VMware Infrastructure 3

Security Hardening

By introducing a layer of abstraction between the physical hardware and virtualized systems running IT services, virtualization technology provides a powerful means to deliver cost savings via server consolidation as well as increased operational efficiency and flexibility. However, the added functionality introduces a virtualization layer that itself becomes a potential avenue of attack for the virtual services being hosted. Because a single host system can house multiple virtual machines, the security of that host becomes even more important.

Because it is based on a light-weight, kernel optimized for virtualization, VMware ESX Server is less susceptible to viruses and other problems that affect general-purpose operating systems. However, ESX Server is not impervious to attack, and you should take proper measures to harden it, as well as the VMware VirtualCenter management server, against malicious activity or unintended damage. This paper provides recommendations for steps you can take to ensure that your VMware Infrastructure 3 environment is properly secured. The paper is divided into sections based upon the components of VMware Infrastructure 3:

The paper also explains in detail the security-related configuration options of the components of VMware Infrastructure 3 and the consequences for security of enabling certain capabilities.

Virtual Machines

The following recommendations apply to the way that virtual machines are configured, as well as interactions with virtual machines.

Secure Virtual Machines as You Would Secure Physical Machines

A key to understanding the security requirements of a virtualized environment is the recognition that a virtual machine is, in most respects, the equivalent of a physical server. Hence the guest operating system that runs in the virtual machine is subject to the same security risks as a physical system. However, attacking an individual virtual machine will result in the compromise of only that virtual machine and not the virtualization server on which that virtual machine is hosted. Therefore, it is critical that you employ the same security measures in virtual machines that you would for physical servers.

In every virtual machine in your virtual infrastructure, install antivirus agents, spyware filters, intrusion detection systems, and any other security measures that you normally would install on a physical server. Make sure to keep all security measures up-to-date, including applying appropriate patches. It is especially important to keep track of updates for dormant virtual machines that are powered off, because it could be easy to overlook them.

Disable Unnecessary or Superfluous Functions

By disabling unnecessary system components that are not needed to support the application or service, you reduce the number of parts that can be attacked. Some of these steps include:

- Disable unused services in the operating system
- Disconnect unused physical devices, such as CD/DVD, floppy, and USB adapters. This is described in the section "Removing Unnecessary Hardware Devices" in chapter 13 of the *Server Configuration Guide*.
- Turn off any screen savers. If using a Linux, BSD, or Solaris guest operating system, do not run the X Window system unless it is necessary.
- Disable copy and paste operations between the guest operating system and remote console so sensitive information is not inadvertently copied over. This is described in the section "Disabling Copy and Paste Operations Between the Guest Operating System and Remote Console" in chapter 13 of the *Server Configuration Guide*.

Take Advantage of Templates

By capturing a hardened base operating system image (with no applications installed) in a template, you can ensure that all your virtual machines are created with a known baseline level of security. You can then use this template to create other, application-specific templates, or you can use the application template to deploy virtual machines. Make sure to keep patches and security measures up-to-date in templates. In VMware Infrastructure 3, you can convert a template to a virtual machine and back again quickly, which makes updating templates quite easy.

Prevent Virtual Machines from Taking Over Resources

VMware ESX Server gives you the ability to control the allocation of host resources with a great deal of granularity. By using the resource management capabilities of ESX Server, such as shares and limits, you can control the server resources consumed by a virtual machine, so that a virtual machine that has been compromised does not affect other virtual machines on the same ESX Server host. You can use this mechanism to prevent a denial of service in which one virtual machine is caused to consume so much of the host's resources that other virtual machines on the same host cannot perform their intended functions.

Limit Data Flow from the Virtual Machine to the ESX Server Host

Virtual machines can write troubleshooting information into a virtual machine log file (`vmware.log`) stored on the VMware VMFS volume. Virtual machine users and processes can be configured to abuse the logging function, either intentionally or inadvertently so that large amounts of data flood the log file. Over time, the log file can consume so much of the ESX Server host's file system space that it fills the hard disk, causing an effective denial of service as the host system can no longer operate.

