# VMware® Infrastructure 3

## Advanced Technical Design Guide

*~and~*

## Advanced Operations Guide

*Two books in one!*

Ron Oglesby
Scott Herold
Mike Laverick

# Chapter 9 –Intro to Security

Security is often one of the most important, yet one of the most frustrating, aspects of any infrastructure. Virtualization compounds security due to the overall complexity of the environment. With VMware ESX there are a lot of components introduced into a single infrastructure and it requires a lot of thought and planning to properly manage a tightly secured environment. The components we are going to discuss have to do the with the core functionality of the virtual infrastructure:

- ESX Hosts
- VirtualCenter
- Virtual Machines
- Network Switching and Routing
- Storage Infrastructure

## ESX Host Security

Host level security is one of the most important aspects of the virtual infrastructure. What people who are newly introduced to ESX don't realize is that like any other host in your infrastructure, consideration needs to go into properly protecting both the service console and the VMkernel. The VMware ESX service console is a heavily slimmed down version of Red Hat Enterprise Linux 3. It is protected extremely well from many attack vectors out of the box, but it is by no means invulnerable.

### Overview of user and group accounts

Access directly the host using either the service console or the Virtual Infrastructure Client will use user and group permissions that are defined by service console authentication. This authentication model is the standard authentication system that is in place on any Linux operating system. Users are created and stored in the /etc/passwd file on the ESX host. Groups, which can contain multiple users, are stored in the /etc/group file. Finally, for security

purposes, encrypted passwords for the user accounts are stored in the /etc/shadow file.

The importance of knowing the low level details has been de-emphasized thanks to enhanced user management and role based policies that are integrated into the Virtual Infrastructure Client. This does not mean it is not important to have an understanding of how the authentication scheme functions within the service console; it's just that there is simply a limited need to use the service console to perform user and group management on your ESX host. In the following sections, we do want to provide more intimate details on protecting authentication to your ESX host by leveraging tried and true methods within the service console.

### *User Access*

VMware provides two methods to access your ESX host; through the service console and by using the Virtual Infrastructure Client. The default behavior of a new user after being created with the Virtual Infrastructure Client is to have no access to the service console or be placed in any default role (which we will find out about when we talk about VirtualCenter). If you are not using VirtualCenter, you can assign a user to a role within the Virtual Infrastructure Client to give them access to perform actions against virtual machines or configuration of the environment. By default, the root user is the only account that can login using the Virtual Infrastructure Client and is assigned to the administrators role, which gives them carte blanch access to manage the virtual machine environment.

When a user is given access to the shell they will be given the capability to log in locally or remotely via SSH to the service console. Once logged into the shell they will have the capability to run a wide variety of commands that can be used to query vital information about the host. For this reason, the only users that should ever be given shell access are the highest level ESX administrators.

By default, the root user is not allowed to connect remotely over SSH to the ESX host. It is not recommended that this default behavior be changed. Any user that connects to the ESX host should first connect as their self and use one of several methods of escalating their privileges to that of the root account. This allows all user activity to be traced back to the person connecting and running commands. If root over SSH were enabled there would be no way to de-

termine which administrator connected as the root user and executed commands at the command line.

The super user (su) command is enabled by default and allows a user to change their shell to that of the root account. Any command that is run after properly entering the root shell will execute with the privileges necessary to run any command on the system. An even more secure mechanism of allowing a user to run privileged commands is through the use of the "super user do" (sudo) application. Unfortunately, sudo does require configuration out of the box, but it is extremely powerful. It has the distinct advantage of giving a user the ability to execute single commands with privileged access without the need to enter the root shell. It can even be configured to only enable specific commands for users or groups. A quick Google search, or brief read of the security chapter of our previous book (available free online at www.vi3book.com) will provide enough information to get started with these enhanced user control commands.

### *Password Expiration and Complexity*

Once you've identified the users that require console access and have appropriately configured them you will need to ensure the accounts themselves conform to corporate policy. VMware provides mechanisms within the service console that provide us with the ability to enhance the security of our user account policies. This becomes very important when dealing with Sarbanes Oxley compliance for your ESX hosts. Since the service console exists and can be accessed, it must be protected like every other "Linux" host in your environment. Fortunately, VMware has the capability to tighten our policies utilizing PAM modules. There are several restrictions that we commonly see across the various environments we implement. Many of the restrictions are best practice and some are actually required by law.

Like many tasks that had to be manually performed in previous versions of ESX, VMware has provided a command to significantly simplify the process of locking down local authentication. The *esxcfg-auth* command has options to chance many of the policies surrounding user passwords including the following:

- Maximum password age
- Minimum number of days before a user may change their password

- Number of days before password expiration that a warning is given
- Minimum password length
- Password complexity

Every organization is going to have different policies surrounding their account expiration and complexity policies. Configuring some of the most secure environments will entail some advanced configuration of the ESX host. As soon as the environment grows beyond a few hosts it becomes near impossible to maintain these account policies on each individual host without a centralized infrastructure. The easiest mechanism to provide centralized management of user account policies is to tie your ESX user authentication into your existing login authority, which is typically Active Directory.

### *Enabling External Authentication*

There are several whitepapers from VMware that have been published in regards to authenticating your ESX servers against an NT4 or Active Directory environment. VMware provides documentation on configuring your ESX host to authenticate users against their Windows account information. Instead of covering the process here we will discuss usage scenarios for this setup as well as some of the benefits and drawbacks of integrating ESX authentication into your Windows password database.

