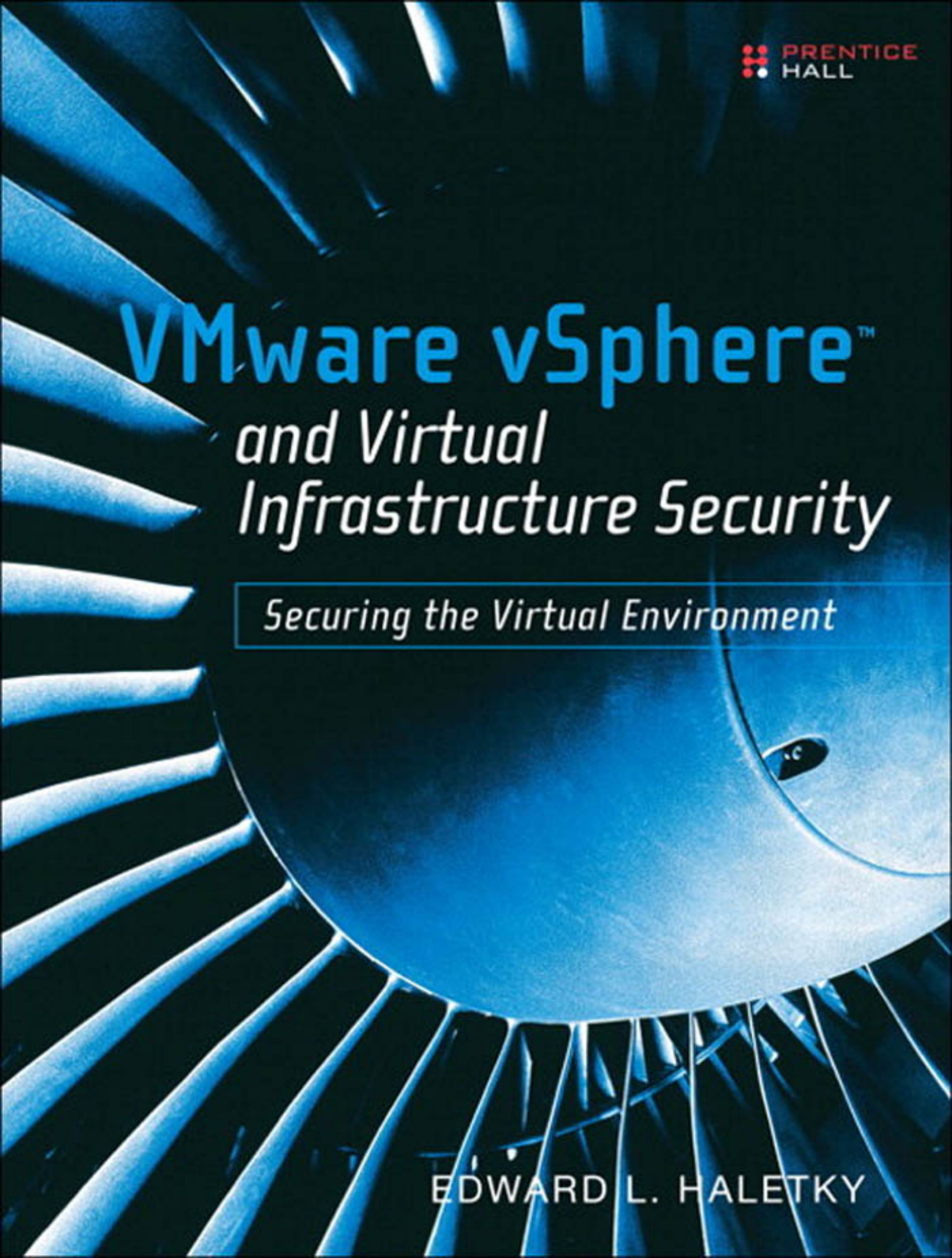
PRENTICE HALL
VMware vSphere™
and Virtual Infrastructure Security
Securing the Virtual Environment
EDWARD L. HALETKY

Library of Congress Cataloging-in-Publication Data

Haletky, Edward.
VMware vSphere and virtual infrastructure security : securing the virtual environment / Edward L. Haletky.
p. cm.
Includes index.
ISBN 978-0-13-715800-3 (pbk. : alk. paper) 1. Virtual computer systems—Security measures.
2. Cloud computing—Security measures. 3. VMware vSphere. 4. Computer security. I. Title.
QA76.9.V5H36 2009
005.8—dc22

2009018924

ISBN-13: 978-0-137-15800-3
ISBN-10: 0-137-15800-9

Text printed in the United States on recycled paper at R.R. Donnelley in Crawfordsville, Indiana.
First printing June 2009

Editor-in-Chief
Karen Gettman

Acquisitions Editor
Jessica Goldstein

Senior Development Editor
Chris Zahn

Managing Editor
Kristy Hart

Project Editor
Andy Beaster

Copy Editor
Barbara Hacha

Indexer
Erika Millen

Proofreader
Linda Seifert

Publishing Coordinator
Romny French

Cover Designer
Chuti Prasertsith

Compositor
Nonie Ratcliff

Foreword

Virtualization and Security. To many, these two terms seem like strange bedfellows considering the perception that they are, by nature, definition and historical application, diametrically opposed.

Virtualization offers agility, mobility, cost-effectiveness, flexibility and an infrastructure dynamism that abstracts our applications and critical information from the tightly coupled affinity of the infrastructure that serves it up.

The atomic unit of the virtualized datacenter is no longer the monolithic server. The physical machine is replaced by the virtual machine—a fluid packaging of operating system, applications and information divorced from its physical underpinnings.

As such, the notion of how we design, deploy, manage, interact with and ultimately secure the interleaved fabrics of our new computing, network, storage, and information resources have fundamentally changed in an amazingly compressed timeframe.

Yet with all this amazing progress, Security as it stands today is still often found designed and operationalized around a primarily static, inflexible, and hardware-centric view of the datacenter and its assets. The folly of our ways is starting to catch up with us.

The Maginot lines drawn in the sand many years ago, encircled by the crumbling walls of outdated approaches and mismatched expectations, are rapidly eroding thanks to ineffective technology, innovative attackers, and an always-on, collaboration-hungry generation of information junkies.

Virtualization is a well-needed forcing function, a wakeup call for the security industry, its practitioners and architects; it reshapes the discussion of who, how, where and why Security gets done.

Virtualization is redefining the charter of Security and causes us to think within the context of solving enterprise architecture and business problems across the entire stack of solutions holistically. In some cases this means that nothing changes, while with others, everything changes. Knowing how to identify these scenarios is critical.

One must embrace a pragmatic approach with regard to how to secure virtualization, how to virtualize security and ultimately become more secure through the application of virtualization.

As virtualization platform providers such as VMware enable a new spectrum of capabilities across our computing experience, it's time we take what works, scrap that which does not benefit us, and move on. It's time to take advantage of this shift in how we do what we do in security.

Use this book, Edward's intimate knowledge of VMware's virtualization platform and his sage, rational security experience to guide you through the process of setting the foundation toward designing, implementing, and managing a secure virtualized infrastructure.

It's the first of many steps, but you've taken the most important one—choosing the right guide for the journey.

Onward toward a secure virtualized future!

Christofer Hoff
Virtualization Security Pundit and Evangelist

Preface

A majority of VMware ESX or VMware ESXi installations trust either old-school security practices or the security provided by VMware. Although these approaches are a good start, neither provides a secure virtual environment. Virtualization security covers a wide range of subjects that either adapt old-school security methods to the virtual infrastructure or provide brand-new security methods. It is a growing field that addresses the issues of securing the virtual infrastructure, including regulatory compliance, system hardening, intrusion detection and prevention, business continuity, monitoring, assessment, and digital forensics, just to name a few. I define old-school security methods as those that treat virtualization hosts just like another physical host within the data center.

Virtualization introduces its own security problems into the mix that composes the data center. The introduction of virtualization will drastically change the security stance of even the most secure environments. New knowledge is required to combat these issues, and this book provides a starting point for those just starting in virtualization security, a reference for the security professional, and a much-needed information source for the existing VMware Virtual Infrastructure administrator. Whereas this author's previous book, *VMware ESX Server in the Enterprise*, provided a primer on virtualization security, this book covers the breadth and depth of knowledge needed to fully design and articulate virtualization security within your new and existing environments.

To the author, VMware vSphere or the Virtual Infrastructure are not just VMware ESX or VMware ESXi, but also the range of VMware and third-party add-on products that provide management, business continuity, disaster recovery, and security. All add-ons need to be secure and protected from the hacker as well as

the inadvertent or purposeful violation of the stated security policy by an administrator or other employee. Figure P.1 illustrates the full VMware Virtual Environment covered by this book. This environment is not limited by versions or types of virtualization servers; all are considered herein. In other words, the techniques discussed apply to VMware Virtual Infrastructure v3.x as well as vSphere 4 and can be applied even to VMware ESX v2.x, Xen, and Hyper-V. The concepts are the same throughout all these products; however, the implementations are different.

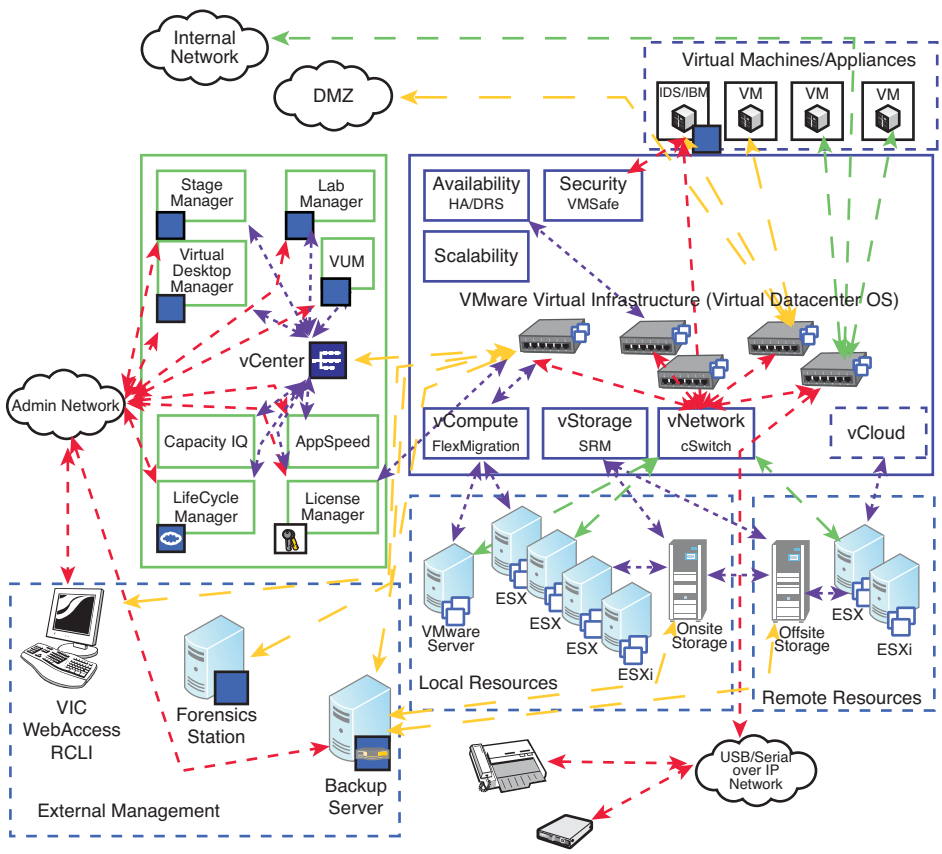

Figure P.1 Full VMware Virtual Environment external view

As you can see, this book covers much more than just the hardening of VMware ESX. We cover every aspect of the virtual environment using the maxim that the security of the virtual environment is based on the security of anything

that directly or indirectly touches the VMware vSphere or Virtual Infrastructure. The virtual environment is composed of seven major areas, each of which we will break down in the later chapters.

1. The solid box labeled vCenter is the management component now named vCenter in the new terminology of the virtual data center operating system (VDC-OS). vCenter encompasses the VMware vCenter Server (previously VMware VirtualCenter Server), VMware vCenter Lifecycle Manager (previously VMware LifeCycle Manager), VMware vCenter Stage Manager (previously VMware Stage Manager), VMware vCenter Update Manager (previously VMware Update Manager), VMware vCenter Lab Manager (previously Vmware Lab Manager), VMware CapacityIQ, VMware View Administrator (previously VMware Desktop Manager), VMware AppSpeed, VMware vCenter Converter (previously VMware Converter), and many other tools that now or in the future integrate with VMware vCenter Server.
2. The solid box labeled VMware vSphere or Virtual Infrastructure (Virtual Datacenter OS) contain the products layered above the traditional virtualization hosts like VMware High Availability (HA), Dynamic Resource Scheduling (DRS), VMotion, Storage VMotion, Fault Tolerance (FT), and so on.
3. The dashed box labeled Local Resources contains the local virtualization hosts involved.
4. The dashed box labeled Remote Resources includes systems at hot sites.
5. In the dashed box labeled Virtual Machines and Appliances, we must also consider to what these are connected.
6. The dashed box labeled External Management contains those aspects outside the realm of the traditional virtual environment: forensic workstations, management workstations, and backup servers using traditional backup tools, those specific to VMware ESX, or VMware Consolidated Backup.
7. The last major area is to all the external networks the virtual environment connects; these clouds could be Storage, Production, DMZ, Serial or USB over IP, Administrative, Test, Development, or QA networks, to name a few. How these networks connect to the virtual environment is very important.

Security Note
The security of the virtual environment is based on the security of anything that directly or indirectly touches the VMware vSphere or Virtual Infrastructure.

These seven areas compose the entire virtual environment, which, mentioned previously, we break down in later chapters of this book. Each of the chapters is described in the section "What This Book Covers."

Figure P.2 presents the internal aspects of the VMware vmkernel that are also covered by this book. Understanding the internal aspects of the hypervisor in use will directly affect your security design. We cover several major concerns with the hypervisor in Chapter 3, "Understanding VMware vSphere and Virtual Infrastructure Security."