There are two methods you can use to address this potential problem. The first is to configure the system to rotate or delete log files when they reach a certain size. This option lets you configure the maximum size of the log file. When this size is reached, ESX Server makes an archive copy of the log file and starts a new one. You should also configure the server to maintain a specific number of old log files. When the configured limit is reached, ESX Server automatically deletes the oldest file. By default, ESX Server rotates the log file any time the virtual machine is powered on. However, if you use size-based log file rotation, ESX Server does not rotate the log file until it reaches the size limit, even if you power on the virtual machine. By default, ESX Server maintains six log files. You can enable and configure size-based log file rotation by performing the following steps:

1. Log on to the Virtual Infrastructure Client and select the virtual machine from the inventory panel. The configuration page for this virtual machine appears with the Summary tab displayed.
2. Click **Edit Settings**.
3. Click **Options** > **Advanced** > **Configuration Parameters** to open the Configuration Parameters dialog box.
4. Click the **Add Row** button and type the following:

   Name field: `log.rotateSize`

   Value field: `<maximum size in bytes of log file>`

   This implicitly turns on size-based log file rotation.
5. Click the **Add Row** button again and type the following:

   Name field: `log.keepOld`

   Value field: `<number of log files to keep>`

A log file size of 500KB is recommended in order to provide enough information for reasonable debugging.

The second option is to disable logging for the virtual machine. You can make this change through the VI Client for individual virtual machines. Note that disabling logging for a virtual machine makes troubleshooting challenging and support difficult, so you should not consider disabling logging unless the log file rotation approach is proves insufficient. Disabling logging on a virtual machine is described in the section “Changing Virtual Machine Options” in chapter 10 of the ***Basic System Administration*** manual. Note that disabling logging in this manner does not completely disable all logging messages. VMware Tools messages that are still logged. If you wish to prevent all forms of logging, you must disable these messages separately. The procedure to do so is described in the section “Disabling Logging for the Guest Operating System” in chapter 13 of the *Server Configuration Guide*.

Caution: The *Server Configuration Guide* is misleading on this point. The text in chapter 13 in the section “Disabling Logging for the Guest Operating System” does not mention that the procedure described in that section affects only VMware Tools messages and not general logging messages. If you follow that procedure, you still need to disable logging for messages from sources other than VMware Tools using the procedure described above.

In addition to logging, guest operating system processes can send informational messages to the ESX Server host through VMware Tools. These messages, known as setinfo messages, typically contain name-value pairs that define virtual machine characteristics or identifiers that the host stores — for example, `ipaddress=10.17.87.224`. A setinfo message has no predefined format and can be any length. Therefore, the amount of data passed to the host in this way is unlimited. An unrestricted data flow provides an opportunity for an attacker to stage a DOS attack by writing software that mimics VMware Tools and flooding the host with packets, thus consuming resources needed by the virtual machines.

This mechanism is available by default to facilitate monitoring and troubleshooting. However, you can disable it, thus eliminating any potential denial of service through this route. Disabling the ability to send setinfo messages is described in the section “Preventing the Guest Operating

System Processes from Flooding the ESX Server Host" in chapter 13 of the *Server Configuration Guide*.

Isolate Virtual Machine Networks

Although the virtual hardware of one virtual machine is isolated from that of another virtual machine, virtual machines also are typically connected to shared networks. Any virtual machine or group of virtual machines connected to a common network can communicate across those network links and can, therefore, still be the target of network attacks from other virtual machines on the network. As a result, you should apply network best practices to harden the network interfaces of virtual machines Consider isolating sets of virtual machines on their own network segments to minimize the risks of data leakage from one virtual machine zone to the next across the network.

Network segmentation mitigates the risk of several types of network attacks, including Address Resolution Protocol (ARP) address spoofing, in which an attacker manipulates the ARP table to remap MAC and IP addresses, to redirect network traffic to and from a given host to another unintended destination. Attackers use ARP spoofing to generate denials of service, hijack the target system, and otherwise disrupt the virtual network.