There are obvious advantages to integrating your ESX environment into your Windows authentication database. Utilizing a centralized database allows your ESX users and administrators to remember a single password. In addition, the challenge of synchronizing passwords across ESX hosts disappears. The use of local authentication into the service console should be used extremely sparingly and only for the highest level ESX administrators. When VirtualCenter is utilized in a virtual infrastructure we recommend that it be utilized to do the things that it does best; one of which is centrally manage user authentication to perform day-to-day tasks.

Integration of ESX into Windows authentication makes managing large amounts of users significantly easier, but is not the best solution in every instance. Several things should be considered before attempting to configure your system to authenticate against a Windows database:

- SMB (NT4) is significantly easier to setup than Kerberos (Active Directory).
- If using a Native mode Active Directory you must use Kerberos. Accurate time synchronization is critical to the proper operation of Kerberos authentication.
- An account must still be created for each user as a point of reference for the service console. Only the passwords are referenced on the backed database.

## Host Firewall Configuration and Recommendations

VMware uses an iptables firewall to limit the communication coming into and out of an ESX Host. By default, the policy is extremely restrictive and only communication required for management of the environment are enabled. There is a very good chance that several modifications will need to be made to the default firewall policy. VMware has provided quite a few default rules that can easily be enabled or disabled using the Virtual Infrastructure Client. These include the required port definitions to enable outgoing NTP queries or for the configuration of hardware based monitoring agents such as IBM Director and HP Insight Manager. There are too many default services to list as filler content for this book, and every one of the services can be easily viewed from the Security Profile configuration screen for the ESX Host.

Naturally, VMware does not have a definition for every possible incoming and outgoing service that can be available to an ESX Host. A good example of this is sending log messages to an external syslog server. In order to open the proper communication you will have to use the Service Console to run the esxcfg-firewall command to enable the proper communication. More details on this command can be found in the "Administrator's Guide" complimenting the design guide.

Finer tuned control can be applied to firewall rules by leveraging the iptables command directly without the use of the esxcfg-firewall wrapper utility. This includes limiting communication to or from a particular host or subnet. Before opening any additional network ports for the iptables firewall, take the time to ensure you are only opening what is absolutely required and that you analyze any negative impact of enabling that type of communication from either entering or exiting your ESX Host.

## Syslog

As we will find out when we discuss VirtualCenter, VMware really provides no capabilities to actually audit your virtual infrastructure. VirtualCenter can only track centralized access to the infrastructure through the centralized management server. Access to individual hosts through the service console and long term tracking of Virtual Infrastructure Client access directly to the host cannot be tracked from outside of the host itself.

The best way to track access for authorized and unauthorized use is through an external syslog server. Fortunately, the service console already leverages syslog locally for its own internal log files. The problem, from a security standpoint, is that leaving these files locally can potentially allow someone attempting to maliciously access the system can cover up their tracks by manipulating these local log files. With a simple configuration file change it is simple to redirect syslog to also write to an external system (Keep in mind you will need to open up the proper firewall access on the host to allow this communication). Using an external syslog server has several very distinct advantages.

First and foremost, it provides a centralized location for the collection of all log files from every host in the virtual infrastructure. This fact alone should be sufficient enough reason for everyone implementing VMware ESX servers in their environment to consider using a syslog server. Without a centralized syslog server the only way to audit remote connectivity and root access is to check the log files on each individual host either through the service console or by connecting directly to each host using the Virtual Infrastructure client and reviewing the log files there.

Another major advantage to syslog that does not necessarily tie to security is in the fact that it can capture far more information than VirtualCenter ever likely will. Simple notifications that should be blatantly to your virtual infrastructure monitoring platform are easily and immediately captured in syslog. With a proper syslog server alerts and actions can be set up to notify and attempt to resolve some of the potential issues. The following are examples of details that can only be captured through log files/syslog.

- Unauthorized use of the root account – Allows the tracking of people trying to remotely connect as root or leverage the sudo command outside of standard security policies.

- Loss of connectivity to fiber storage – Losing a single fiber path in a properly planned infrastructure will not cause a storage outage, but going down to a single path can definitely cause performance issues and puts your environment at risk.
- Loss of network connectivity for a physical NIC – Like storage, losing one NIC will not likely cause an outage to network access for your virtual machines, but it doesn't mean you don't need to know when it happens.
- In depth information about the VMkernel – Most of this information is overkill for typical situations, but a properly configured syslog parser can give notice to the end user when something unexpected, but normally invisible, occurs.

Any typical Linux distribution can serve as a syslog server in the environment. In addition, many enterprise monitoring solutions offer syslog integration and parsing. For small to medium sized organizations looking to enhance monitoring through centralized syslog management, there are many tools available. The most popular, and ultimately one of the best out there is the Kiwi Syslog Daemon, which runs on a Windows host. It can be downloaded from the Kiwi Enterprises website at http://www.kiwisyslog.com/.

## Time Synchronization

When you start centralizing your log management and system management it is critical that every system display the exact same time. If a series of events is occurring to multiple hosts at the same time it will be critical for people responsible for troubleshooting or post mortem reports to properly reassemble the chronological order of events.

VMware ESX uses the standard Linux implementation of NTP and it is highly recommended that it be enabled and configured on every host in the environment. All hosts should synchronize their time against a time source external to the virtual infrastructure such as a timekeeping appliance or an Active Directory domain controller. It is not recommended that an internal component to the virtual infrastructure such as a virtual machine or ESX host be referenced as a time source.

At this point it is only possible to configure the NTP service from the service console. You will also need to make sure that the proper firewall access is opened up on each host to allow NTP requests to leave the host. This is typically done by ensuring outgoing UDP port 123 is open.