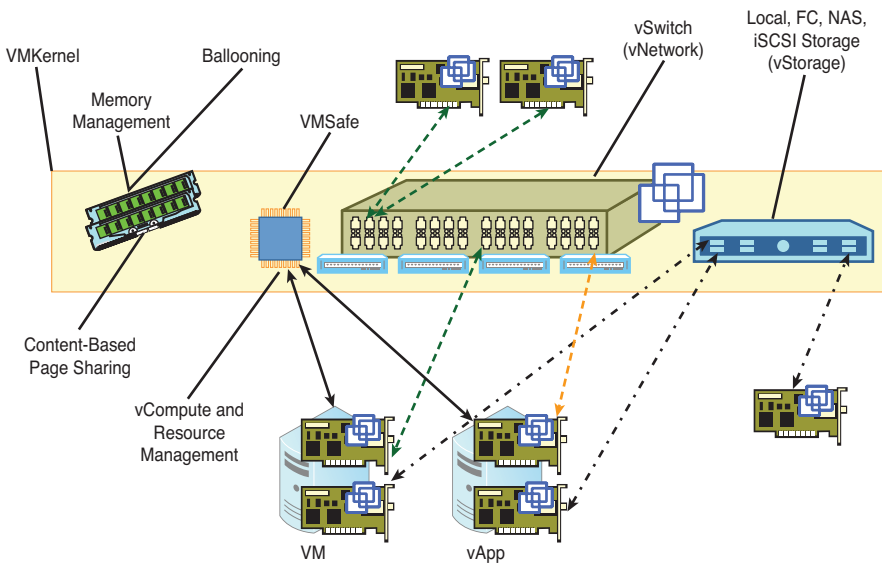

Figure P.2 VMware vSphere and Virtual Infrastructure internal view

Figure P.2 covers the key areas of the hypervisor, or in VMware technical parlance, the vmkernel. The vmkernel controls how the memory of the virtualization host is accessed and overcommitted using three major features, Content-Based Page Sharing (CBPS), commonly referred to as Transparent Page Sharing (TPS),

memory ballooning, where memory can be borrowed from a VM that is not currently using it, and standard memory management and assignment. In addition, the vmkernel offers many new application programming interfaces (APIs) that fall under the VMsafe heading. VMsafe allows specialized virtual appliances access to vmkernel to enable even more security features. In addition, the vmkernel is a computing resource for running virtual machines (VMs) using tools like resource pools and resource shares to limit how many resources VMs, virtual appliances, or vApps, can use during their life. The vmkernel also contains within it a Layer 2 switch, which is referred to as the virtual switch (vSwitch) and is a major feature of the virtual network or vNetwork component of the VDC-OS. Finally, the vmkernel interacts directly with local, network, and Fibre Channel storage. The storage layer is a major component of the vStorage component of the VDC-OS.

The vmkernel presents a single virtualization host that acts as a computing resource, a member of your network switch infrastructure, and your storage infrastructure; the virtualization host is therefore a hybrid device. Old-school security would not consider the virtualization host as a hybrid network, storage, and compute appliance, but as a single physical resource that must follow the security policy for all physical hosts.

In Figure P.2 the solid lines represent vCompute interactions between the vmkernel and VMs or vApps. The dashed lines represent the vNetwork interactions between the vmkernel, physical network interface cards, and VMs or vApps. The dot-dot-dashed lines represent the interactions between the vmkernel, physical storage adapters, and VMs or vApps.

As you can see from the figures, there is quite a bit to worry about with regard to security. The types of lines in Figure P.1 are commonly used to define different zones within a firewall. We use these throughout the book. Because none of the figures are in color, the mappings are defined in Table P.1. The color mappings are used by security experts to determine the risk associated with the various networks, components, and the appropriate protections required for each. Although no standard exists for the color mappings, these colors are frequently used within firewall documentation. For those who understand what these colors mean, it will aid in helping you understand how to think about the virtual environment. The types of lines used for boxes within the diagram are there to define the different areas of the diagram and are unrelated to the line types defined in Table P.1.

Table P.1

Line Type to Color Mapping			
Security Zone	**Line Type**	**Definition**	**Sample Line**
Green	Short Dash-Dot-Dot Line	Internal Protected Network deemed safe.	__ .. __ .. __ .. __ ..
Purple	Dotted Line	Internal Protected Network needing isolation (generally wireless but applies in our case to storage networks and those things within the VDC-OS).	.....
Orange	Long Dash Short Dashed Line	DMZ deemed unsafe.	__ ____ __ ___ __
Red	Short Dash Line	Deemed unsafe; in our case this is not necessarily the Internet, but everything outside the virtual environment.	__ __ __ __ __ __

We use the preceding definitions throughout our discourse on virtual security. In essence, we are predefining the attack zones that we are concerned about. Attack sources exist within all areas of Figure P.1 as well as the interactions between the areas. This sounds rather broad to some people; however, it is important to realize that threats come from everywhere, and we should at least consider all sources within this book. Although they may not apply to your specific installations, they should be considered from a security perspective. That is the main goal of this book—to bring up often overlooked bits of information, tying together system hardening, network security, penetration testing, and digital forensics to improve the overall security of the virtual environment.

Who Should Read This Book?

Virtualization administrators should read this book to aid their discussions with the security administrators within their organizations. Security administrators should also read this book to understand how virtualization affects their security plans. Those who design and architect business processes should also read this book, because security starts at the beginning or may need to be redesigned and architected into an existing business process when virtualization is imposed upon or implemented within an existing environment. Last, this book is designed for the virtualization beginner as well as the expert. Although this could be considered an intermediate and very specific book, all that is required to derive value

from this book is an open mind and a basic understanding of virtualization or security. Most discussions on security require thinking outside the box, because the attackers are doing just that. An open mind will be a great aid in reading this book.

What This Book Covers

What follows is a rundown of the chapters in this tome, which encompasses virtualization security applied to the VMware vSphere and Virtual Infrastructure.

Chapter 1: What Is a Security Threat?

This chapter describes the major differences between existing security thinking and the new thinking required for virtualization security. We start by looking at security from a 10,000-foot view, comparing the new to the old to clearly define the common body of knowledge required to understand virtualization security and why we need to improve it. In addition, we define these key concepts: security threat, fault, and vulnerability. Using these definitions we look at the virtual environment for issues regarding the specific targeting of VMs, the fact that virtualization administrators are not security administrators, and that often, simple mistakes and misunderstandings can lead to security issues. Last, we look at how a security policy helps to define what is acceptable and what is not and why implementing virtualization will change security policies.

Chapter 2: Holistic View from the Bottom Up

The first thing this chapter does is reverse itself to look at virtualization security from the point of view of the attacker, whether a hacker, script kiddie (one who runs a script created by a hacker), or disgruntled employee. We discuss specific threats to and within the virtual infrastructure while applying the definitions from Chapter 1. If you do not understand the threats to the virtual infrastructure, how can you secure it? This chapter covers the first oft-overlooked aspect of virtualization, as most of the bolt-on security tools protect you from network threats to the VMs, but what about the core of virtualization, the host? This chapter answers the question of whether virtualization security requires a change to a security policy, architecture, design, or implementation. This chapter was coauthored by contributing author Tim Pierson.

Chapter 3: Understanding VMware vSphere™ and Virtual Infrastructure Security

Now that we understand the threats to the virtual environment, we'll discuss the vmkernel. We look at how it alleviates security threats by looking at how it handles memory, networking, and processes. Small but vital, the vmkernel alleviates only part of the security issues within the environment. In addition, VMware introduced VMsafe, access to the vital vmkernel API so that third parties can add in their own security features. This new feature will have a large impact on the implementation and security in general within the virtualization environment. This chapter also discusses what is known about VMsafe and the impact of its use. Last, this chapter discusses the use of common virtual machine interface (VMI) paravirtualization, VMware Tools, and their possible impact on security.

Chapter 4: Storage and Security

Because storage plays a large part in the capability to use the VMware Virtual Infrastructure and its tools, this chapter reviews each of the supported storage technologies with an eye toward security and current threats. We investigate authentication, hardware/software disk encryption, virtual storage networks (VLAN, NPIV), and isolation. Last, we include suggestions for securing local and remote storage.

Chapter 5: Clustering and Security

VMware vSphere 4 and Virtual Infrastructure 3 clustering employs five distinct technologies, each of which affects security in different ways. This chapter looks at security from the perspective of VMware High Availability (HA), VMware Dynamic Resource Scheduling (DRS), VMware vMotion, VMware Distributed Power Management (DPM), and VMware Storage vMotion. vSphere 4 introduces three other technologies into the VMware Cluster: VMware VMware Host Profiles (HP), VMware Distributed Virtual Switch (dVS), and VMware Fault Tolerance (FT). Each of these technologies change the way the virtual infrastructure behaves and adds constraints to the virtual world. These constraints influence how a system is architected and implemented. We discuss data commingling (Classification Level Constraints on data) on the virtual network as well as the storage network. We also look at how DPM, FT, DRS, HA, and vMotion (storage or normal) affect security. Last, we make suggestions for securing the VMware Cluster.

Chapter 6: Deployment and Management

Many people do not consider the deployment and management of virtual machines to be much of a security issue, but it is. Because most, if not all, of VM deployment and physical-to-virtual conversions are done over the network, it is an important aspect to discuss in the context of virtualization security. Several threats to the VMware Virtual Infrastructure can target the specific management tools, whether they are vCenter Server (VC), the VMware vCenter Client (VIC), VMware vCenter Lab Manager, or even Web Access. Some vulnerabilities are easier to expose than others, but they do exist. In addition to the straightforward vulnerabilities, issues occur with authentication, roles and permissions, and access restrictions. Out of this discussion, we develop steps to protect your deployment and management environment.

Chapter 7: Operations and Security

Daily operations are affected by and have an effect on the security of your virtualization environment. Your business implementations may require you to expose a part of the environment in ways you do not currently understand, or you may restrict access so that daily operational tasks fail to run. In addition, operational tasks can overload a host, storage, or virtual network and cause things to appear to fail. Is the failure from a security issue or a normal operational issue gone bad? We discuss the most common operation issues and ways to protect and audit your environment while allowing the required access. This chapter also includes discussions of backups and performance tools that interact with the virtual environment.

Chapter 8: Virtual Machines and Security

VMware ESX and VMware ESXi run virtual machines. That is the main idea behind virtualization, so it behooves us to discuss security of the virtual machines. In this chapter, we discuss how virtual machines affect virtual infrastructure security and how virtualization affects VM guest security. We look at areas of overlap and information leakage, as well as how virtualization isolation changes the impact of the VM on security. In essence, although the guest security is left up to the guest, a VM's placement within the virtual infrastructure will impact the overall security of the infrastructure. VMsafe can and will change this impact. This chapter answers the question of whether a guest is more or less secure within a virtual environment. It leaves you with a list of steps to take to protect the VM from the environment and protect the environment from a threat within a VM or from a VM.

Chapter 9: Virtual Networking Security

This chapter delves into the virtual network using real-world questions brought up on the VMTN forums regarding security. We look specifically at the virtual networking concepts of multiple security zones, iSCSI Initiators within VMs, VLAN tagging, intrusion detection systems, and virtual firewalls, as well as how these concepts can be implemented securely in the VMware enterprise products. Last, we look into tools you can use to audit and monitor your network security and discuss how you would use other network security tools to, in effect, harden the virtual network.

Chapter 10: Virtual Desktop Security

Virtual Desktop Infrastructure (VDI) can be implemented in different ways, but which is the most secure? What are the caveats of using VDI? This chapter looks at a specific case of virtual desktop VMs and how they are made available to users. It also covers what is required to secure the environment as well as user data. What are the steps necessary to harden VDI and the Virtual Desktop Manager to create a secure environment? This chapter leaves you with the steps to secure VDI/VDM. In addition, this chapter was written by contributing author Tom Howarth; our thanks to Tom.

Chapter 11: Security and VMware ESX

You may think that all this was covered in the preceding chapters, but it was not entirely. We look at the various authentication methods available to the virtualization hosts, detailed steps to take to harden the hosts, as well as auditing and monitoring for compliance and security, patching, and the subtle changes that affect security. This chapter finishes with a checklist for you to follow as well as a discussion of security, compliance, and business policy that drives the checklist.

Chapter 12: Digital Forensics and Data Recovery

Virtualization security forensics is a growing field, and currently no hard and fast rules exist for investigating the full virtual environment. There are, however, certain steps and tools that can be used for digital forensic analysis of an individual VM, and we survey some of those tools. Some steps can be applied to the virtual host or cluster that aid in forensic analysis of an attack against a host. Outside of preparation, we look into Ulli Hankeln's Multiple Operating System

Administration (MOA) tool to be used for both forensics and data recovery. We end with some thoughts on what is needed in the digital forensic science.

Conclusion: Just the Beginning: The Future of Virtualization Security

We end the book with some thoughts about the future of virtualization security. Where will we go from here? What attacks, hacks, and cracks will be developed in the future or be applied in the future to virtualization? Where do we even find this information, and how should you go about getting a hold of it? This book is the beginning of our journey.

Appendix A: Patches to Bastille Tool

Appendix A provides patches to the Bastille-Linux tool referenced in Chapter 11.

Appendix B: Security Hardening Script

Appendix B provides a script that will further harden the VMware ESX service console so that it can pass the DISA STIG for VMware ESX, CISecurity ESX Benchmark, and the VMware VI 3 Hardening Guide as presented by ConfigCheck.

Appendix C: Assessment Script Output

Appendix C provides the full output of the security assessment tools used within this book when run against a virtualization host that has *not* been hardened.

Appendix D: Suggested Reading and Useful Links

Appendix D has a list of references created by the author, the contributing authors, as well as the book reviewers for further reading and information on the subjects of virtualization, security, and forensics.

Glossary

This element contains terms and definitions used throughout the book.

Chapter 6

Deployment and Management

Many people do not consider the deployment and management of virtual machines to be much of a security issue, but it is. Because most, if not all, VM deployment is done over the administrative network on which the VMware ESX Service Console, VMware ESXi Management Appliance, and vCenter Management Server (VC) hosts, it is an important aspect to discuss in the context of virtualization security. There are several threats to the VMware vSphere™ and Virtual Infrastructure that can target the specific management tools, whether they are vCenter Management Server, the Virtual Infrastructure Client (VIC), Lab Manager, or even webAccess. Some vulnerabilities are easier to exploit than others, but they do exist. In addition to the straightforward vulnerabilities, issues exist with authentication, roles and permissions, and access restrictions.