Segmentation has the added benefit of making compliance audits much easier, because it gives you a clear view of which virtual machines are linked by a network.

You can implement segmentation using either of two approaches, each of which has different benefits:

- Use separate physical network adapters for virtual machine zones to ensure that the zones are isolated. Maintaining separate physical network adapters for virtual machine zones is probably the most secure method and is less prone to misconfiguration after the initial segment creation.
- Set up virtual local area networks (VLANs) to help safeguard your network. Because VLANs provide the almost all of the security benefits inherent in implementing physically separate networks without the hardware overhead, they offer a viable solution that can save you the cost of deploying and maintaining additional devices, cabling, and so forth.

For more information on VLANs with virtual machines, see the section "Security Virtual Machines with VLANs" in chapter 10 of the *Server Configuration Guide*.

Minimize use of the VI Console

The VI Console allows a user to connect to the console of a virtual machine, in effect seeing what a monitor on a physical server would show. However, the VI Console also provides power management and removable device connectivity controls, which could potentially allow a malicious user to bring down a virtual machine. In addition, it also has a performance impact on the service console, especially if many VI Console sessions are open simultaneously. Instead of VI Console, use native remote management services, such as terminal services and ssh, to interact with virtual machines.

Service Console

Whether you use a management client or the command line, all configuration tasks for ESX Server are performed through the service console, including configuring storage, controlling aspects of virtual machine behavior, and setting up virtual switches or virtual networks. As with an IPMI or service processor on a physical server, someone logged in to the service console with privileged permissions has the ability to modify, shut down, or even destroy virtual machines on that host. The difference is that, instead of a single physical server, this can affect many virtual machines. While VMware ESX Server management clients use authentication and encryption to prevent unauthorized access to the service console, other services might not offer the same protection. If attackers gain access to the service console, they are free to reconfigure many attributes of the ESX Server host. For example, they could change the entire virtual switch configuration, change authorization methods, and so forth. Because the service console is the point of control for ESX Server, safeguarding it from misuse is crucial.

Isolate the Management Network

Network connectivity for the service console is established through virtual switches. To provide better protection for this critical ESX Server component, VMware recommends that you isolate the service console using one of these methods:

- Create a separate VLAN for management tool communication with the service console.
- Configure network access for management tool connections with the service console through a single virtual switch and one or more uplink ports.

Both methods prevent anyone without access to the service console VLAN or virtual switch from seeing traffic to and from the service console. They also prevent attackers from sending any packets to the service console. As an alternative, you can choose to configure the service console on a separate physical network segment instead. Physical segmentation provides a degree of additional security in that it is less prone to later misconfiguration.

Configure the Firewall for Maximum Security

ESX Server 3 includes a firewall between the service console and the network. By default, the service console firewall is configured at the high security setting, which blocks all incoming and outgoing traffic except for that on ports 902, 80, 443, and 22, which are used for basic communication with ESX Server. Medium security blocks all incoming traffic except on the default ports (902, 433, 80, and 22) and any ports that users specifically open. Outgoing traffic is not blocked. Low security does not block either incoming or outgoing traffic and is equivalent to removing the firewall.

Because the ports open by default are strictly limited, you may need to open additional ports after installation. For example, Veritas NetBackup™ 4.5 backup agent uses ports 13720, 13724, 13782, and 13783. These are used for NetBackup client-media transactions, database backups, user backups, or restores.

Unless you need access for some particular reason, such as enabling backup agents, it is best to leave the host with the high security firewall setting. If you do open ports, make sure to document the changes, including the purpose for opening each port. You configure the service console firewall using the `esxcfg-firewall` command. For more information on how to use this command, please see the section "Changing the Service Console Security Level" in chapter 12 of the *Server Configuration Guide* or type `man esxcfg-firewall` on the command line.