## Agent Installations and Local Process Execution

One of the most controversial subjects when virtualizing infrastructures with VMware's ESX product is whether or not the service console should be treated like a Linux installation or as a hardened appliance. Unfortunately, the answer is "both". As long as we can see, touch, and interact with the service console the appropriate actions must be taken to properly protect and audit the operating system instance. At the same time we need to be conscious of the fact that the service console is performing a very specific task and any processes running inside the service console could potentially impact the performance of the virtual machines being powered by the host's resources. Typical applications such as virus scan software are normally avoided at all costs due to the amount of potential load they can add to a host.

This information obviously begs the question, "So what is acceptable when considering running agents or processes inside the service console?" We first need to determine the different methods that software vendors and system administrators are using to perform actions inside the service console. Typically, this consist of three various mechanisms, often a combination of more than one. There is really nothing wrong with using any of these mechanisms at this point, but there are some critical items that must be considered upon determining that a specific application or product requires direct access to the service console. These guidelines will ultimately determine whether leveraging the service console for these applications is acceptable or not for your environment.

### *General Guidelines*

Regardless of the mechanism that is used to execute code at the service console a few things should be considered before installing and using the application. The first is determining if the application truly requires the use of the service console. There are many things that can be done through the VMware SDK or other external mechanisms that do not require the use of the service console, but a software vendor may have already had something that worked for Linux that they decided to jump on the virtualization bandwagon with. The service

console is heavily modified and locked down from typical Linux installations, and installing software that was written purely for a Linux installation will cause more problems than it will likely ever solve.

The second critical guideline is to determine whether benefit will truly be utilized by installing software or executing processes within the service console. The benefit of the application should also be compared against potential performance impact of the software running inside the service console and the resources it is going to require. I alluded to an example of installing antivirus software inside the service console not being recommended. This is due to the fact that even a default installation without additional configurations is highly secured out of the box. Running a nightly virus scan and performing real time file system protection typically uses a lot of system resources nearly constantly. This has a very negative effect on the ability to schedule resources for your virtual machines in a timely fashion. If an agent or process has major benefit to your organization and the resource requirement is low or execution is required in infrequent after-hours bursts, there is nothing wrong with installing an agent or executing binaries in the service console.

### *Scripting*

Running scripts locally inside the service console is the least preferred method of interaction with the service console for several reasons. First, while scripting is easy for anyone to manipulate, it is also easy for anyone to screw up. Most scripts often require customization to work in a particular environment. A simple mistake in modifying a script can wreak havoc on the resources assigned to the service console. Add that to the fact in most organizations one or two people are intimate with the script, and others are simply responsible for making sure it runs. If one of the script maintainers leaves the company or gets hit by a bus, there will be some very upset, but newly promoted ESX administrators that have to wade through someone else's script to determine how to make further modifications. If you will be using scripting for executing processes or performing actions within the service console, make sure it is very well tested in development before rolling it into production. In addition, make sure it is extremely well documented and commented to make it easier on the next person that has to assume ownership of further script maintenance.

We do need to specify here that remotely running scripts that communicate with the virtual infrastructure from outside the hosts themselves is actually quite

a common practice. With the introduction of the VI Perl Toolkit regular users can finally easily interact with VirtualCenter and automate many tasks. By moving scripts from the ESX host and to an external system there will be a loss of some lower level interaction with the host itself, but scripting really be used for these purposes anyways.

### *Agent Installation*

Many people treat agent installations like they are a rare and deadly infectious disease. Most of the time the reasoning behind this is the fact that "It runs in my service console", which doesn't really hold any water when comparing it against what is occurring with scripting or, as we will see, binary injection. A properly written agent will use an extremely small amount of system resources and will often enable functionality that is simply not available through native means such as the SDK or VMware command line applications. There are actually advantages to manually installing an agent within the service console. First, since it must be manually installed, it is very easy to track exactly when the agent was placed on the system and who installed it. Since the installation is often either in RPM or scripted installer format, it is also extremely easy to see exactly what changes are occurring to your system. You never want to blindly execute anything in your service console. That is a recipe for disaster.

A properly written agent will not constantly run and will only leverage system resources while an action is occurring. This can be controlled several different ways such as through xinetd or other service control mechanisms. The exception to this would be some form of monitoring agent that is capturing statistics that simply cannot be provided by the SDK. These are typically very lightweight and leverage resources on regularly scheduled intervals when polling metrics. The key to installing an agent is knowing exactly what is occurring when the agent is installed. As with all service console based applications, you must be conscious of the performance impact to the host and virtual machines compared to the benefit gained from the application itself.

### *Binary Injection*

Binary Injection is the process in which an application connects to an ESX Host, typically over SSH, to send compiled binaries for execution within the service console as a part of some form of automated action. These components are often, but not always, completely removed from the ESX Host when the

process or set of processes have completed. This is how most software vendors can claim that they are "agent less" which, based on the methods being leveraged to inject binaries, is typically marketing fluff. This method is ideal for administrators new to ESX and Linux and who are not familiar with the typical methods for installing applications through scripted processes or RPM packages. It provides a simple mechanism to keep the end user away from the service console, which is definitely not a bad thing in many cases. As a matter of fact, this is the method that VMware themselves use to inject their "agent" when a host is configured to be managed by VirtualCenter.

On the other side of the coin, this method is the one you also need to be most careful with. Many users go along not knowing that these applications are touching the service console, and thus have no idea what changes are occurring on the system. Any software vendor that leverages this technique should provide a checklist of what their application is doing to an ESX Host that includes at least the following information:

- Does the injected process require root access to run, and if so, can they accommodate non root over SSH?
- What binaries are being placed onto the system, where they are being placed, and what purpose are they serving by being injected? Any software vendor not willing to part with this basic information, which can be provided without giving away intellectual property, should be approached with caution.
- When the process completes are the injected components completely removed or are they left in place for easier execution the next time the process executes?