In addition to the normal tools that ship with the VMware ESX, ESXi, and Server hypervisors, we will branch our discussion to include the VMware Stage, Lab, and Life Cycle Managers. At this time we are not pulling into our discussion the VMware Virtual Desktop Manager (VDM) but we will discuss this in Chapter 10, "Virtual Desktop Security."

One of the first things to understand is how data flows among all the management tools within the virtual environment. In some cases, there are settings that will change how data flows, and will discuss those as well.

Management and Deployment Data Flow

Of chief interest when discussing security and management is how the management data flows around the virtual and physical network. Several management clients and tools are in use when we attempt to manage the virtual infrastructure, and they all have their own management methodologies and constraints. We will discuss all the primary VMware management products except the Virtual Desktop Manager and VMware View Manager within this chapter. Chapter 10 discusses the ins and outs of VDI including VDM and VMware View Manager.

All traffic from management clients to either virtualization hosts or VC and back is encrypted using the secure socket layer (SSL) with the exception of the initial handshake done to establish SSL connectivity. SSL allows for end-to-end encryption as defined in Chapter 2, "Holistic View from the Bottom Up." Also, as defined in Chapter 2, SSL is susceptible to a MiTM certificate injection attack. We discuss this further within this chapter.

Most of the mechanisms discussed in the following sections use SSL over port 443, and you may wonder how it can keep everything straight. How does the system know to send data to the SDK versus webAccess versus VIC access? VMware solved this problem with extensive use of reverse proxies based on the entry point into the VC, VMware ESX, VMware ESXi, or VMware Server hosts. Reverse proxies hide the destination port from the client, which also decreases the overall attack surface of exposed ports. Everything appears to tunnel through port 443. However, this does create a series of daemons within VC, VMware ESX, and VMware ESXi that could be listening on external ports yet do not need to do so.

When VMware ESX, ESXi and VC are installed, they create a set of self-signed certificates to enable SSL to be used. These self-signed certificates are based on a root certificate not within any normal browser, so they will generally trigger requests for approvals when they are used. The exception is when ESX speaks to VC; that is approved automatically.

VIC to VC (Including Plug-Ins)

The most commonly used management tool is the Virtual Infrastructure Client (VIC), which can either connect directly to a VMware ESX or ESXi host, or to the VMware VirtualCenter (VC) server. This key management tool behaves differently when connected to VC than the host. When connected to VC, data flows from the VIC to VC and then stops or heads on to the host. Initially the VIC will talk on

port 80, switch to port 443 for initial SSL setup, and then be switched to port 902 for all further communication.

Most traffic goes from VIC to VC, then either stops at VC or is sent on to the virtualization host. Those items that are global in nature do not always go on to the host. For example Datacenter, Cluster, Alarms, Tasks, and Permissions do not go on to the host. Yet, subsets of clusters, specifically VMware HA configurations, do go on to the host. Communication between VC and the host uses SSL over port 902.

Of particular interest is the case of using the remote console, because this path goes to VC to retrieve the host on which the VM resides and then goes directly to the host using SSL over port 902. Because the name of the VMware ESX or ESXi host comes from VC, name resolution related errors often occur when using the remote console from a workstation that does not have the DNS entries for the VMware ESX or ESXi host.

The VIC to VC data flow can be summarized as follows:

- VIC to VC for Overview pages (these do not use SSL).
- VIC to VC for Datacenter, Cluster, Alarms, Tasks, and Permissions settings using SSL
- Remote Console to Host setup using SSL
- VIC to VC to Host for all other actions using SSL

Figure 6.1 presents the data flow in a visual fashion.

The easiest way to get a handle on what goes where is to look at the list of permissions within the VIC when connected to VC. Why the permissions list? Because it is a very good breakdown of the functionality allowed for the VIC as Figure 6.2 depicts. Multiple VMware vCenters are shown within the diagram; these are all the same vCenter server. This format makes the diagram easier to understand because you can easily tell which data would be forwarded on to the virtualization hosts and the VMware Update Manager, as well as what would stay within the vCenter server.

The data flow is also affected by the role in use. The read-only role, for example, will not display data that is not already within the VC database. This seems counterintuitive because read-only access can also see real-time performance graphs. This is because VC is gathering the data in the background and storing this data within its own database. This happens regardless of the role currently logged in.

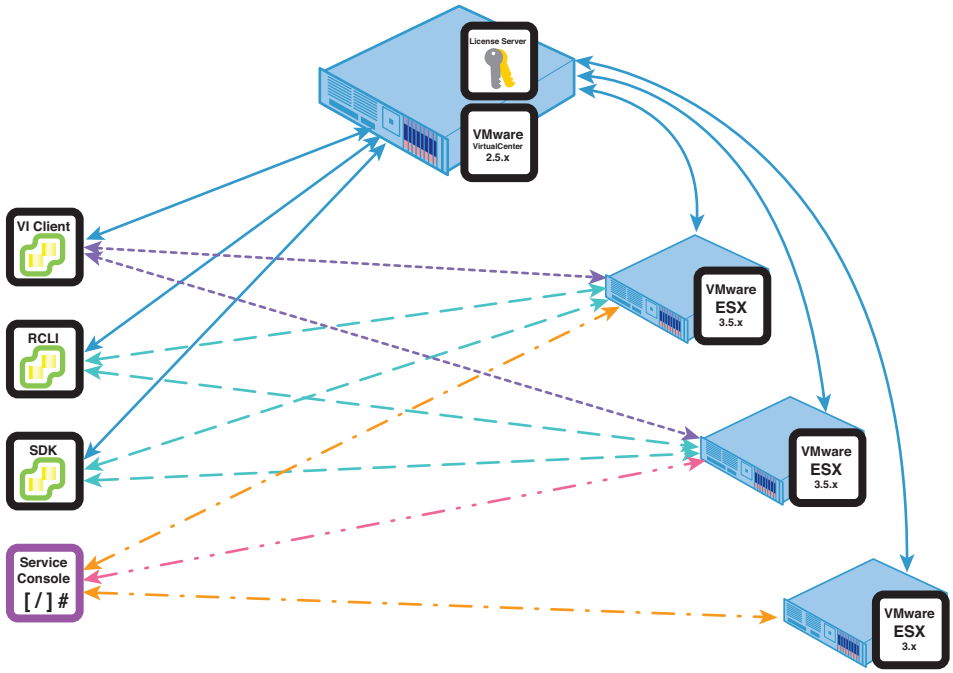

Figure 6.1 VIC to VC data flow

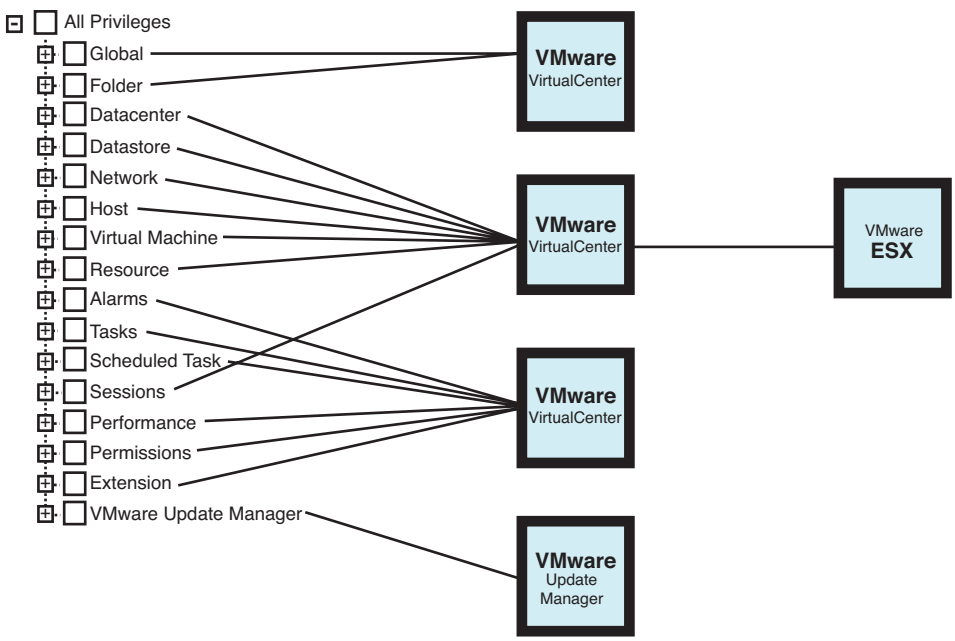

Figure 6.2 Breakdown of dataflow per permission

On the host, one important daemon is involved in all this processing, which initially listens on port 443 and then switches to port 902. Specifically, most traffic talks to the `vmware-hostd` daemon which is the `vmware-authd` service. Host Health Status information is gathered by VC via the `cimserver` or `Pegasus` using SSL port 902. The 5588 and 5589 ports are also open on the virtualization host for health status but are not used by VC or the VIC directly, yet these ports can be used by other tools.

Security Note

VIC to VC travels over port 80, switches to port 443 for SSL initiation, and then finally talks over port 902 using SSL.

Remote Console retrieves the hostname of where the VM resides using port 902 to VC, but then communicates using port 902 using SSL direct to the host.

All other VC to Host traffic is over SSL on port 902 but initiates on port 443.

SSL attacks can happen between VC and Host communication set up on port 902.

SSL attacks can happen between VIC and VC communication set up on port 443 and 902.

SSL attacks can happen between VIC and Host remote console communication set up over port 902.

The current VIC does not accept two-factor authentication directly; it relies entirely on the authentication of the workstation from which the VIC is running. There is a way to use this existing authentication as credentials for the VIC to VC connection and not require another challenge response platform, and that is to use the `-passthroughAuth` option for the VIC to provide a form of single-sign-on to VC.

VIC plug-ins either communicate with VC, go direct to a host, or go to other resources using as many communication paths as there are plug-ins. VC plug-ins bend all the rules about which ports to open and how communication happens over them. The plug-in generally runs within the context of the current credentials within VC and directly on the host. In other words, plug-ins do not necessarily require further authentication. We discuss plug-in issues later within this chapter.

VMware webAccess also does not support two-factor authentication but relies on the workstation from which the system was run to handle this aspect of authentication. VMware webAccess does not have a pass-through authentication mode, yet many single-sign-on tools are available, such as HP's ProtectTools Security Manager, which ships with most HP laptops today.

VIC to Host

The VIC to host data flow is a direct connection to the host over port 80, which is reversed proxy to port 443, which in turn talks SSL and eventually uses port 902 talking SSL for all future communication between the VIC and host. This is similar to how VC connects to the host in the VIC to VC discussion. There is no need to discuss this one in as much detail as we did in the previous discussion because everything talks to the host as shown in Figure 6.3.

If VC is employed, and changes are made directly on the host, VC will need to query the host for updates to its databases. However, this generally happens in the background. In some rare cases, or in cases where you need the information immediately within a VIC to VC connection, a VIC to VC connection can be forced to refresh its data through a manual refresh link that is on nearly every screen of the VIC. It should be noted that the roles and permissions used for VIC to VC are ignored, and the host specific roles and permissions are now used.

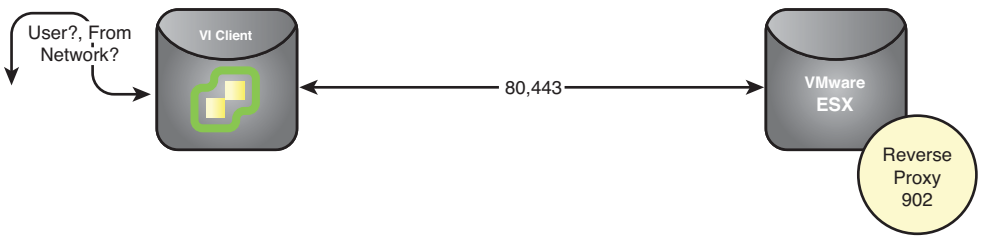

Figure 6.3 VIC to host data flow

Like the VIC to VC discussion, VIC to Host connectivity also does not have the capability of two-factor authentication. However, the same option discussed previously can provide a single-sign-on capability as long as the host is also participating in a directory service that the workstation is using. We discuss setting up directory services later in this chapter. This implies that single-sign-on will not work for VMware ESXi, because it is not possible to configure a directory service on this type of host. However, there is hope for VMware ESXi administrators with the Hy-Trust security appliance, which sits between your VMware ESX or ESXi hosts to apply additional granular credentials to all access whether from the VIC, VC, or other management tools.

VC webAccess

VC webAccess, shown in Figure 6.4, occurs over port 80 and then uses a reverse proxy to connect to port 8009, which speaks SSL. webAccess does not grant the same capabilities as the VIC with VC version 2.5.x, yet does provide a method to review and control individual VMs. Similar to the VIC to VC, further communication may occur from VC to the host to gather information about VMs over port 902. However, any remote console connection is direct to the host over port 902. webAccess is subject to the roles and permissions set up within VC.

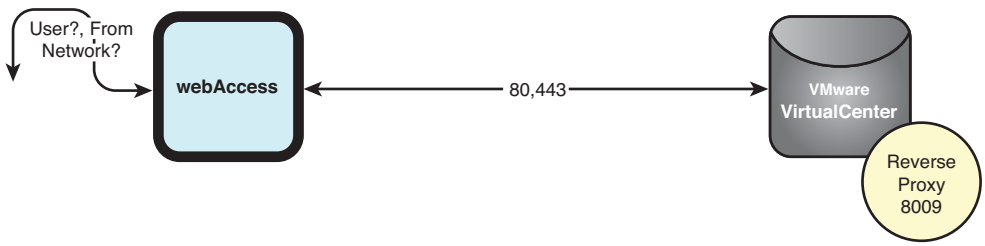

Figure 6.4 VC webAccess data flow

With VC webAccess there exists no support for two-factor authentication, yet single-sign-on is possible using Web browser add-ons or hardware devices within your workstation.

