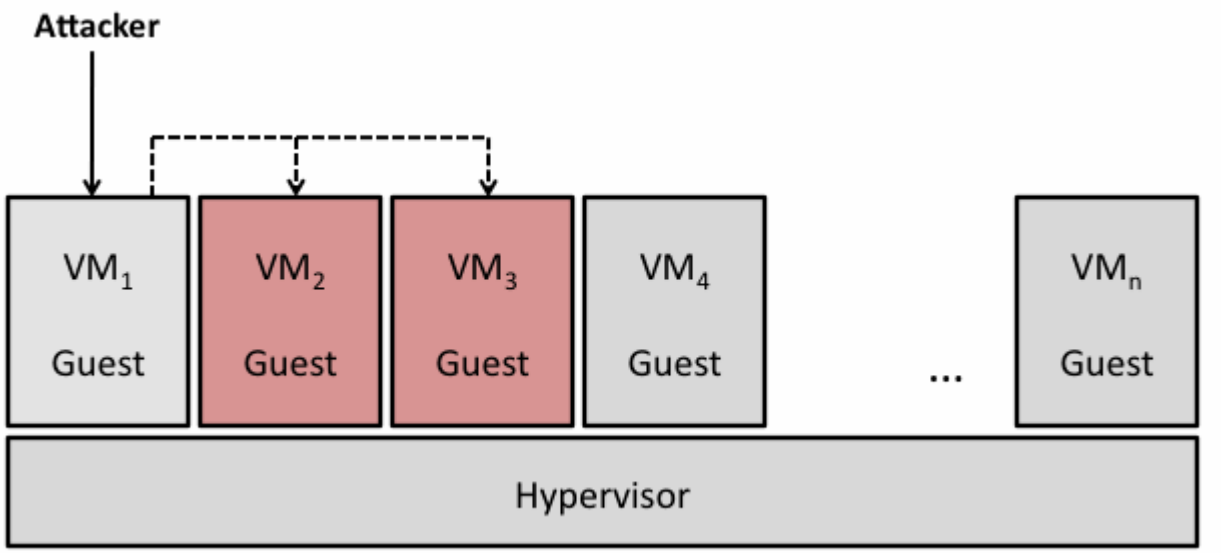

Figure 5. VM Communication Attack on VM2 and VM3

Sometimes it is desirable for two VMs to be able to communicate. This has many applications, such as for dedicated monitoring of a VM or implementing a network technology that requires multiple peers. Instead of providing full isolation, IBM's sHype is intended to be a secure architecture for addressing inter-VM communication in a VM infrastructure. sHype allows system administrators to define policies for communication between each VM and ensure that only authorized VMs have access to communicate with each other [Olzak07].

4.5 Denial of Service

DoS attacks are a threat to all servers, however an improperly configured hypervisor can allow a single VM to consume all resources, thus starving any other VM running on the same physical machine. DoS attacks make network hosts unable to function since critical processes do not have the hardware resources to execute in a timely manner. Fortunately, the solution is simple. Hypervisors prevent any VM from gaining 100% usage of any resources, including CPU, RAM, network bandwidth, and graphics memory [Kirch07].

Additionally, the hypervisor can be configured so that when it detects extreme resource consumption, it can evaluate whether an attack is being made and automatically restart the VM. VMs can usually be initialized much faster than physical machines because their boot sequence does not need to initialize and verify hardware, so restarting the VM has a smaller effect than restarting a physical machine.

---

5. Summary

The rapid advancement of computer hardware and the growing demand for specialty OSs has led to the implementation of VMs on server platforms. VMs allow multiple OS installations to share the same hardware resources. The hypervisor manages these resources and to create the virtual environment for each guest OS.

Modern VMs have many enticing features that have different security qualities than physical machines. The hypervisor provides an additional layer of abstraction from physical hardware and further restricts malicious attempts to control the machine from the hardware. This abstraction encapsulates malicious attacks and allows external monitoring for malicious attacks on a VM. Since the hypervisor monitors each VM, it can record the states and allow the VM to return to a previous state, which has many backup and malware removal advantages. However, this feature also means that a VM could return its guest OS to a previous state where a virus is present or a security patch isn't. VMs are usually not left running all the time like physical servers, drastically limiting attackers' ability to plan and attempt attacks.

Virtualization itself is not inherently unsecure; it is a new technology that potentially has new vulnerabilities and requires restructuring of manual security processes. One of the biggest challenges is to maintain and secure all of the VMs, since many instances and configurations can be rapidly created. The contents of each guest OS is a virtual disk, stored as a file. If this file is accessed, copied, or modified on the host by an unauthorized party, then the privacy and integrity of the VM is compromised. Likewise, if an attacker accesses the host and directly modifies the hypervisor, then he or she will be able to run arbitrary code. The hypervisor should strictly control communication between VMs and limit resource consumption of each VM to a finite bound to prevent DoS attacks. All known vulnerabilities of VMs can be prevented, but it is absolutely essential to secure the host and each guest OS in order to create a secure virtual environment.

References

1. [Garfinkel05] T. Garfinkel, M. Rosenblum, "When Virtual is Harder than Real: Security Challenges in Virtual Machine Based Computing Environments," USENIX Association, 2005. http://www.stanford.edu/~talg/papers/HOTOS05/virtual-harder-hotos05.pdf

2. [Kirch07] J. Kirch, "Virtual Machine Security Guideline," The Center for Internet Security, 2007. http://www.cisecurity.org/tools2/vm/CIS_VM_Benchmark_v1.0.pdf

3. [Ormandy07] T. Ormandy, "An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments," Google, Inc., 2007. http://taviso.decsystem.org/virtsec.pdf

4. [Olzak07] T. Olzak, "Secure hyporvisor-based virtual server environments," Tech Republic, 2007. http://blogs.techrepublic.com.com/security/?p=160

5. [Higgins07] K. Higgins, "VMs Create Potential Risks," Dark Reading, 2007. http://www.darkreading.com/security/management/showArticle.jhtml?articleID=208804369

6. [Brodkin08] J. Brodkin, "Virtual server sprawl highlights security concerns," Network World, 2008. http://www.networkworld.com/news/2008/043008-interop-virtual-server-sprawl.html

7. [Payne07] B. Payne, M. Carbone, W. Lee, "Secure and Flexible Monitoring of Virtual Machines," IEEE Xplore, 2007. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4413005

8. [McLaughlin07] L. McLaughlin, "How to Find and Fix 10 Real Security Threats on Your Virtual Servers," CIO, 2007. http://www.cio.com/article/154950

9. [Sailer05] R. Sailer, E. Valdez, T. Jaeger, R. Perez, L. Doorn, J. Griffin, S. Berger, "sHype: Secure Hypervisor Approach to Trusted Virtualized Systems," IBM, 2005. http://domino.watson.ibm.com/library/cyberdig.nsf/3addb4b88e7a231f85256b3600727773/265c8e3a6f95ca8d85256fa1005cbf0f

10. [King06] S. King, P. Chen, "SubVirt: Implementing malware with virtual machines," IEEE Symposium on

Security and Privacy, 2006. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1624022

List of Acronyms

DoS	Denial of Service
OS	Operating System
RAM	Random Access Memory
TPM	Trusted Platform Module
vCPU	Virtual CPU
VM	Virtual Machine
VMBR	Virtual Machine-Based Root Kit
vRAM	Virtual RAM

Last Modified April 21, 2009
This and other papers on latest advances in network security are available on line at http://www.cse.wustl.edu/~jain/cse571-09/index.html


Back to Raj Jain's Home Page