Again, we want to emphasize that there is nothing wrong with leveraging the service console to execute processes. Often times, it enables advanced functionality that simply cannot be performed through external methods such as leveraging the SDK. With any of the methods, take special care to analyze the benefit of any service console based process against potential performance impact to the service console resources. As VMware moves closer to removing the service console entirely, make sure you test every support application, as you will find a vast majority of them rely on having access to the service console for process execution.

### Patching

Like any other system in your environment your ESX Hosts are prone to vulnerabilities. VMware is extremely responsive in providing updates when new vulnerabilities are discovered. While not every patch or update requires a reboot of the ESX Host, many do; especially if any critical files used by the VMkrenel are updated. Fortunately VMware has provided the capability to put your host into "Maintenance Mode" to safely apply updates to the system. If you have VMotion and DRS properly configured in your cluster, putting a host into maintenance mode will automatically move your virtual machines off to other hosts in the cluster while you perform the necessary maintenance on the targeted host.

You should plan your updates carefully and ensure to test all functionality prior to rolling the updates into the production infrastructure. Security updates are often relatively clean and do not impact support applications or other core functionality of the system. You will, on the other hand, need to be quite a bit more careful with update releases that introduce new functionality, as these are more likely to change components that third party applications may depend on.

## VirtualCenter Security

VirtualCenter is the centralized point of all management in your virtual infrastructure. Unauthorized access to this host could be disastrous as a malicious would have full control over your entire environment, including the capability to shut down, delete, and worst case, take with them any virtual machine managed by the infrastructure. Proper security of the VirtualCenter server itself as well as user roles and policies for managing the infrastructure are the key to a secured virtual environment.

### Access to VirtualCenter Server

Access to your VirtualCenter server should be tightly controlled and as restrictive as possible. A user that has administrative access to the VirtualCenter server has a full set of keys to the kingdom when it comes to manipulating the virtual infrastructure since, by default, anyone who is a local administrator of this system is put into the "Administrators" group for the virtual infrastructure. Because of the volatility of this system, the administrative privileges should be extremely restrictive with only the most trusted of users having access.

It is not uncommon for the VirtualCenter server to reside on a separate management network along with ESX Host Service Console connectivity so tight network ACL controls can be put in place on this management VLAN. We will discuss this in further detail when we discuss network security later in this chapter.

When considering VirtualCenter security, many people overlook the fact that all VirtualCenter data is stored in the back end database backend. Unauthorized access to the database itself will allow a user to quite easily capture an entire inventory of the virtual infrastructure and even make modifications to certain configuration aspects of the infrastructure, including user access rights. The database server/instance should be treated just as securely as the VirtualCenter server itself.

## Virtual Infrastructure Management

Whether you are using the Virtual Infrastructure Client to communicate directly with the ESX host or to VirtualCenter you will need to be aware of the structure that VMware uses to assign permissions for users to manage the virtual infrastructure. The same methodology is used whether you are communicating directly to the host or are using VirtualCenter to manage the infrastructure so we will not discuss them separately. Rather, we will point out the subtle differences in the three key components of assigning permissions to your users; Users and Groups, Roles, and Policies.

### *Users and Groups*

I am hoping it doesn't take a lot of explaining to describe what a user and a group is, and if I do, you guys have the wrong book. If you do not leverage VirtualCenter for your virtual machine management and work on a per host level, users and groups are defined by using the standard Linux user and group configuration of the service console. Environments that are managed by VirtualCenter leverage the Windows account database of the server VirtualCenter is installed on. VirtualCenter may use either local or domain based users and groups for assigning permissions to users.

The default permissions of VirtualCenter allow anyone who is a local administrator to have full control of the virtual infrastructure. It is best to change this

default behavior shortly after installation. A common practice is to create an Active Directory Global Group with the domain users that are VirtualCenter administrators. This global group must then be added to a local group on the ESX host. This local host is then given administrative privileges through the VirtualCenter client. It is not possible to assign domain global groups to roles directly within VirtualCenter; a fact that is extremely annoying. It is also recommended that a local (to the VirtualCenter server) user account be created and assigned to the administrator role within VirtualCenter. This will ensure that even if domain connectivity is lost for whatever reason, the entire virtual infrastructure does not become unmanageable. Once the initial user and group configuration is complete, the local administrators group should be removed from having any direct access to VirtualCenter.

### *Privileges*

A privilege is a single action that can occur within the virtual infrastructure such as powering on a VM or creating a new datacenter. Overall there are over 100 privileges that are broken up into manageable categories at both the host and VirtualCenter levels. Many privileges go hand in hand, but must be configured separately. As an example you will often configure the ability to power off a virtual machine to anyone who has the ability to power them on. These are both unique privileges that would typically be assigned at the same time. The key to properly organizing these various privileges to create templates for management is done through the use of roles.

### *Roles*

Roles are a collection of privileges that are grouped together to allow a specific type of management for users of the virtual infrastructure. As an example, a standalone ESX host has three default roles configured; Administrator, Read Only, and No Access. A user or group that has been assigned to the administrator role has full control to the virtual infrastructure, while a user assigned to the read only role can only view the configuration of the infrastructure without having the ability to make any changes. The no access role is something of a special case that is used to deny specific privileges to a small set of individual users that may be a part of a group that has access to a particular role. Within the virtual infrastructure, the most restrictive set of privileges takes precedence for users that are assigned to multiple roles that are assigned to the same object.