ESX(i) webAccess

ESX(i) webAccess works identically to VC webAccess except that there is a direct connection to the host to get information about each VM. Figure 6.5 depicts this data flow, which is subject to all the roles and permissions configured on the ESX(i) host and are unrelated to those configured on VC. ESX(i) webAccess occurs over port 80 and then switches to port 443, where it stays and speaks SSL.

With VMware ESX and ESXi webAccess, no support exists for two-factor authentication, yet single-sign-on is possible using Web browser add-ons or hardware devices within your workstation. One example of such a device is the previously mentioned HP ProtectTools Security Manager.

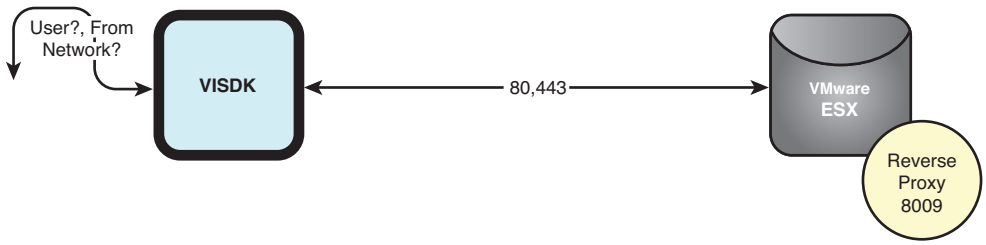

Figure 6.5 ESX webAccess data flow

VI SDK to VC

Similar to the VIC to VC discussion is the VI SDK to VC data flow in that some aspects of the VI SDK will stop at the VC server and others will go directly onto the ESX server using the normal VC to ESX data flow. However, the initial connection using the VI SDK to the VC server is done using the method of VC webAccess except that the target URL is https://virtualcenterserver/sdk instead of just http://virtualcenterserver. Specifically, the SDK will attach to port 443, which talks over SSL and then reverse proxy to port 8086.

The SDK is extremely complex and supports many bindings to its Simple Object Access Protocol (SOAP) interfaces. The Web Service Definition Language (WSDL) interfaces are the key components of the VI SDK. Everything that can be done within the VIC can be duplicated using the VI SDK. Figure 6.6 depicts this data flow, which is subject to all the roles and permissions that VIC to VC must obey.

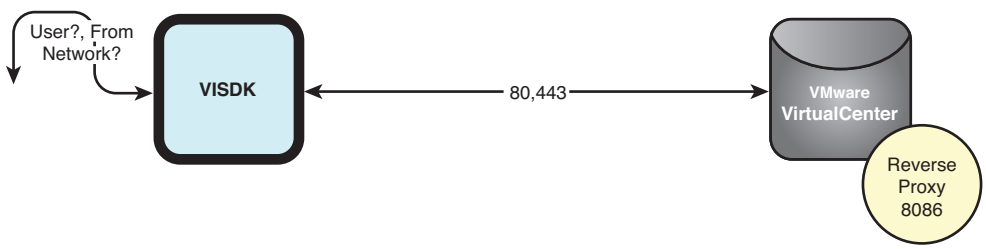

Figure 6.6 VI SDK to VC data flow

However, no mechanism exists to duplicate the remote console within the VI SDK. Yet you can create your remote console links that go directly to the host

using the following PERL VI Toolkit code. You can create multiple types of links, as well. In the following code, `$rcview` has a value of 33, which tells MKS to display the virtual machine's console as well as details such as event logs. A value of 9 tells MKS to display the virtual machine's console as well as the inventory panel. A value of 36 tells MKS to display the virtual machine's console only. A value of 12 tells MKS to add an inventory panel to the display of just the console.[1]

```
# Create MKS link partial code
my $rcview = 33;
my $vm_view = Vim::find_entity_views( view_type => 'VirtualMachine' );

for my $vm (@$vm_view) {
      my $name = $vm->name;
      my $moref= $vm->{'mo_ref'}->value;
      my $url =
qq{wsUrl=http://localhost/sdk&vmId=VirtualMachine¦${moref}&ui=${rcview}}
;
      my $rcurl = "https://${virtual_center}/ui/vmDirect.do?view=" .
encode_base64($url, q{}) . "_";
      print "$name $rcurl\n";
}
```

The developer of a VI SDK application should include the capability to use two-factor authentication or single-sign-on because the VI SDK does not provide this feature. Furthermore, the VI SDK does not currently verify the veracity of the SSL certificates in use, other than the typical self-signed check that occurs. Further verification of server certificates is required.

VI SDK to Host

Similar to the VI SDK to VC discussion, the VI SDK to host bypasses VC completely and can perform all the actions that the VIC connected to the host can perform. Figure 6.7 depicts this data flow, which is subject to the roles and permissions set on the host and not those set on VC.

1. http://communities.vmware.com/message/891004

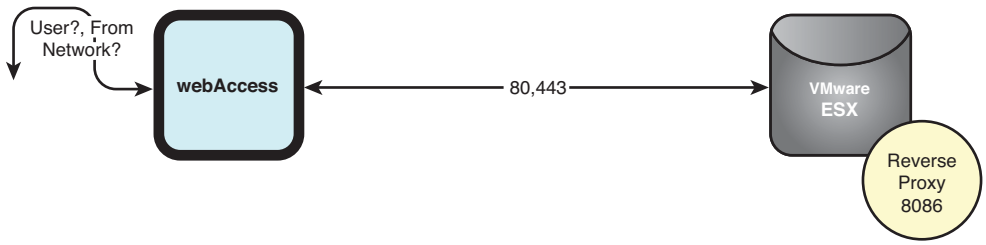

Figure 6.7 VI SDK to Host data flow

RCLI to Host

RCLI to Host is a combination of VI SDK to Host and VIC to Host activity defined previously. Some actions use the VI SDK explicitly, whereas others talk over port 902 to complete their tasks, as does the VIC. The RCLI, like the VI SDK, does not have the capability to support two-factor authentication or single-sign-on within its functionality.

RCLI to VC

Similar to RCLI to Host, RCLI to VC makes use of the VI SDK as well as the VIC to VC communication channels. VC to Host communication happens over port 902.

SSH to Host

SSH to Host is specifically for VMware ESX and not generally for VMware ESXi. By default, VMware ESX denies direct root access using SSH, which a very good setting to keep because it maintains the defense in depth available to the GNU/Linux service console environment. To enable SSH on VMware ESXi, you must first break the only security shield VMware ESXi really has, because there is no defense-in-depth capability within VMware ESXi. SSH talks over port 22 by default, but that can be changed as necessary.

SSH has the capability to create preshared keys for communication to enable single-sign-on. In addition, it is possible to implement two-factor authentication using SSH.

Console Access

Console access can be granted while at the keyboard in front of the host or via a remote access card such as the HP Integrated Lights Out (ILO) or Dell Remote

Access Card (DRAC) adapters. These provide network access directly to the console as if you were physically at the console. The tools generally talk over port 80 and switch to port 443 for SSL communication via a Web browser. Because a web browser is in use, single-sign-on is supported as described in the VC webAccess section. In addition, most remote access devices also support their own two-factor authentication.

For VMware ESX console, but not for VMware ESXi, it is possible to enable two-factor authentication using PAM modules provided by the vendors of these products like RSA.

Lab Manager

Lab Manager presents to the user its own interface, yet behind the scenes it communicates with VC using the VI SDK to VC path. Lab Manager can have its own authentication methods outside the ones normally used for VC or even ESX.

Site Manager

Site Manager, like Lab Manager, presents to the user its own interface, yet behind the scenes it communicates with VC using the VI SDK to VC path. Site Manager can have its own authentication methods outside the ones normally used for VC or even ESX.

LifeCycle Manager

LifeCycle Manager, like Lab Manager, presents to the user its own interface, yet behind the scenes it communicates with VC using the VI SDK to VC path. LifeCycle Manager can have its own authentication methods outside the ones normally used for VC or even ESX. VC to ESX communication happens over port 902.

AppSpeed

AppSpeed, like Lab Manager, presents to the user its own interface, yet behind the scenes it communicates with VC using the VI SDK to VC path. AppSpeed can have its own authentication methods outside the ones normally used for VC or even ESX.

CapacityIQ

CapacityIQ, like Lab Manager, presents to the user its own interface, yet behind the scenes it communicates with VC using the VI SDK to VC path. CapacityIQ can have its own authentication methods outside the ones normally used for VC or even ESX.

VMware Update Manager

VMware Update Manager (VUM) runs as a service to communicate to the VMware Patch repository on the `vmware.com` Web site. The update manager will download the patches and provide a local repository to use when patching your VMware ESX and ESXi hosts. VMware Update Manager uses the `esxupdate` function on each ESX server to connect to port 80, which then reverse proxies to the update manager server on port 8084. On the update manager host, `esxupdate` pulls down each patch in the necessary order and updates the host, rebooting as necessary. It is also possible to use an Internet facing host to download the VMware Update patches so that your VUM host does not need direct access to the Internet.

Management and Deployment Authentication

Now that you understand how the data flows around the management network, we should discuss how each of these data paths is authenticated. It is possible that there could be many authentication methods to gain access to the same set of data. If multiple paths exist to the same sets of data and multiple authentication and authorization paths into that data, then security issues could exist that are currently unknown. The most prevalent security issue is inadvertent information leakage. For example, if one user was denied access within the VIC to see a VMs data within VC but is then allowed to see this data within LifeCycle Manager, this could be a potential for information leakage and unauthorized access. The claim could be made that this information about the VM is trivial—the name of the VM and its virtual hardware makeup, among other things. However, if this VM is a classified VM, none of that information would normally be seen by anyone not within the appropriate classification. Therefore, this example would provide a breach in security because of information leakage.

This possibility is the main reason nearly all the current security guidelines dictate that some form of directory service be employed to control authentication and authorizations.

Difference Between Authorization and Authentication

There is a huge difference between authorization and authentication. Authentication is the act of confirming the identity of the user. Authorization is to what that identity has the right to access. In VC terms, authorization is the roles and permissions assigned to a user. Yet roles and permissions exist for VC and the VMware ESX or ESXi hosts. Roles and permissions do not translate to direct access to the management appliance through other means. Roles and permissions definitely do not expand to include all the other management tools. This often creates a split-brained authentication and authorization situation, even when directory services are in use, because several sets of roles and permissions are possibly in use. This leads to confusion at the very least!

Split-Brain Authentication

Split-brain authentication occurs when there is more than one method to authenticate a user to a given role. Each part of the VMware Virtual Environment has its own authentication method. Let's look at the special example of the administrative user. The administrative user for VMware vCenter is the user named `administrator` by default, whereas for VMware ESX and ESXi, the administrative user is named `root`. For VMware Server, `root` is used when VMware Server is hosted on a Linux system and `administrator` is used when VMware Server is hosted on a Microsoft Windows system. Each of the other management tools, such as LifeCycle Manager, uses a different default administrative user.

This causes quite a bit of confusion in most cases. Different usernames imply that first there needs to be a mapping between the users, which includes the authorizations for the user that we will discuss in the next section. The tool often used to mitigate this confusion is the use of a directory service like active directory, LDAP, NIS, eDirectory, and so on. This implies that all systems within the virtual environment should also use the same directory service. However, it is impossible to use the same directory service everywhere and even if you did, the directory service could not be used for any user for VMware ESXi or the root user for VMware ESX and VMware ESXi.

You may wonder why it is not possible to override the root user, as we can set any user within the directory service to be part of an administrative group and be granted the proper authorizations on Microsoft Windows systems. The answer is fourfold. First, root is just a name for a user with a user id of 0. Any user with a user id of 0 is in effect a root user. It is not possible to change the definition of userid 0. Second, if you are hosting your directory service within a VM and you

need to boot that VM, you may have to log in to the system as the root user to start the VM whether this is via SSH, using the VIC, or using the console. If you cannot authenticate the root user because its authentication is direct to the directory server, you cannot start the VM. Unlike Microsoft Windows, where credentials can be cached locally for direct login, with GNU/Linux the credentials cannot be cached, so a chicken and the egg situation may exist. Note, however, that the VIC and RCLI do not use cached credentials. The third reason is that with VMware ESXi version 3.5, it is impossible to set up a directory service on the system because the capability does not exist within this version of the product.

The fourth reason is that to give a user these privileges, you must in effect make the user root (with a user id of 0). This is frowned upon because no audit trail exists in this situation. Instead, we use other tools to gain this audit trail and remove multiple root accounts, which can increase the possible attack surface of the virtualization host. In addition, improperly set up directory service access can also lead to an increase in the possible attack surface of the virtualization host.

So there can be local accounts as well as directory service accounts, and at least one local account should be available in case the directory service authentication fails for some reason. There can also be users with local accounts as well as directory service accounts. This leads to multiple forms of authentication for a given user and therefore split-brain authentication.

Split-Brain Authorization

Of the two, split-brain authentication and authorization, the worse problem is split-brain authorization, or the capability for one system in the virtual environment to allow access to data that other systems do not allow. With so many management tools, it is quite possible that one of the tools will allow access to data that would not be available via another tool. I will highlight these issues with a few examples.

A user who knows an administrator user and password, but not necessarily the root password of a VMware ESX or VMware ESXi host, can directly attach the VIC to the host. By doing so they are presented with the capability of adding more users directly to the hosts. This capability is not normally available if you use the VIC directly connected to VC. However, note that unless the administrator makes a change within the permissions for VIC to VMware ESX or ESXi host, it is impossible to log in using a user that is not root. This use has two views of the same virtual environment, and one provides the chance for further abuse of the system by being able to create new users.

Security Note

Only the root user is allowed to log in directly to the ESX or ESXi host using the VIC initially.

Only the administrator user is allowed to log in directly to VC via the VIC initially.

VIC to VC and VIC to ESX display different tabs within the VIC and therefore different authorizations.

The Administrator role and permission within the VIC to ESX connectivity allows the user to modify different aspects of the host as compared to the Administrator role and permission when using VC.

Another example of split-brain authorization is a user who can approve virtual machines within LifeCycle manager; yet when the user logs in to vCenter, the user can also create virtual machines within the environment. Should the approver also be the creator of the virtual machines? If so, this can also lead to abuse because the approver has the rights to approve VMs in one system, which automatically create the VMs for the requestor. Yet the approver can bypass the approval stage and create his own VMs without a record of the VM being created. In this example, the user once more has two views for the same data and more privileges within one than the other system. The approver may need only read-only access to check on performance and other metrics but not need to be able to do anything else within the system.