Use VI Client and VirtualCenter to Administer the Hosts Instead of Service Console

The best measure to prevent security incidents in the service console is to avoid accessing it if at all possible. You can perform many of the tasks necessary to configure and maintain the ESX Server host using the VI Client, either connected directly to the host, or, better yet, going through VirtualCenter. The VI Client communicates using a well-defined API, which limits what can be done. This is safer than direct execution of arbitrary commands.

Going through VirtualCenter has the added benefit that authorization and authentication are performed via your standard central Active Directory service, instead of using special local accounts in the service console. In addition, roles and users are stored in a database, providing an easy way to view the current permissions as well as take a snapshot of them. VirtualCenter also keeps track of every task invoked through it, providing an automatic audit trail.

There are some tasks that you cannot perform via the VI Client. For these tasks, you must log in to the service console. Also, if you lose your connection to the host, executing certain of these commands through the command-line interface may be your only recourse — for example, if the network connection fails and you are therefore unable to connect using VI Client. These tasks are described in Appendix A of the *Server Configuration Guide*.

Of course, there may be some cases in which you want to automate certain configuration tasks using scripts that run in the service console, but for interactive administration, VI Client is the most secure access method.

Use a Directory Service for Authentication

Advanced configuration and troubleshooting of an ESX Server host may require local privileged access to the service console. For this circumstance, you should set up individual host-localized user accounts and groups for the few administrators with overall responsibility for your virtual infrastructure. Ideally, these accounts would correspond to real individuals and not be accounts shared by multiple people. Although you can create on the service console of each host local accounts that correspond to each global account, this presents the problem of having to manage user names and passwords in multiple places. It is much better to use a directory service, such as NIS or LDAP, to define and authenticate users on the service console, so you do not have to create local user accounts.

Because service console authentication is Unix-based, it cannot use Active Directory to define user accounts. However, it can use Active Directory to authenticate users. In other words, you can define individual user accounts on the host, then use the local Active Directory domain to manage the passwords and account status. You must create a local account for each user that requires local access on the service console. This should not be seen as a burden; in general, only relatively few people should have access to the service console, so it is better that the default is for no one to have access unless you have created an account explicitly for that user.

Authentication on the service console is controlled by the command `esxcfg-auth`. You can find information on this command in its man page. Type `man esxcfg-auth` at the command line when logged in to the service console. For information on authentication with Active Directory, see the technical note at *www.vmware.com/vmtn/resources/582*.

Strictly Control Root Privileges

Because the root user of the service console has almost unlimited capabilities, securing this account is the most important step you can take to secure the ESX Server host. By default — that is, with the high security firewall setting — all insecure protocols, such as FTP, Telnet, and HTTP, are disabled. Remote access via ssh is enabled, but not for the root account. Files can be copied remotely to and from the service console using an scp (secure cp) client, such as WinSCP.

Enabling remote root access is not recommended, because it opens the system to network-based attack should someone obtain the root password. A better approach is to log in remotely using a regular user account, then use `sudo` to perform privileged commands. The `sudo` command enhances security because it grants root privileges only for select activities, in contrast with the `su` command, which grants root privileges for all activities. Using `sudo` also provides superior accountability because all `sudo` activities are logged, whereas if you use `su`, ESX Server only logs the fact that the user switched to root by way of su. The sudo command also provides a way for you to grant or revoke execution rights to commands on an as-needed basis.

You can go a step further and disallow root access even on the console of the ESX Server host — that is, when you log in using a screen and keyboard attached to the server itself, or to a remote session attached to the server's console. This approach forces anyone who wants to access the system to first log in using a regular user account, then use sudo or su to perform tasks. Ideally, only a limited set of individuals would need permission to run su in order to perform arbitrary administrative tasks. If you decide to disallow root login on the console, you should first create a non-privileged account on this host to enable logins, otherwise you could find yourself locked out of the host. This should be a local account — that is, one that does not require remote authentication — so that in case the network connection to the directory service is lost, access to the host is still possible. You can assure this access by defining a local password for this account, using the passwd command, which will then override authentication via directory services (as discussed in the previous section). The net effect is that administrators can still access the system, but they never have to log in as root. Instead, they use sudo to perform particular tasks or su to perform arbitrary commands.