At the VirtualCenter level there are quite a few more roles that may be assigned for infrastructure management. Each of these roles is tailored for providing very specific capabilities to users or groups that are assigned to them. The following are the default roles that exist within VirtualCenter:

- Virtual Machine Administrator
- Datacenter Administrator
- Virtual Machine Power User
- Virtual Machine User
- Resource Pool Administrator

In addition to the preconfigured set of roles, the Virtual Infrastructure Client provides the capability to create custom roles, clone roles to base new roles off an original, or modify existing ones (with the exception of the no access, read only, and administrator roles). This capability allows you to tailor specific roles for your environment based on the needs of the users accessing the infrastructure.

### *Object Level Permissions*

Each level of the infrastructure can have unique permissions. Any object of the virtual infrastructure such as Datacenters, Clusters, Hosts, VMs, Folders, and Resource Pools may have unique permissions assigned. A permission is defined by stating a particular user or group is assigned a role at an object level. Permissions can be configured to propagate down the tree and be inherited by child objects. The default configuration of VirtualCenter states that members of the VirtualCenter server's local administrators group is a member of the Administrators role within the virtual infrastructure and that permission is propagated down throughout the entire infrastructure. All other users are automatically given the same rights as the no access role.

There is no way to limit how far down the virtual infrastructure a propagated permission has access. If permission is defined at a datacenter level and is selected for propagation, that permission goes down all the way to the virtual machine level. If there is a reason to limit or increase the permissions of a user or group to any object, a new permission can be assigned for that particular object.

Permissions assigned at the object level always take precedence over propagated permissions.

## Virtual Machine Security

Virtual machine security in a virtual infrastructure is not that different than protecting a physical machine environment. The same procedures that you would need to follow to protect your physical infrastructure should also be carried over into your virtual. Scheduling of CPU or disk intensive tasks as they relate to security, such as full virus scans require special attention to prevent an instance in which performance for an entire host or LUN is impacted.

### Process and Memory Isolation

Each virtual machine is run through its own virtual machine monitor (VMM) within the VMkernel. These VMMs are unique to the virtual machine in which they are handling processor calls and do not share instructions with any other VMM. This provides a secure model to ensure than processes that are being executed within one guest instance cannot be seen by others.

Memory does not have the same level of isolation as processing resources. A major benefit of using VMware is in its capability to overallocate memory on a given host by using memory sharing techniques. Virtual machines that have identical memory pages across multiple instances will actually share the same physical memory space at the host level. If any operating system modifies the memory that is shared it will immediately be detected by the VMkernel and be rewritten to its own unique physical memory page. Due to the method and speed in which unique memory can be rewritten there is no risk in which a virtual machine can access the memory resources of another system that is unique to the secondary system. VMware not only provides high speed memory access and the capability to over-allocate resources; they manage to do it securely.

### Patching

Keeping your virtual machines up to date with the newest patches should not be handled any differently than it is done in the physical environment. If anything patching becomes more critical because of the simplified process of de-

ploying new virtual machines in minutes. Many organizations are introducing new challenges to patching because the sheer number of operating system instances being deployed is increasingly dramatically thanks to virtualization. One thing about patching that is significantly improved is that the virtual machine templates can be very quickly and easily updated. This ensures that any newly deployed virtual machine will already be as up to date as possible.

### *Template Management*

All users should leverage the template capabilities of VirtualCenter. By properly managing and updating templates an organization can ensure that any new virtual machine that is deployed is done so in the most secure fashion. Templates should be stored in a centralized repository that each ESX host has access to. NFS volumes are often prime candidates for template storage due to the cost of storage and native integration as shared centralized storage for the entire infrastructure. Any time there is a security or maintenance update to any major component of the template it is critical that the template is not accidentally left out of the update process due to the fact that it s a powered off virtual machine.

## Virus Protection

Running a virtual infrastructure provides no additional protection for your virtual machines in regards to their ability to become infected by viruses. Standard virus policies should exist for all virtual machines just as they do for physical. Special consideration does need to go into engineering the schedules for full virus scans. Full scans are often extremely CPU and Disk intensive; both of which are valuable resources to a host. I have seen entire ESX hosts grind to a halt simply because 4-5 VMs kicked off a full virus scan at the exact same time. It will be a significant pain to configure these schedules the first time, but once a policy is in place it will be relatively simple to follow for any new virtual machine that gets added to the infrastructure. If using a serial-based schedule it is also important to note the impact that VMotion and DRS will have when configuring scheduled windows at a host level. Before you know it you could be right back in the same situation where virus scanning was bringing down a host.

# Network Security

Network security is often the most complex aspect of any environment, which is quite understandable. If systems had no way to communicate with one another, security would be a heck of a lot easier. Unfortunately, not only is there plenty of communication going on in a virtual infrastructure, securing it is quite a bit more difficult based on the fact that the ESX hosts themselves take on roles that have been typically reserved for the physical network infrastructure and the network administrators. When considering network security in the virtual infrastructure you not only need to think about protecting the ESX hosts, but also your virtual machines and any network based storage that is being used to house and run virtual instances.