When all management tools are in use, it is important to have very well-defined roles for all users and then assign the appropriate authorizations for each user across the virtual environment. This requires you to create a mapping from one management tool to another because not all tools have the same names for each role or permission.

Security Note

Create well-defined roles for each user.

Map all authorizations across all management tools for the user.

Grant only those authorizations to roles and assign the user to the role that fits the user's required authorizations.

Mitigating Split-Brain Authorization and Authentication

Several items must be implemented to mitigate split-brain authentication and authorization. First, use a directory service for all non-emergency users. Second, create well-defined roles for each user. Third, enable remote logging for auditing of possible violations of these roles. In some cases even the use of a directory service will not be allowed specifically for VMware ESXi v3.5 and Web applications that do not contain directory service integration. In addition, with VMware ESX, VMware ESXi version 3.5, and VMware Server, it is possible to have multiple local accounts that mimic administrator or domain user accounts that could use different credentials, even with a directory service in play. The use of a directory service can mitigate split-brain authentication for all but the emergency use accounts that should never be overridden. Yet, the emergency use accounts should still have an audit trail, so on VMware ESX, ESXi, and VMware Server hosts, this emergency account should never be the users `root` or `administrator` but some other account that can access the proper commands while providing an audit trail.

Several audit trails are available to virtualization servers, but for VMware ESX or Linux-based VMware Server hosts, the most widely used command to provide this audit trail is the `/usr/bin/sudo` command with its log file of issued commands within `/var/log/secure`. VMware ESXi does not have this command because direct access to the management appliance console is not a suggested practice. Use of the VIC or RCLI is the suggested practice, which leads us to an audit trail that is also available for VMware ESXi. This is the use of the `hostd` log file as an audit trail, which is located within the `/var/log/vmware` directory. This log file, as well as others, can be sent to a remote logging host using changes to the syslog daemon configuration. Use of a remote logging host is the recommended way of maintaining an audit trail because attackers have been known to remove their entries from the local log files to hide their tracks. A remote logging server presents yet another system to which they need access in order to hide their tracks. You may even go so far as to further firewall the logging server from your virtual environment to further prevent attacks. However, this is outside the scope of the book.

The preceding discussion on auditing is Linux-centric, but remote logging is also available for Microsoft Windows-based VMware Server hosts, as well as VMware VirtualCenter, and those Linux and Microsoft Windows based workstations where the VIC, RCLI, and VI SDK are used and launched. An audit trail should tell you when, where, who, and how the system was changed, as well as provide a way to repeat all actions and get the same result. There is quite a bit to setting up a proper audit trail within the virtual environment.

Security Note

Configure remote logging of user actions to a centralized log server.

Audit the logs on the centralized log server for authentication and authorization issues, among other security issues discussed throughout this book.

When using a directory service, limit local accounts on virtualization hosts to the local administrative user and an emergency use user who can access the administrative commands using an interface that can be audited.

The unfortunate truth is that log files are not generally small things, and in even a small environment you can very shortly have gathered gigabytes of data for review. No one has time to do this review by hand, so it is best to use some form of tool that will do this for you and notify you of issues that should be brought to your attention. I have used the Linux-based tool named `logcheck`, but there are many other tools to use. The key to using one of these tools is to train it to ignore those items that you have no need to view on a regular basis and those items that are purely informational. You also want it to have the capability to send email or other notifications based on the severity of the issue found. A host of these tools are available for both Linux and Microsoft Windows hosts.

It should be noted that such a tool is often required by Sarbanes-Oxley and other compliance standards to which many companies now adhere, so you may already have something in use within your environment.

Security Note

Log files should tell you who did what, when, from where, and how.

Log files should provide enough detail to allow you to duplicate the exact event that occurred.

Setting Up Microsoft Windows Systems for Remote Logging

Microsoft Windows does not have built-in methods to log items that appear within log files or the Event Viewer directly to a remote logging server. To perform this action, you need to use third-party tools. There are several from which to choose, however. When you are choosing a remote logging tool for Windows, you need to be sure that it can remotely log all entries that you see within the event viewer, as well as the specific text-based logs for VMware vCenter, Lab Manager, Site

Manager, and LifeCycle Manager. Another thing to consider for your Windows systems is what to log. A good place to start your search could be the open source Snare project at www.intersectalliance.com/projects/index.html. Snare provides the Snare Agent for Windows and Snare Epilog for logging general text log files to a remote syslog server.

Because there is no tool specific to the Microsoft Windows platform, the setup is outside the scope of this book. However, from where to log is not outside of the scope of this book.

You will want to remotely capture event and other text logs from your VC, Lab, Site, LifeCycle, and Update Manager hosts as well as from whichever Microsoft Windows workstation you launch the VIC, RCLI, or VI SDK applications. Depending on the application, you may also want to remotely log local log files.

It is also recommended that you put a logging wrapper around all access to the RCLI. This way, you will know what commands were issued with which arguments. Such a wrapper should not display passwords if any exist within the logs. The reason for creating this wrapper is that not all actions will be logged by the VMware ESX or ESXi hosts.

For VirtualCenter, enable remote logging of all files within `C:\Documents and Settings\All users\Application Data\VMware\VMware VirtualCenter\Logs`. This is a very good directory to watch for log files. Unfortunately, unlike the VMware ESX or Linux logs, the logs increase in number, so you may have to specify a directory when using remote logging with VC.

Setting Up VMware ESX for Remote Logging

Before I explain how to set up remote logging, we must first decide what to log remotely. On a VMware ESX or VMware ESXi running on a Linux host, there are a number of useful logs. Remote logging would encompass, minimally, the following log files.

```
/var/log/vmkernel
/var/log/secure
/var/log/messages
/var/log/vmware/*.log
/var/log/vmware/aam/{*.log,*/*.log}
/var/log/vmware/aam/{*.err,*.out,*/*.err,*/*.out}
/var/log/vmware/webAccess/*.log
/var/log/vmware/vpx/vpxa.log
/vmfs/volumes/*/*/vmware.log
```

To enable remote syslog support on your VMware ESX or host modify `/etc/syslog.conf` and add the following line to the file.

```
*.*                    @remotehost
```

Location of this line is unimportant. It would be best if the remote host could be resolved using the `/etc/hosts` file over DNS, just in case DNS is not available. This also alleviates ARP cache DNS style attacks for remote logging. Unfortunately the RCLI does not set this up for VMware ESX.

The difficult part is to now get all the log files that syslog does not know about to go over the wire to the remote log server. This is where the Snare Epilog tool for Linux comes in handy. There is no default mechanism in place to perform this type of logging shipping with VMware ESX.

Setting Up VMware ESXi for Remote Logging

For ESXi there are several means to enable remote logging. One is via the graphical interface and the other is by using the RCLI. For the RCLI, you use the following:

```
vicfg-syslog --server ESXiServerName --username root --password password --
setserver remotehost --port remotehostport
```

If the `remotehost` is running a version of `syslog` then the `--port remotehostport` option will not be necessary. Unfortunately, if the logs are not collected by syslog on an ESXi system, no current supported mechanism exists to get those logs to appear on a remote logging host because it is not currently simple to install third-party software that will survive a reboot when using ESXi. In addition, the third-party software must be specifically coded for ESXi. At the moment no such tools exist.

Directory Services

One of the major tools used to alleviate split-brain authentication and authorization is a directory service such as active directory (AD), lightweight directory access protocol secure (LDAP-S), Novell eDirectory, or network information services (NIS). However, for directory services to mitigate this possibility, it is important that all management tools and hosts involved also participate within directory services.

One notable exception to this is VMware ESXi, which does not support directory services natively. However, all the tools do support them as long as you are first connecting to vCenter and not a direct connection to the host. However, as we

all know, direct connection to the host is often required, specifically when we run the preceding command to control remote logging for VMware ESXi. So although use of a directory service mitigates many aspects of split-brain authentication and authorization, it is not a 100% surefire solution. There is no complete solution at the moment.

Configuring the VC to Start if Directory Services Are Not Available

In some cases when you virtualize your directory server, or when your directory server is not available, you will first have to boot your vCenter server. Normally you cannot do this if the directory service is not available. To mitigate this possible chicken and the egg situation, add the following lines near the end of the file `C:\Documents and Settings\All users\Application Data\VMware\VMware VirtualCenter\vpxd.cfg` prior to the closing `</config>`.

```
<security>
<ignoreUserResolveFailures>true</ignoreUserResolveFailures>
</security>
```

This change requires you to restart the VMware vCenter service for it to take effect. Look within the log files within the directory `C:\Documents and Settings\All users\Application Data\VMware\VMware VirtualCenter\Logs` to look for any errors after changing anything within the `vpxd.cfg` file, because it will report on any errors within the log file.

Setting Up Directory Services on VMware ESX

There are many articles on the use and configuration of directory services within VMware ESX and Linux-based VMware Server distributions. Although much is written about the way these services are configured, there is not much on how to secure them. Also, there are many levels of integration with directory services. Some require more manual maintenance than others.

The integration methods all lack the capability to translate user login restriction using a group policy object set within the directory service. Some of the more manual modes do not require this group policy translation, because you are forced to use local logins for each user you want to access the system. However, that can be quite laborious to maintain on more than a few systems, and it pretty much ignores one of the major strengths of using a directory service: control of authorizations as well as authentications. There are a few issues to clear up before we implement any directory service.

First, do not set up directory service authentication for any user with a user ID less than 500, because these are the system users, and they must remain untouched for the system to run properly. A user ID is a unique integer assigned to all users, and this number is used internally, not the name associated with the number such as we described previously about the root user. Second, never set up directory service authentication for the `root` user, because that is your emergency login in case directory service is broken for some reason. In addition, you should never log in directly as the `root` user unless it is an emergency. Third, if you have to create user accounts on a system to finish the directory service integration, those user accounts could be the source of an attack if directory services fail, because the passwords default to those set on the system.

The last issue is the most important reason why partial directory service integration is frowned upon. It is also why you need to set up access policies to deny those users who are not in the proper groups, regardless of authentication success. One simple way to mitigate this is to make sure the local users are not in any special groups and to allow access only if you are logging in using a specific group.

The quickest way to implement this is to use the `pam_access` module for VMware ESX authentication and authorization. To do this, you need to follow some very basic steps.

1. Add the following line to `/etc/pam.d/system-auth`.

   ```
   account [default=bad success=ok user_unknown=ignore]  /lib/security/
   $ISA/pam_access.so
   ```

2. Modify the file `/etc/security/access.conf` to reflect your group login policy, which will limit who can log in to only those within the given group. You can add the appropriate lines to the end of the file. Note this file is order dependent; you would not want to deny all access as the first line. Do not copy this verbatim; it is just an example explained afterward.

   ```
   +:root:crond console
   -:ALL EXCEPT root:vc/1
   +:GROUPNAME: NETWORK/NETMASK
   +:GROUPNAME: IP1 IP2 IP3
   -:ALL:ALL
   ```

In this example, we are denying root access to the `cron` daemon, `crond`, as well as the console. Next we are disallowing root access to all but virtual console 1

(vc/1), which is accessed using ALT+F1. Next we are allowing all users in the group GROUPNAME to log in as long as they are either on the network defined by NETWORK/NETMASK or from either of the IP addresses: IP1, IP2, or IP3. Last, we deny access to all other users and groups from all other locations. The /etc/security/access.conf file can be as complex or as simple as you desire. In general, and at a bare minimum, you will want to disallow logins unless they are coming from users within the appropriate group and from the appropriate network or IP addresses. For more detailed information, use man access.conf from any Linux system or your VMware ESX service console.

However, this works only for login style attempts. There are other ways to control what additional services a person can access from the network. Unfortunately, these other methods do not know about groups or users, so they are not a part of directory service integration and thus will not be discussed here.

Many VMware ESX installations do not allow installation of third-party packages from RPM repositories. If this is the case, your ability to integrate with directory services will be hampered It is possible to do so, but testing of the integration will suffer, as well as future problem determination. Going forward, be sure to test on a development box before applying to a production server.

Integration with NIS

For those who are *NIX centric, VMware ESX can integrate with NIS to provide directory service functionality. The steps to enable this functionality follow.

1. Use the following command from the service console (SC) command-line interface (CLI).

```
esxcfg-auth --enablenis --nisdomain=NISDOMAIN --nisserver=IPofNISServe
```

2. Modify /etc/nsswitch.conf to look like the following. The main changes are to add the keyword nis to the group and shadow lines, which may not be there by default but must be there for full integration.

```
# Autogenerated by esxcfg-auth
aliases:         files nisplus
automount:       files nisplus nis
bootparams:      nisplus [NOTFOUND=return] files
ethers:          files
group:           files nis
hosts:           files dns nis
netgroup:        nisplus
```

```
netmasks:       files
networks:       files
passwd:         files nis
protocols:      files nis
publickey:      nisplus
rpc:            files
services:       files nis
shadow:         files
```

3. Test to be sure everything shows up as expected. The following should show your normal password file contents plus any other users shared out by NIS. The `group` command will list your groups based on NIS as well.

```
getent passwd
getent group
```

4. Test to be sure NIS is working using NIS commands. The following commands will list the NIS specific users and groups. Note that if the third command does not return anything, then `netgroup` support does not exist on your NIS server. Investigate this with your NIS administrators because it will help with setting up the `/etc/security/access.conf` for the necessary `pam_access` configuration discussed previously.

```
ypcat passwd.byname
ypcat group.byname
ypcat netgroup
```

Partial Integration with Active Directory, LDAP, or LDAP-S

For those who do not implement NIS, other avenues exist for setting up VMware ESX hosts to use directory services. The three most popular are to use AD, LDAP, or LDAP-S. It is recommended that you never use LDAP for authentication because it is a clear-text or unsecured protocol. So we will not discuss this within this section. Partial integration implies that to complete the integration, you will need to manage user accounts per a VMware ESX host, and only the credentials and groups are handled via directory services. This is overall somewhat more secure than other methods, because if you do not have a login on the host there is no way to gain shell access. However, this has the drawback of requiring users to be maintained on all VMware ESX hosts. If you have more than a few hosts, this maintenance becomes a significant issue; you will need to remove accounts or add accounts when users leave or join the virtualization administration team.

Partial Integration with AD

Partial integration with AD requires the running of a simple set of commands. However, to test things you will need to add a new package to your system. This package is the krb5-workstation RPM, which you can find at any CentOS-3 or Red Hat Enterprise Linux 3 repository of packages. After you have it downloaded, install from the SC CLI using the following line. Note the version is relatively unimportant, so use the latest one you can find that came from the aforementioned repositories.

```
rpm -ivh krb5-workstation*.rpm
```

Another package to add is the pam_krb5 package from the same repository. This is not a requirement but will add better integration and protections to keep system accounts we discussed before from using AD authentication. You need version 2.11 of pam_krb5 to get the benefit of this capability for those emergency use accounts. Add the pam_krb5 package by doing the following:

```
rpm -Uvh pam_krb5*.rpm
```

The next step is to configure the VMware ESX firewall to allow AD and its components to speak with the host.

```
esxcfg-firewall -e activeDirectorKerberos
esxcfg-firewall -o 445,tcp,out,MicrosoftDS
esxcfg-firewall -o 445,udp,out,MicrosoftDS
esxcfg-firewall -o 389,tcp,out,LDAP
esxcfg-firewall -o 464,udp,out,kpasswd
esxcfg-firewall -o 464,tcp,out,kpasswd
```

Then enable AD authentication using the following, which will enable the VMWARELAB active directory domain using the domain controller `dc.vmwarelab.com`. It should be noted that you do not need to specify the domain controller if your domain can resolve within DNS.

```
esxcfg-auth --enablead --addomain=VMWARELAB --addc=dc.vmwarelab.com
```

Modify the `/etc/pam.d/system-auth` file so that it looks similar to the following. Note that only the highlighted lines need to be modified. One is to add in the pam_access line we discussed previously, and the other is to ensure that the emergency use users are not authenticated using AD, which could be broken during an emergency.

```
#%PAM-1.0
# Autogenerated by esxcfg-auth
account     required  /lib/security/$ISA/pam_unix.so broken_shadow
account     required  /lib/security/$ISA/pam_krb5.so
account [default=bad success=ok user_unknown=ignore]
➥/lib/security/$ISA/pam_access.so
auth       required   /lib/security/$ISA/pam_env.so
auth       sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
auth       sufficient /lib/security/$ISA/pam_krb5.so use_first_pass
➥minimum_uid=1000
auth       required   /lib/security/$ISA/pam_deny.so

password   required   /lib/security/$ISA/pam_cracklib.so retry=3
password   sufficient /lib/security/$ISA/pam_unix.so nullok use_authtok md5
➥shadow
password   sufficient /lib/security/$ISA/pam_krb5.so use_authtok
password   required   /lib/security/$ISA/pam_deny.so

session    required   /lib/security/$ISA/pam_limits.so
session    required   /lib/security/$ISA/pam_unix.so
session    optional   /lib/security/$ISA/pam_krb5.so
```

Now it is time to use the previously installed RPM programs to test the Kerberos connection that composes part of AD.

```
/usr/kerberos/bin/kinit Administrator
Password for Administrator@VMWARELAB.COM:
kinit(v5): Clock skew too great while getting initial credentials
```

If any errors occur like the one in the example, they need to be fixed. The one listed implies that the VMware ESX host is out of time sync with the domain controller. To fix properly, configure NTP on your VMware ESX host to match that used by your domain controller. Other errors require changing the encryption parameters used to establish the connection. To fix an encryption issue, edit the file `/etc/pam.d/krb5.conf` to look like the following with the appropriate changes for your domain. To fix an encryption problem we added the `default_tkt_enctypes` and `default_tgs_enctypes` lines to the existing file.

```
# Autogenerated by esxcfg-auth
[domain_realm]
vmwarelab = VMWARELAB
.vmwarelab = VMWARELAB

[libdefaults]
default_realm = VMWARELAB
default_tkt_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md5 rc4-hmac
default_tgs_enctypes = des3-hmac-sha1 des-cbc-crc des-cbc-md5 rc4-hmac

[realms]
VMWARELAB = {
     admin_server = dc.vmwarelab.com:464
     default_domain = dc.vmwarelab.com
     kdc = dc.vmwarelab.com:88
}
```

After `kinit` works, you have AD authentication available to you, and you just need to create and maintain local user accounts. AD uses LDAP to retrieve group and user information but uses Kerberos to retrieve authentication information. Also, note that the krb5-workstation package can now be removed because it is no longer needed, unless you want to keep it around to help solve AD integration problems. Remove using the following command:

```
rpm -e krb5-workstation
```

Partial Integration with LDAP over SSL or Secure LDAP

To use secure LDAP or LDAP over SSL you must follow the preceding steps for partial integration with AD. After you have that working, you need to modify the configuration to use secure LDAP. However, to do this we must first add another RPM package to the installation. This is the `cyrus-sasl-gssapi` package, and you can retrieve this from the same location you retrieved krb5-workstation. Install using the following line. Unlike the `krb5-workstation` package, you will not be able to remove this RPM when the integration is completed. If your company has concerns about third-party packages, this is not necessarily the integration you desire.

```
rpm -ivh cyrus-sasl-gssapi*rpm
```

Run the following command to add LDAP authentication to your existing AD integration.

```
esxcfg-auth --enableldapauth --ldapserver=vmwarelab.com --ldapbasedn=DC=vmwarelab,
DC=com
```

Next edit the `/etc/openldap/ldap.conf` file to look like the following using your base DN and LDAP server as appropriate. Even though we specified them in the preceding command, we should also double-check everything.

```
BASE dc=vmwarelab,dc=com
URI ldaps://vmwarelab.com:636 ldaps://vmwarelab.com:636
TLS_CACERT /etc/openldap/cacert.cer
SASL_SECPROPS maxssf=0
```

The last line is required when using secure LDAP with Kerberos, which is the configuration we are using. The certificate specified, `cacert.cer`, points to a file containing your exported root certificate and your consolidated certificates. This you would get from your certificate authority. If DNS is configured properly and you have multiple LDAP servers, DNS will handle which server to query. Note that you will want to ensure your DNS is configured properly and all names are resolvable going forward.

Like the previous section, we need to further configure the firewall to allow SSL-based LDAP queries to be made to the LDAP server.

```
esxcfg-firewall -o 636,tcp,out,LDAP over SSL
```

Export your root certificates from your CA in a base64 encoded X.509 file. If a certificate chain exists, ensure they are placed in one file. Place this file in the location specified using the `TLS_CACERT` variable within the `/etc/openldap/ldap.conf` file. On the LDAP server, create an account to which you will bind and use to search the directory from your VMware ESX host.

Using kinit as we did within the "Partial Integration with AD" section, we test the integration once more and fix any issues that show up.

```
/usr/kerberos/bin/kinit Administrator
Password for Administrator@VMWARELAB.COM:
kinit(v5): Clock skew too great while getting initial credentials
```

Last, test to be sure that the SSL connection works as expected.

```
openssl s_client -CAfile /etc/openldap/cacert.cer -connect vmwarelab.com:636
```

If the last line reads `"Verify return code: 0 (ok)"`, everything is set up properly and you can properly use LDAP over SSL. You now need to create and maintain local user accounts. You will authenticate using Kerberos and use LDAP over SSL to retrieve group and user information. Also, note that the krb5-workstation package can now be removed because it is no longer needed, unless you would like to use it to solve integration problems. One way to create and maintain local user accounts is to query the LDAP or LDAP over SSL server periodically using the following script from your VMware ESX service console. This script is reprinted here with permission from its author, Steve Beaver.

```
#########################################################################
#!/bin/bash
# Secure LDAP Search Script to add and remove users based on Group
Membership

# Stephen Beaver

#########################################################################
# variables
base="-b DC=domain,DC=com"    # Replace with your domain name
# This is the user that we will bind to LDAP with
user="-D MyLDAPUser@domain,com" # or can be in the form of
# user="-D CN=MyLDAPUser, OU=OU, DC=Domain, DC=COM
pass="-w password"     # The LDAP user password
group="ESX_ADMIN"  # The directory group you will search for
esxgroup="ESX-Admin" # The ESX group you would like the users to be a
member off
programdir="/usr/LDAP"                 # The directory this script will run
from
# More Variables that do not need to be edited
cmd="ldapsearch -Y GSSAPI -LLL"
pipe="-u -tt -T ${programdir}"
pipe2="-u -tt -T ${programdir}/Member"
filter2="CN=${Group} member"
```

```
filtersam="samAccountName"
########################################################################
# Get Kerberos Ticket
echo password ¦ /usr/kerberos/bin/kinit -V $user
# Sanity Check to make sure all the files and folders needed are in
place or create them
if test ! -x "$programdir" ; then
    mkdir $programdir
    mkdir $programdir/Member
    mkdir $programdir/Member/New
    mkdir $programdir/Member/Old
    echo > $programdir/Member/New/$Group.txt
    echo > $programdir/Member/Old/$Group.txt
fi
#################             NEW SEARCH             #################
# The first search to find the group and see who if any are members
LDAP_search ()
{
    ${cmd} ${base} ${user} ${pass} ${pipe} ${filter1}
    if [ "$?" -ne "0" ]; then
        printf "ERROR running LDAP Search script exiting"
        return
    fi
    LDAP_search_member
}

# Now that I have a temp file for each user. I need to collect and list
in a file to read from
# If I find no users in the group then no need to continue. Return and
move on

LDAP_search_member ()
{
    cd $programdir
    ls -1 $programdir/ldapsearch-member-* > $programdir/filelist.txt
    if [ "$?" -ne "0" ]; then
```

```
        printf "No Members moving on...  "
        return
    fi
    declare LINE
    declare MEMBER
    cat $programdir/filelist.txt ¦
        while read abc
            do case $abc in
            Member) echo $abc ;;
            *) awk '{print $0}' $abc >> $programdir/ulist.txt ;;
            esac
        done
    sed 's/,OU=.*//g' $programdir/ulist.txt > $programdir/mlist.txt
    sed 's/CN=//g' $programdir/mlist.txt >
$programdir/Member/filelist.txt
    LDAP_search_sam
}

# Now I have a list in a usable format. Time to search again to get the
samAccountName # or userid of each user in the group.

LDAP_search_sam ()
{
    rm -R $programdir/ldapsearch*
     rm -R $programdir/filelist.txt
     rm -R $programdir/ulist.txt
     rm -R $programdir/mlist.txt
     mv -f $programdir/Member/New/$Group.txt
$programdir/Member/Old/$Group.txt
    LDAP_search_create
}

# Now that I have a temp file for each user. I need to collect and list
in a file to read from
# Sort the list and compare the old with the new to see if I need to add
or remove users
```

```
# The useradd command below to add the user

LDAP_search_create ()
{
    cd $programdir/Member
    awk '{print $0}' $programdir/Member/filelist.txt ¦ tr [:upper:]
[:lower:] >> $programdir/Member/$Group.txt
     rm -R $programdir/Member/filelist.txt
     mv -f $programdir/Member/$Group.txt
$programdir/Member/New/$Group.txt
    sort -f -o $programdir/Member/New/$Group.txt
$programdir/Member/New/$Group.txt
        comm -1 -3 $programdir/Member/New/$Group.txt
$programdir/Member/Old/$Group.txt > $programdir/remuser.txt
        comm -2 -3 $programdir/Member/New/$Group.txt
$programdir/Member/Old/$Group.txt > $programdir/adduser.txt
    cat $programdir/remuser.txt ¦
        while read oldlist
            do userdel -r $oldlist
        done
    rm -R $programdir/remuser.txt
    cat $programdir/adduser.txt ¦
        while read newlist
            do useradd -M -g ESX-Admin $newlist
            /usr/bin/chage -M 99999 $newlist
        done
    rm -R $programdir/adduser.txt
}

#########  This section is the main body which calls all the functions
listed above
LDAP_search
exit
```

Full Integration with AD

Full integration with AD implies that you manage all the users directly from your AD server. In other words, there will be no need to create local accounts for administrators to be able to access the VMware ESX host. However, without pam_access implemented, as discussed previously, this is not a secure option, because AD users could log in to your VMware ESX service console if they have the network access to the box. So it is very important to configure pam_access properly.

> **Security Note**
> Implement `pam_access` to control who can log in to your hosts and from where.

Full integration starts at the end of the "Partial Integration with AD" section discussed previously, but instead of creating users, we will add functionality so that the users will no longer need to be created.