## Management Network Isolation

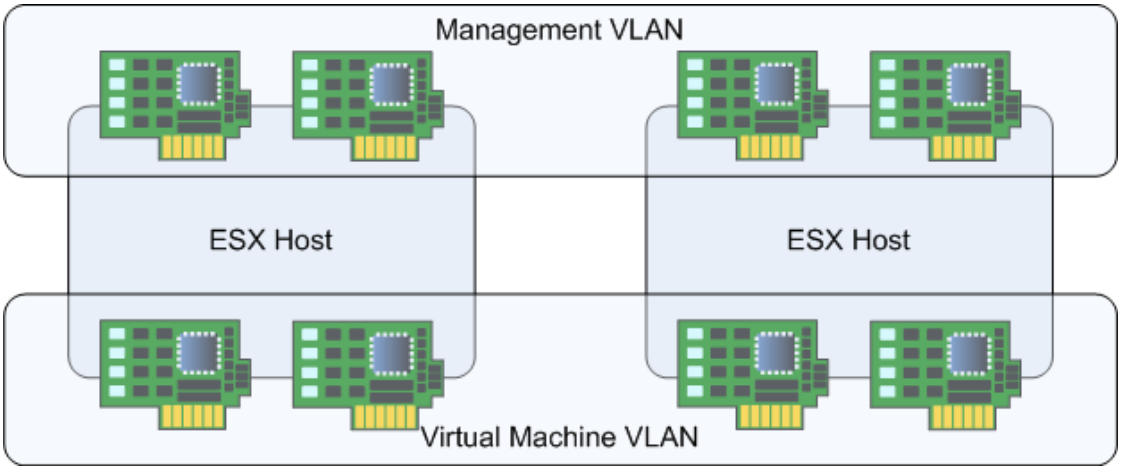
When we discussed ESX 3 networking earlier in this book we made quite a few references to securing your management services on a separate network from your actual virtual machines. This recommendation was made to better isolate and protect the traffic that can potentially communicate with the service console or interfere with other management services.

### *Console Access*

The default behavior of VirtualCenter and ESX is to only allow communication through secure communications such as SSH, ssl, or encrypted communications for virtual machine management over tcp ports 902 and 903. Since the host has an iptables firewall running we can easily ensure the host only accepts traffic that is used for management of the environment.

There are several varying degrees that organizations will go to in order to protect their management interfaces for their ESX hosts. Of course, each additional layer protection adds complexity to the management of the environment.

**Figure 9- 1: Secured VLANs**




One thing is certain in that all VMware physical management NICs should be contained within their own VLAN as shown in Figure 9-1. These NICs will support communication to the host's service console from VirtualCenter or the Virtual Infrastructure Client as well as VMotion traffic between hosts.

The management VLAN itself can be secured and isolated so nearly all communication from other networks cannot enter and attempt to access the ESX hosts without authorization. In this scenario the first question that comes to many people's minds is "How do we manage the environment if it is completely isolated?" In this case the VirtualCenter server would also exist within this management VLAN. Depending on the organizations policies Authorized desktop systems could be added to the ACL to communicate with VirtualCenter and possibly the hosts. If an organization is, in my opinion, overly stringent on its security policy, management desktops with the Virtual Infrastructure Client and possibly putty installed could also be set up in the management VLAN with RDP access from authorized desktops. This significantly complicates the process of maintaining the virtual infrastructure and has scalability issues as more ESX administrators are introduced into the organization.

### *VMotion*

More serious security concerns begin to arise when additional services are enabled for advanced functionality such as VMotion, which is not encrypted. All communication that occurs between hosts during a VMotion migration is unencrypted and can technically be picked up with a network sniffer. This where the

use of an isolated VLAN as described in the previous section is critical. The fewer systems that can be used to attempt to view network activity during a VMotion, the better protected the infrastructure is as a whole. Sending VMotion traffic across VLANs should be proceeded with caution and fully investigated for potential risk before ever being configured and attempted.

### *Network Storage*

VI3 introduces the capability to configure network based storage in the form of iSCSI and NFS for storing and running virtual machines. The traffic that travels between the ESX host and the back end storage is not encrypted or protected once it hits the wire. With this in mind it is extremely critical that the network storage be isolated in its own VLAN at a minimum. For maximum protection and performance it is recommended that the storage communication also occur on its own network infrastructure as shown in Figure 9–2. Assuming this infrastructure is properly configured, it is very difficult to accidentally configure a physical or virtual machine to share the same network as the storage network. Stringent policies around the connectivity, configuration, and mapping of physical adapters should be created and enforced to guarantee a secure communication channel between the hosts and the back end storage infrastructure.

We've gone over the fact that network storage is not encrypted, but this does not mean you are powerless from having someone find a way to connect to your network based storage resources. Configurations that can be applied to both the source hosts and target storage arrays can be used to require authentication and masking to prevent unauthorized systems from seeing storage they should not.

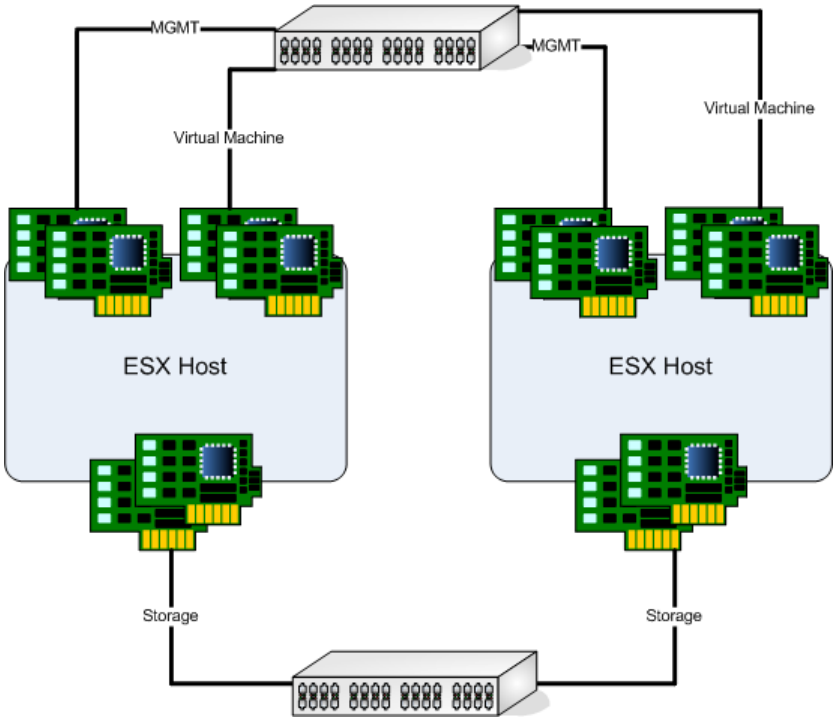
Protecting NFS volumes that are used for template storage is not as critical as protecting NFS volumes that are being used for running virtual machines. Regardless of what you are using NFS volumes for there is an authentication mechanism that requires the use of a username/password combination to properly access the disk resources. Users who use NFS for running virtual machines should go the extra mile to ensure the traffic between the ESX hosts and the storage infrastructure is as isolated as possible.