The first thing we do is modify `/etc/pam.d/system-auth` once more to change the following line:

```
account     required  /lib/security/$ISA/pam_krb5.so
```

Create this new line, where the default `required` keyword has been modified to look like the following line.

```
account     [default=bad success=ok user_unknown=ignore]
/lib/security/$ISA/pam_krb5.so
```

After that is completed, add one more line to the end of `/etc/pam.d/system-auth` to allow home directories to be created when users log in. This removes the need for extra management.

```
session      required         /lib/security/$ISA/pam_mkhomedir.so skel=/etc/skel
➥umask=0077
```

Now we need to add a few more RPM packages from the ESX media: namely the `samba-server` package. We will not be running the entire Samba server but only a small part of it, because enabling the Samba server opens up the ESX host for possible SMB/CIFS attacks.

```
rpm -ivh samba-server*rpm
```

In general, if you are allowed to do so you will want to update all Samba packages to a minimum of v3.0.25 to alleviate the need to make any changes to your AD server to lower its security stance as the lack of encryption could lead to a MiTM vulnerability. The changes you may have to make are to the AD servers' local security policies to disable the following options.

```
Domain member: Digitally encrypt or sign secure channel data (always)
Microsoft network server: Digitally sign communications (always)
```

Next modify `/etc/samba/smb.conf` to look like the following so that the `winbind` daemon can be used to query authorization and credential information from the AD server.

```
[global]
    workgroup = VMWARELAB
    server string = Samba Server
    printcap name = /etc/printcap
    load printers = no
    cups options = raw
    log file = /var/log/samba/%m.log
    max log size = 50
    security = ads
    socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
    dns proxy = no
    idmap uid = 16777216-33554431
    idmap gid = 16777216-33554431
    template shell = /bin/bash
    template homedir = /home/%D/%U
    winbind use default domain = yes
    password server = dc.vmwarelab.com
    realm = VMWARELAB.COM
```

The `template_homedir` option will be used in conjunction with the `pam_mkhomedir` module added previously to the end of the `/etc/pam.d/system-auth` file. To complete this, we need to create a directory and set up permissions to allow the creation of the user directories and assigning the proper permissions automatically. If user directories are not set up, the user will be placed within the top level directory on

log in. This is an undesirable result. The `template_homedir` option in `/etc/samba/smb.conf` contains two variables, %D, which refers to the domain to use, in our example VMWARELAB, and the %U variable, which refers to the user directory to create. The following commands create the directory for the domain and set the permissions of the directory so that when users create files within it, they assume the ownership of the user creating them and not the `root` user.

```
mkdir /home/VMWARELAB
chmod 1777 /home/VMWARELAB
mkdir /var/log/samba
```

Before we join the VMware ESX host to the domain, we need to modify the `/etc/nsswitch.conf` file so that queries for users and groups go through winbind instead of just querying the local files. Add the winbind keyword to the highlighted lines per the following example.

```
# Autogenerated by esxcfg-auth
aliases:         files nisplus
automount:       files nisplus
bootparams:      nisplus [NOTFOUND=return] files
ethers:          files
group:           files winbind
hosts:           files dns
netgroup:        nisplus
netmasks:        files
networks:        files
passwd:          files winbind
protocols:       files
publickey:       nisplus
rpc:             files
services:        files
shadow:          files winbind
```

Now we are ready to join the VMware ESX host to the domain. If the `kinit` test outlined in the previous section "Partial Integration with AD" passed, it is safe to add the host to the domain using the following commands. They will not only join the host to the domain but start winbind and enable it to start on reboot of the host.

```
net ads join -UAdministrator
 Administrator's password:
 Using short domain name -- VMWARELAB
 Joined 'HOST' to realm 'VMWARELAB.COM'
service winbind start
chkconfg winbind on
```

Winbind is an important aspect of this type of integration because it will speak to the AD server in an encrypted fashion and is the only part of the Samba server package that we will be using. At no point should the `smb` daemon be started or need to be started. Now it is time to test our configuration using a winbind tool named `wbinfo`.

Verify that the groups are picked up from the AD server.

```
wbinfo -g
 domain computers
 domain controllers
 schema admins
 enterprise admins
 domain admins
 domain users
 domain guests
 group policy creator owners
```

Verify that users are picked up from the AD server.

```
wbinfo -u
 administrator
 guest
 support_388945a0
 krbtgt
 testauser
 smbservice
```

Verify that trusted secret via RPC calls succeed. Note and fix any errors.

```
wbinfo -t
```

Verify that VMware ESX sees the AD groups properly.

```
getent group
 root:x:0:root
 ...
 domain computers:*:16777220:
 domain controllers:*:16777218:
 schema admins:*:16777222:administrator
 enterprise admins:*:16777223:administrator
 domain admins:*:16777219:administrator
 domain users:*:16777216:
 domain guests:*:16777217:
 group policy creator owners:*:16777224:administrator
```

Verify that VMware ESX can resolve the AD users. Note that this could list hosts as well as users, depending on how the organizational unit was set up within AD. Note that the part of the command return should be the path we set up to be used by the `pam_mkhomedir` module.

```
getent passwd
 root:x:0:0:root:/root:/bin/bash
 ...

administrator:*:16777216:16777216:Administrator:/home/VMWARELAB/administ
rator:/bin/bash
 guest:*:16777217:16777217:Guest:/home/VMWARELAB/guest:/bin/bash
 ...
 krbtgt:*:16777220:16777216:krbtgt:/home/VMWARELAB/krbtgt:/bin/bash
```

Verify that an AD user picks up the proper user ID and group ID for a specific user as well as the complete group list associated with the user. Compare the results to the results found in the previous test. It should also be noted that sometimes AD integration has issues if a user belongs to too many groups. With the latest versions of Samba, this should no longer be the case.

```
id testauser
 uid=16777221(testauser) gid=16777216(domain users) groups=16777216(domain users)
```

Now we have full integration with AD, which does not require user management on all your ESX hosts. Occasionally, you will want to go through and remove the deleted users from the `/home/%D` directory we created previously. This could easily be accomplished with a simple script to query the AD server and remove any directories for the users that do not exist within the directory service. This script follows and could be run from within the `cron` daemon at least once per day.

```
wbinfo -u > /tmp/wbinfo.$$
for x in `ls /home/VMWARELAB`
do
      u=`basename $x`
      grep "^$u$" /tmp/wbinfo.$$ >& /dev/null
      if [ $? -gt 0]
      then
            rm -rf /home/VMWARELAB/$u
      fi
done
rm -rf /tmp/wbinfo.$$
```

If there are reasons you would not use winbind, you can also configure LDAP or SSL over LDAP to query the same information using different tools but the same approach.

Setting Up Directory Services on Other Management Hosts

It is also very important to set up directory services on all the workstations and hosts in use by other aspects of the virtual environment. This includes but is not limited to backup servers, those workstations running the virtual machine management tools, the Virtual Infrastructure Management Appliance (VIMA) from VMware if it is in use, hosts for LifeCycle, Lab, Site, and Update Managers, as well as any hosts used for monitoring the virtual environment.

This is where those well-defined user roles come into play; you must maintain the same view of all data across a multitude of hosts, virtual machines, and possibly appliances.

Lifecycle manager will use directory services within itself, yet Lab and Site manager tools will pick up roles and permissions from vCenter. Update manager uses different authentication as well.

Setting up directory services on these hosts is outside the scope of the book, but nonetheless should be done to maintain auditing across the entire management spectrum on a remote logging host.

Security Note

Maintain a well-defined set of roles and permissions across all management tools.

Maintain directory services across the management hosts.

Maintain logging of all management hosts to a remote logging server.

Perform periodic audits of the logs on the remote logging server.

Use log server tools that will spot inconsistencies and warn the appropriate people in a timely fashion.

No directory services are possible for VMware ESXi, so use VirtualCenter whenever possible.

Security of Management and Deployment Network

We have discussed how to enable directory services and the need for remote logging of data from all systems on the management and deployment network, and in Chapter 9, "Virtual Networking Security" we discuss the networking constraints of this network. However, it is important to also maintain good encryption using the tools within this network. Many use SSL to pass data back and forth while others use clear-text protocols.

Using SSL

As discussed in Chapter 2, SSL is susceptible to MiTM attacks using certificate injection because the client blindly accepts the certificate given to it. In general, the certificate, if it is a self-signed certificate with a root certificate authority not registered within the system, is checked by a human, and humans are notorious for just wanting to get their job done regardless of security. Part of the use of SSL is to educate users on the features of a good certificate so that they do not make this common mistake.

The other option is to use a set of certificates that you control and maintain based on well-known root certificate authorities. To do this we need to replace the certificates created on the installation of the components of the virtual environment. We will replace these certificates with the ones you received from your certificate authority.

Using Certificates

Certificates contain information about the server to which you are going to connect. This information should be verified either programmatically or by your own eyes and information. How to verify certificates is outside the scope of this book. After the certificate is verified, it is then available for use. There is a second part to the certificate for the server component, and that is a private key to encrypt the traffic. Each certificate must be in the form of a base64 encoded X.509 file, commonly known as a base64 encoded PEM file because of its extension. For more information on these formats, review Appendix D, "Suggested Reading and Useful Links." Also refer to Appendix D if you are unfamiliar with the role of certificates, how they are created, from where to receive them, or how SSL works in general.

Certificate Authority

As explained in Chapter 2, the certificate authority is the important aspect of the securing of data via SSL. If you do not know or trust the certificate authority, how can you trust that there is no man in the middle or that the certificate is not a weak certificate with a well-known key? Well-known certificate authorities such as Verisign and RSA are already well known by all security principals involved. The security principals will be the Web servers, applications, and browsers we will be using to serve up data and access this data. However a self-signed certificate can also be used. A self-signed certificate is one that is created by a certificate authority that is not known. For example, you may get a certificate from a vendor using its own certificate authority with its own root certificate unrelated to the known authorities.

Self-Signed Certificates

Self-signed certificates are not necessarily insecure certificates. However, their use is sort of claiming you are who you are just because you said so. Yet VMware uses self-signed certificates by default. Most of the tools will detect if a certificate is in use, and if they cannot determine the root certificate, they will consider the certificate to be self signed and ask for human intervention to determine if this is acceptable. If you trust the certificate authority, whether it is self-signed or not, the certificate is valid. Some companies use self-signed certificates internally.

The tools that do not ask for human intervention are the VI SDK, RCLI, and VC to ESX connections. These tools do not present to the user a chance to review the certificate and do not programmatically verify much of anything.

Security Note

All certificates are susceptible to MiTM attacks, not just those that are self-signed.

The client must verify the certificate from the server, and the server must verify the certificate from the client.

This verification process is more cumbersome than just checking for self-signed certificates.

This process should not involve humans.

Replacing Certificates

Many companies, namely several government agencies, require the replacement of the default certificates with the ones given to them by their certificate authority. How to create a certificate authority is outside the scope of this section, but given a certificate and the public key for the certificate, we will show how to replace the certificates. Because multiple components exist, we will go through each one. It should be noted that we are replacing the certificates on servers and not those for clients, because VMware management tools do not currently support preshared certificates and keys, which is the only real way to prevent a MiTM attack.

Replacing VC Certificates

Replacing the VC certificates will be a very good start if you are using this tool within your environment. If not, I would ignore this particular section. The steps are very straightforward.

First, back up your old certificates, which are located in `C:\Documents and Settings\All Users\Application Data\VMware\VMware VirtualCenter\SSL`. The two files to back up are `rui.crt` (the certificate) and `rui.key` (the private key). A backup can easily be created by renaming these files.

The next step is to upload the new key and certificate files to the directory `C:\Documents and Settings\All Users\Application Data\VMware\VMware VirtualCenter\SSL`. Name the new key file to be `rui.key` and the new certificate to be `rui.crt`. These names are important to maintain. Also, make sure that the `rui.key` and your backup of the previous file grant read access to only the System User and the Administrators group. Because this is the private key for the entire vCenter server, it is important to protect appropriately by denying read access by anyone not a system user or within the Administrators group.

Security Note

Change the permissions on the `rui.key` file to be read-only by the system user and the Administrators group only.

This should be done whether you are replacing the certificate or not.

The last step is to restart the vCenter service on the host. Replacing virtualization host certificates is the second phase.

Replacing VMware ESX Certificates

The VMware ESX certificates are composed of two files, a key and the base64 encoded X.509 or PEM file (in the following, the PEM file is the `rui.crt` file).

First back up your old certificates:

```
cd /etc/vmware/ssl; cp rui.key rui.key.orig; cp rui.crt rui.crt.orig
```

Next, upload the new key and certificate files using your favorite secure copy tool. Place them in `/etc/vmware/ssl` and name them `rui.key` for the key file and `rui.crt` for the certificate file.

Modify the permissions of the `rui.key` and `rui.key.orig` file so that they are readable only by the `root` user.

```
/bin/chmod 600 rui.key rui.key.orig
```

Now restart the `hostd` and `webAccess` daemons to use the new certificates.

```
service mgmt-vmware restart; service vmware-webAccess restart
```

Replacing VMware ESXi Certificates

The VMware ESXi certificates are composed of two files: a key and the base64 encoded X.509 or PEM file.

First, back up your old certificates using the RCLI supplying the appropriate hostname and username to the following commands:

```
vifs --server hostname --username username --get rui.crt rui.crt.orig
vifs --server hostname --username username --get rui.key rui.key.orig
```

Next, upload the new key and certificate files using your favorite secure copy tool. Place them in `/etc/vmware/ssl` and name them `rui.key` for the key file and `rui.crt` for the certificate file.

```
vifs --server hostname --username username --put ssl_crt rui.crt
vifs --server hostname --username username --put ssl_key rui.key
```

Now use the Restart Management Agents option available through the local console, as shown in Figure 6.8.

Figure 6.8 ESXi Restart Management Agents menu option

Mitigating SSL MiTM Attacks

The only true way to mitigate a MiTM attack for SSL is to use preshared keys and certificates between the clients and the servers. Unfortunately, this is not possible at the moment because it is not supported within VMware. So that leads us to other options to use preshared keys to alleviate MiTM attacks. None of these options, however, addresses the communication between VC and the VMware ESX or ESXi host. It is extremely important that this communication be behind an administrative firewall.

Security Note

Place the vCenter Server on the same side of the firewall in which the VMware ESX or ESXi hosts live.

It is possible for the administrative clients to live on the other side of the firewall, as long as a secure mechanism exists to access VC or the hosts. Preferably, that secure mechanism would be over some form of tunnel. However, because that is not always possible, it is also recommended that the management clients be on the same side of the firewall as the VC server and hosts and that a secure mechanism is used to access the management workstation. VMware has created the VMware Infrastructure Management Appliance to aid in this endeavor. I use an internal VPN built in to my firewall between the production and administrative servers to secure this aspect of my management network.

It is possible to configure VMware vCenter to verify ESX host certificates by using the following within the VIC, use the menu option Administration -> VirtualCenter Management Center Configuration, select the SSL Settings link and check the "Check host certificates" checkbox and click OK.

Using IPsec

One option is the use of IPsec in main mode with preshared certificates to enable an encrypted communication between the management clients and the servers and hosts involved in the work to be done. Often, IPsec implementations switch to aggressive mode when preshared certificates are in use. However, this does require quite a bit of work to configure. Namely, you need to set up a Public Key Infrastructure (PKI) configured for IPsec, including the use of preshared certificates. Without preshared certificates, IPsec is no more secure than SSL. This option will work for clients connecting to VC but not VMware ESX or ESXi.

Security Note

When using IPsec, use preshared keys.

The IPsec option works only between clients and the vCenter server using IPsec supported by the vCenter server operating system.