Properly protecting iSCSI is a bit more complicated than simply supplying a username and password, although this is highly recommended as well. iSCSI

supports protecting connectivity using CHAP authentication at the storage array. Hardware iSCSI initiators and the VMware software iSCSI initiator both have the necessary functionality to authenticate against the target. The VMware iSCSI initiator does not have the capability to authenticate at a per target level so once your host is properly connected it will have access to any LUN that is zoned to the iSCSI adapter name.

Zone plays a crucial role in securing your data volumes at a SAN level and we will discuss it towards the end of this chapter when we talk about storage security in particular.

**Figure 9- 2: Networked Storage**




## VLAN Isolation

Many ESX environments leverage VLAN tagging to support various network configurations across a large number of virtual machines. Network and security administrators often have concerns about the security surrounding the virtual switch model, especially when they have to start trunking multiple VLANs to the host. Networking capabilities for virtual switches and port groups in ESX 3

operate inside the VMkernel. Virtual switches operate at Layer 2 of the network stack only. They simply receive and forward packets to the proper virtual ports of the switch. There are no layer 3 capabilities to route traffic or bridge traffic across multiple virtual switches. The only way in which multiple VLANs can communicate with one another without leaving the host is if there is a virtual machine that is set up as a router to manage the layer 3 communication.

Physical NICs do have the capability to be configured to listen to traffic tagged for more than one VLAN. This is important if you do plan to leverage a DMZ or other secured network on your existing ESX hardware. This is the reason we highly recommended using a different set of physical NICs if you do intend to use a DMZ on your host. Physical NICs cannot belong to more than one virtual switch at a given time. Configuring multiple virtual switches helps isolate this traffic as much as possible. Overall, leveraging multiple VLANs on a single host does not pose any form of security risk by allowing traffic to bridge across various networks without following the standard layer 3 routing policies of the physical network.

## Bridging Network Zones

A topic that is very commonly discussed when designing a virtual infrastructure is the layout of a secured network such as a DMZ. There are two schools of thought with this one; the first being to provide a separate cluster of ESX hosts for highly secured network access. In this case, a group of ESX hosts are configured into their own cluster and given access to VLANs that are considered "secure". They are not given any access (outside of the management interface) to the internal network. It is not uncommon to have the management interfaces configured on an internal DMZ as to completely isolate the ESX host to keep it as secure as possible. This is easily the most secure way to provide a virtual infrastructure for customer facing web servers. In smaller environments or organizations that do not have a lot of external facing or secured systems, this option could lead to underutilized hardware.

**Advantages of creating a separate cluster for secured systems**

- Highly secured environment for public facing or highly sensitive communication
- Removes end user error of improperly configuring a VM

**Disadvantages of creating a separate cluster for secured systems**

- Requires an additional cluster, consisting of several ESX hosts for redundancy
- Often leads to highly underutilized servers in small environments or environments with few sensitive servers

The second school of thought is to create an additional virtual switch with additional physical NICs that are isolated from the internal network. This allows users to leverage their existing ESX infrastructure to support public systems. There is an inherent risk involved with this configuration since a careless administrator could accidentally place an internal virtual machine onto the external network. Worse yet, they could accidentally set up a virtual machine that has connectivity to both the internal and external networks at the same time…which is quite bad. The virtual switches and virtual machines on the host should be audited regularly to ensure no breach in security policy has occurred.

**Advantages of creating public or isolated virtual switches**

- Can use existing host/cluster architecture and layout
- Does not change server configuration or management policies
- Maximizes ESX host utilization by running all virtual machines on a single virtual infrastructure

**Disadvantages of creating public or isolated virtual switches**

- Not as secure as an isolated network environment
- Leaves open the opportunity for human error in the configuration of virtual machines

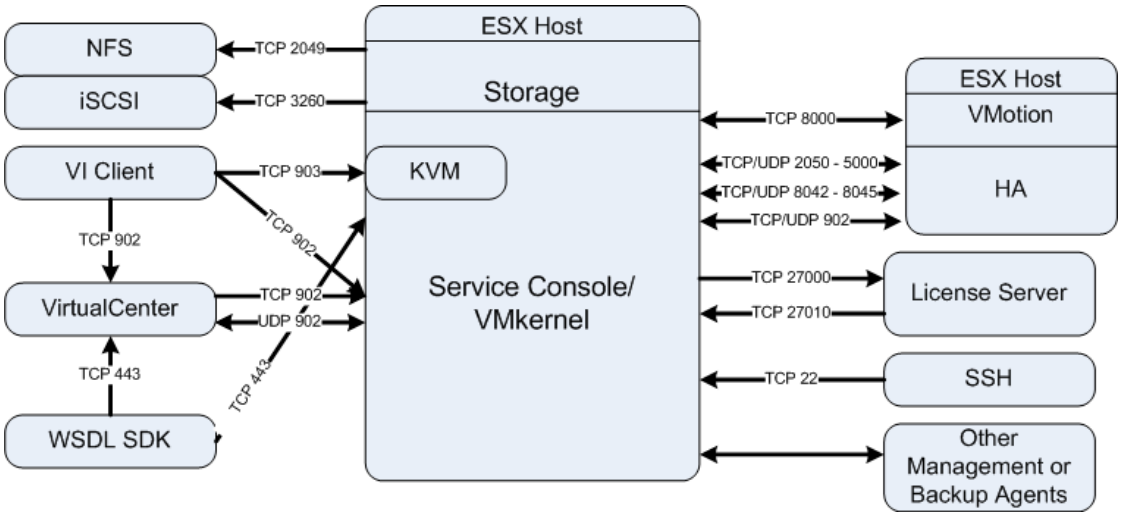
When taking a look at network isolation and configuration of public access for virtual machines it is extremely important that people understand that under no circumstance should the ESX management functions (service console, VMotion, etc) EVER be plugged into a public network. Anyone who configures their environment in this fashion deserves to have all of their data stolen and be brutally criticized on Slashdot or Fark.