Using Tunnels

Another option is to use various tunneling or VPN tools that are not IPsec. One of the most commonly used tunnel tools is secure shell (SSH). SSH allows the creation of an encrypted tunnel between one host and another. It does this by

allowing you to send packets to various ports on the local host, and then sending these packets over the tunnel to the remote host. As long as you use a preshared key created on the client and stored on the host, this type of tunnel is also secure. The creation of the preshared key differs between each OS. The most common tool to create this is the ssh-keygen tool. The resultant file's contents must be placed within the appropriate file on the host. This will work for VC as well as VMware ESX hosts. Unfortunately, it will not work very well for VMware ESXi because SSH is not enabled, nor should it be enabled.

Security Note

When using SSH tunnels, use preshared keys.

SSH works between clients and VMware vCenter as well as clients and VMware ESX.

SSH does not work between clients and VMware ESXi.

Using Deployment Servers

One of the largest insecurities is the physical to virtual (P2V) conversion and deployment. In many cases, a P2V will cross security zones—that is, from a production to a virtualization administration network—or even worse, from a DMZ to a virtualization administration network. The solution to this problem is quite simple. Use a P2V helper machine that has access to enough disk space to convert the entire machine. One such use case is a laptop that has been blessed by the security administrators to work within the target security zone—for example, the DMZ. The laptop would contain the VMware Converter tool and enough external storage (USB, eSATA, FireWire, and so on) storage to contain the entire virtual machine.

After the conversion takes place, detach the USB or FireWire storage and move it to the virtual environment management security zone. From there you would again power up the converter and import the VM into the appropriate virtualization host with the proper network connections to run within the appropriate security zone that was already prepared within the host—in our example, a DMZ.

This approach has several advantages: first, there is no network involved that would cross security zones, and second, because the resultant VM will not be powered on until it is back within the appropriate security zone, there is no chance of contamination from one security zone to another. Contamination could be considered a change in DHCP server, registration in the wrong DNS server, or the inadvertent introduction of a virus, worm, or other nasty things.

Security Note

When converting physical to virtual across security zones, use a P2V helper machine that has been blessed to run within the source security zone.

Use removable media to move the resultant VM to the virtualization management network for import into the virtual environment.

Do not power on the VM until it is again on the proper network.

The other option to consider is the use of a deployment server to which you would deploy all VMs before moving them to the appropriate virtualization hosts. The deployment server would have all the required networks that are within the virtual environment but would not be connected to them, except through virtual firewalls that disallow direct access to the new VMs. This gives you the capability to properly patch VMs in a safe and secure environment before you migrate them to the true production network using either cold migration, VMware VMotion, or SVMotion.

In this fashion, the systems can be deployed within an extremely secure virtual network without the risk of zero day or other attacks. The virtual firewalls involved would disallow all access to the VMs but allow the VMs to access AD, DNS, DHCP, and other necessary servers and services to complete any patches, updates, and software as required.

This deployment method provides the added advantage of being able to test the VM before final migration.

Security Note

All VM deployment and physical-to-virtual conversion occurs over the management network.

The first boot of the VM should be in a safe, secure environment that will allow the necessary patching and updates to the VM.

Security Issues during Management and Deployment

A set of common problems that occur when working with VMs could be considered security issues as well. These range from simple mistakes made by

administrators to serious issues of data crossing security zones. This section covers a few of the more prevalent issues.

VIC Plug-ins

Several Virtual Infrastructure Client plug-ins are available. These third-party tools allow access to otherwise inaccessible data. Some plug-ins use the authentication contained within the VIC after you log in and start your session, whereas others access other authentication methods. Either way, the plug-ins in use should be used with extreme care because there is no inherent protection from a plug-in within the VIC other than the capability to disable a plug-in, if it provides you the option. Not all plug-ins provide this capability. After you log in, it has the access to the virtual environment that you do. One of the plug-ins available logs all actions within the VIC. If this was done behind the scenes and sent to a hacker, the hacker would now have enough information to possibly pivot an attack into your virtual environment.

Table 6.1 provides a partial list of the currently known plug-ins, what they do, and the possible risks.

Table 6.1

Partial List of Available Plug-ins

Plug-in Name	Function	Security Issues
Andrew Kutz's SVMotion Plugin	Interface to Storage VMotion	Superseded by VMware's own Migrate Storage option within the VIC. Andrew was the first to reverse engineer the plug-in functionality.
Chat	Embeds a Meebo Room chat into the VIC	Meebo Rooms is an offsite chat system. Most companies do not allow any form of unauthorized chat programs.
Console	Embeds access to the VMware ESX service console via SSH	This plug-in requires you to enter login credentials, which can be for any user, which could be a way to elevate privileges.
Invoke	Allows you to invoke third-party commands from within the VI client using an existing, authenticated session cookie	This plug-in can be used to invoke some very damaging scripts without requiring any other form of authentication. Before invoking a script, it is important to know what you are invoking.
Add Port Groups	Enables the creation of ultiple portgroups across multiple ESX hosts	This timesaving plug-in will be superseded by the distributed virtual switch.

Plug-in Name	Function	Security Issues
RDP	Adds RDP capability within the VIC	This plug-in requires you to enter login credentials, which can be for any user, which could be a way to elevate privileges.
Twitter	Adds Twitter functionality to the VIC	This plug-in can be used to access any Web page to which the VIC can connect.
KeySniffer	Sniffs all keystrokes that occur within the VIC and logs to a file	This type of plug-in could send the data to a third party instead of just logging to a file. It was created as a demonstration of how easily the plug-in mechanism within the VIC can be abused.
Hyper9 VI Client Plugin	Allows you to search your vCenter server	This type of plugin could access data for which you do not have access as it ties into the Hyper9 Server. If you do not tie into the Hyper9 server you will be limited to what you can see within vCenter.
H9Labs GuessMyOS Plugin	Adds icons to your VIC inventory review that represent the OS within the VM	Not a security risk per say as the OS determination is easily performed within the VIC.

Plug-ins are not currently digitally signed, and although some of the plug-ins are extremely useful, they should be used with caution. If possible, be sure you retrieve them from a trusted source, or if they are open source, review the code and compile them yourself to ensure that they are not going to do anything that would be considered dangerous.

VMs on the Wrong Network

It is extremely easy to place a virtual machine or appliance on the wrong virtual network using the VIC connected to either VC or the host. This could cause quite a problem if the VM, for example, was from a hostile environment such as a DMZ. There are different levels of hostile environments. From the virtualization host, all VMs are considered to be hostile, but within the virtual networks, other networks can be seen as hostile. If the VM ended up on a production network from a DMZ network, the VM could have been set up to detect this possibility and pivot an automatic attack into the production network.

These attacks could happen even if the VM had an IP outside the range of the virtual network on which it now resides. If the VM has been compromised, it is quite possible that the VM now has the proper routes to allow traffic through to

the VM or has a valid IP. At the very least, it could use MAC-based attacks. At the very least, this could create inadvertent data commingling.

If the VM ends up within a portgroup where promiscuous mode ethernet adapters are allowed, the VM could be used to sniff traffic across a virtual switch. Much of this depends on whether the VM was compromised in some way. As shown in Chapter 2, it is quite possible that the VM has been compromised and could be a ticking time bomb waiting for such an event.

The only solution at this point is to maintain multiple virtual infrastructure clusters, each for its own security zone. In other words, do not place VMs within the DMZ on the same hosts with VMs on other networks. Sometimes this is impractical from a cost perspective, but it is the best solution for preventing the possibility of a VM being placed on the wrong network. The second-best solution is to maintain diligent auditing that will tell you if this possibility has occurred.

Security Note

Create virtual infrastructure clusters for each security zone you manage.

Do not place VMs from one security zone on the wrong hosts.

Diligently monitor your configurations for misplaced VMs.

VMs or Networks Created Without Authorization

As we discussed earlier in the chapter, it is possible that the person who approves the creation of VMs could also be an administrator of the virtual environment. Any administrator is allowed to create VMs without going through the VM creation approval process. This often leads to the concept of VM sprawl, where VMs exist on the system that are unknown to other administrators or the managers who oversee the virtual environment. It is also possible for administrators to create virtual networks without going through an approval process. These unknown machines and networks could be the source of severe issues further down the line.

In the 2008 Verizon Data Breach Report,[2] there was a category for unknown unknowns, which comprised unknown machines and networks in use. The fact that they existed was an issue because they were created outside the approval

2. www.verizonbusiness.com/resources/security/databreachreport.pdf

process and therefore may not have the latest set of patches, updates, monitoring, and auditing performed, which implies that they could be the source of security incidents.

The solution is to have a review process in place that you must go through to create a virtual machine, appliance, or network within your virtual environment. This process could be as formal as necessary. Tools exist to help solve this problem (VMware LifeCycle Manager) but a process is still required.

Security Note
There must be a process for the creation and deletion of virtual machines, appliances, and networks.

VMs on the Wrong Storage

Another issue that happens is placing VMs on the wrong datastore. Datastores should mimic your security zones at the very least. If you follow the rules of at least one cluster per security zone, you will want to have datastores specific to each cluster. Just because a datastore has space on which to place the VM does not imply it should go there.

This can create three issues. The first is performance. Balancing loads across datastores is very important, and the inadvertent loading of a disk intensive VM onto a datastore that is well balanced for performance across all VMs could cause performance issues, and may cause denial of service as storage becomes overloaded. The second issue is that now there will be commingling of security zone data on a single datastore. In general, this is not a problem from a VM perspective, but a backup tool that has rights to see only one set of VMs will inadvertently be able to see backup VMs that the software or human is not authorized to access. This could lead to information leakage. The last item is that you could end up with a VM on the wrong storage network, or you now have a cluster than handles more than one security zone, which could lead to problems discussed previously. At the very least, we now have data commingling, and if the storage network is ever breached, you now have the possibility of losing more than you expected.

The solution is to audit your systems on a regular basis to ensure that VMs and virtual appliances are not placed on the wrong datastore.

VMs Assigned to Improper Resource Pools

Another issue that happens is placing VMs within the wrong resource pool. This can happen currently when the wrong options are chosen when a VMotion is performed. This could lead to a VM using more resources than planned. If the VM is one that would normally use quite a lot of resources, it could also lead to a denial of service for the resource in question or for the entire virtual environment, such as when CPU and memory get overloaded within a cluster. VMware DRS would kick in. This would force VMs to move about the environment, which could increase contention on disk and the network outside of the expected boundaries.

Normally this would be considered a performance issue, but the culprit is a VM that is outside the constraint of its supposedly assigned resource pool. Unfortunately, the only solution to this problem is to be watchful and audit the placement of each VM within the resource pools.

Premature Propagation of VMs from Quality Assurance to Production

With the advent of VMware Stage Manager it is possible to prematurely propagate a VM from one stage to another, whether that is from development to quality assurance or from quality assurance to production. This premature propagation could bypass a security step because the last phase could be to inspect the virtual machine or appliance for any security defects, install the latest virus signatures, and other security related configuration steps.

If this inspection does not take place, there may be security requirements not being met. In addition, a VM in QA could be reading different data than in production, and if there is a premature propagation the data sources could be incorrect. This could lead to further down time to fix the problem and perhaps a lengthy restoration process to recover the previous iteration of the virtual machine or appliance.

The only solution is to have a very good change control process that has written documentation to follow when VMs are propagated.

Physical to Virtual (P2V) Crossing Security Zones

In some cases you need to virtualize systems that are in one security zone—for example, a DMZ—but to do so you need to access the virtualization administrative security zone. In other words, you need to copy data from the DMZ through the VMware ESX service console or VMware ESXi management appliance. When this happens you are crossing security zones, which you should not do because it

allows the hostile environment of the DMZ to directly access your virtualization host's management appliances, which is to be avoided at all costs. This sticky process, however, has a very easy solution. Break the P2V into multiple stages using an intermediary system. Here is how the process works.

1. Create a virtual network for the security zone from which you want to virtualize, in our example, the DMZ. This virtual network should be connected to the physical network in the same security zone.
2. Create a pseudo security zone virtual network that mimics the real virtual network but is fully firewalled off, except to allow the necessary services to boot the VM: Perhaps DNS, DHCP, and the like.
3. Work with the security team to get a computer blessed to work within the source security zone—in this case, the DMZ.
4. Ensure this computer has enough attached removable storage to contain the entire source physical machine's disks.
5. Before attaching the removable storage to the computer, perform a complete disk wipe of the storage device and reformat with the appropriate file system.
6. Run a virus scan on the computer blessed by the security team, and then fix any issue or reinstall as necessary.
7. Using your favorite P2V tool, convert the physical machine to a virtual machine, storing the virtual machine on the removable media.
8. Move the removable media to a workstation outside the DMZ security zone. This workstation should now run a virus scan on the removable media. In addition, this workstation should most likely *not* be connected to the Web and should mount the removable storage read-only. A good system for this would be a forensic workstation or one blessed by the security team for this purpose. You are not looking for a virus within the VMDK, but one within the removable media itself that could have slipped on while it was within the hostile environment. If there is a virus or worm footprint, start the process over. However, be aware that VMDKs often show false positives.
9. Attach the now safe removable media to a workstation within your virtualization administration network.

10. Use VMware Converter or your favorite P2V tool to import the VM from the removable media into the virtual environment. The target virtual network should be a pseudo security zone that mimics the one from which the VM came.
11. Boot the VM within this protected environment and make any modifications as necessary. Common items to remove are any hardware agents and devices. Install VMware Tools, make any patches for the new hardware, and so on. This is also a good time to boot from a utility CD-ROM that contains tools to find rootkits and analyze the disk for viruses and other issues. Fix any issues found.
12. Power off the VM, move to the real DMZ virtual network, and then power on the VM.

These steps will guarantee that the P2V happens in a safe and secure fashion. There are many checks within these steps to test each phase of the P2V to prevent infiltration of viruses and worms into the administrative and virtual networks of the host.

Some may consider this overkill. However, I do not. It works very well to protect your investment minimally against virus infiltration and inadvertent information leakage while maintaining the integrity of the virtual environment.

Conclusion

In this chapter we have laid out how many management components communicate between themselves and the virtualization hosts. We have also discussed some of the pitfalls inherent within this communication. Last, we have laid out some common problems that can occur. In the next chapter we take things a step further and discuss everyday operational issues that occur within the virtual environment and the impact of security on them.

Index

Numbers

A

B

C

D

E-F

G–H

I

J-K-L

M

N

O

P

Q-R

S

T

U

V

W-X-Y-Z