## Using VI3 through a firewall

We have made several recommendations here to protect and isolate the management connectivity of the ESX host by providing a management VLAN for all management tasks. The best way to protect this VLAN is through the use of

a firewall or ACL. ESX requires a well defined and very specific list of ports for network communication. Figure 9 – 3 displays the various network ports that are required for the various communication components of the virtual infrastructure.

**Figure 9- 3: Port Mappings**




## Storage Security

Any time you decide to use an enterprise storage infrastructure it is important to meet the security best practices while doing so. This is far outside the scope of this book, but what we do want to talk about are some of the extra considerations that are required when securing the centralized storage environment used in a VMware virtual infrastructure.

### Highly Sensitive Data

Virtualization is one of the fastest growing technologies and with new innovations by VMware and the hardware vendors we are starting to see more and more enterprise level applications virtualized. These enterprise level systems often have more stringent requirements on the sensitivity of their data and often require additional planning around their placement in the infrastructure.

The most common practice I have seen is creating a cluster of ESX hosts particularly for the purpose of enterprise applications and secured data. The shared

storage architecture of VMware means any host in the cluster has the capability to see the same LUNs. The best way to prevent a virtual machine from accessing a VMDK file or raw LUN with sensitive data is to make sure it cannot see it at all. Providing a cluster just for this purpose makes this possible. Of course, by segregating the environment into a secured and non secured environment does introduces some management challenges as increases the likelihood that some of your infrastructure will be underutilized. Secured data configurations also typically have a direct relationship to systems that have heightened security requirements over the network. This fact may help justify the use of several standalone ESX servers in a "secured" cluster.

## Zoning

The ability to specify which hosts can and can't see a particular data LUN is controlled at the SAN level by using zoning. Zoning at the SAN level allows a user to specify exactly which HBAs or hosts are allowed to see a particular LUN or group of LUNs. ESX relies quite heavily on zoning, especially when multiple clusters are being used in larger environments. If you plan on using VMotion and DRS you will need to ensure that every host in a cluster is properly zoned to see the same storage or the benefit of real time migrations will be lost to those hosts.

This added level of security further enhances the iSCSI mechanism of using CHAP authentication to determine if a system has the proper access to a LUN. Not only must the host properly authenticate, but the HBA must also be defined by the SAN as having the capability to see the LUN. Once the data volumes are properly secured we need to consider the data that will eventually be stored on them.

## VMDK Creation

In chapter 5 we discussed VMDK files in detail and found out that there are several ways in which they can be created. The default behavior for an ESX host is to create VMDK files in "zeroedthick" format. Any data that resides on the physical disk still exists on the VMDK file until a read operation is requested from the virtual machine. At the time of the read operation the data is zeroed out and its original contents are cleared. From a guest operating system level there is no way to read data that originally existed on the VMDK file. This

is not the case when accessing the VMDK file at a block level from outside the virtual machine.

If you are using a storage infrastructure that has had multiple virtual machines configured, deleted, moved, etc, you will likely create VMDK files that contain existing data. If you perform image level backups of these virtual machines you will be able to capture this "stale" data and read it from an external system. Most people do not realize that this level of access to data from a previous system exists. The safest way to prevent this behavior is to create your VMDK files as "eagerzeroedthick" from the service console before assigning them to a virtual machine. Disks created in "eagerzeroedthick" format a zero-filled at creation time, which overwrites old data. The obvious downside to this is that it also takes more time to allocate disk files for your virtual machines. In addition, there is no way to create "eagerzeroedthick" VMDK files through the Virtual Infrastructure Client; it must be performed from the service console directly. For those that are concerned about ensuring highly sensitive data doesn't make its way into virtual machines due to the back end storage infrastructure it is highly recommended that you take the time to create VMDK files in "eagerzeroedthick" format.

## VMDK Access

The ability to create snapshots and capture fully running virtual machines from the ESX host also introduces another challenge. It is now quite easy to capture and move an entire virtual machine. While this portability is actually a huge benefit, it is also a security risk. Once the virtual machine is removed from the ESX host it can be moved to any Windows workstation and very easily mounted as a drive or run on free VMware products like VMware Server as the actual virtual machine itself. Protecting access to the ESX service console will help a user from manually capturing this data, but similar practices must be put into place to protect data once it has been removed from the host. Image level backup utilities should have data encryption mechanisms. If data is not directly encrypted it should be stored on a file system that is locked down and likely encrypted as a target for backups only. Any user that can take a VMDK file in an unencrypted format will have full access to all data of that virtual machine.

## Conclusion

ESX requires significant planning around security when introducing it into your environment. Combining the typical best practices of your existing physical infrastructure and taking the additional requirements for virtualization into consideration will ensure your virtual environment is as protected as is possible.