To prevent direct root login on the console, modify the file /etc/securetty to be empty. While logged in as root, enter the following command:

```
    cat /dev/null > /etc/securetty
```

After you do this, only non-privileged accounts are allowed to log in at the console.

Limiting Access to su

Because su is such a powerful command, you should limit access to it. By default, only users that are members of the wheel group in the service console have permission to run su. If a user attempts to run su - to gain root privileges and that user is not a member of the wheel group, the su - attempt fails and the event is logged.

Besides controlling who has access to the su command, through the pluggable authentication module (PAM) Infrastructure, administrators can determine what type of authentication is required to successfully execute the command. In the case of the su command, the relevant PAM configuration file is /etc/pam.d/su. To allow only members of the wheel group to execute the su command, and then only after authenticating with a password, find the line with auth required and remove the leading pound sign (#) so it reads:

```
    auth required /lib/security/$ISA/pam_wheel.so use_uid
```

You can also use sudo to restrict access to the su command. Usage of sudo is described in the following section

Using sudo

The following list of tips will help you configure sudo:

- Configure local and remote sudo logging (see Logging section below).
- Create a special group, such as vi_admins, and allow only members of that group to use sudo.
- Use sudo aliases to determine the authorization scheme, then add and remove users in the alias definitions instead of in the commands specification.

- Be careful to permit only the minimum necessary operations to each user and alias. Permit very few users to run the su command, because su opens a shell that has full root privileges but is not auditable.
- If you have configured authentication using a directory service, sudo uses it by default for its own authentication. This behavior is controlled by the /etc/pam.d/sudo file, on the line for auth. The default setting — service=system-auth — tells sudo to use whatever authentication scheme has been set globally using the esxcfg-auth command.
- Require users to enter their own passwords when performing operations. This is the default setting. Do not require the root password, because this presents a security risk, and do not disable password checking. In sudo the authentication persists for a brief period of time before sudo asks for a password again.

For further information and guidelines for using sudo, see *www.gratisoft.us/sudo/*.

Establish a Password Policy for Local User Accounts

For any local user accounts that are created, the service console provides password controls on two levels to help you enforce password policies to limit the risk of password cracking:

- **Password aging** — These controls govern how long a user password can be active before the user is required to change it. They help ensure that passwords change often enough that if an attacker obtains a password through sniffing or social engineering, the attacker cannot continue to access the ESX Server host indefinitely.
- **Password complexity** — These controls ensure that users select passwords that are hard for password generators to determine. Instead of using words, a common technique for ensuring password complexity is to use a memorable phrase, then derive a password from it — for example, by using the first letter of each word.

Both of these policies are described in the section "Password Restrictions" in chapter 12 of the *Server Configuration Guide*.

Limit the Software and Services Running in the Service Console

Additional software that could run in the service console includes management agents and backup agents. Services that could run include NIS, SNMP, or CIM HTTPS. Although this software could have a legitimate purpose, be aware that the more components there are running in the service console, the more potential objects are susceptible to security vulnerabilities. In addition, these components often require specific network ports to be open in order to function, thus further increasing the avenues of attack.

By default, an ESX Server host is installed with maximum network security settings, which means that only the absolutely necessary network ports are open, in both incoming and outgoing directions. You should carefully weigh the decision to open additional ports for functionality needed by extra components against the potential risk and your organization's security policies.

For more information and recommendations on running third-party software in the service console, see *www.vmware.com/vmtn/resources/516*.

Do Not Manage the Service Console as a Linux Host

The service console is generated from a Red Hat Linux distribution that has been carefully stripped down and modified to provide exactly the functionality necessary to communicate with and allow management of the VMkernel. Any additional software installed should not make assumptions about what RPM packages are present, nor that they can modify them. In many cases, the packages that do exist have been modified especially for ESX Server.

It is particularly important that the service console not be treated like a Linux host when it comes to patching. Never apply patches issued by Red Hat or any other third-party vendor. Apply only patches that are published by VMware specifically for the versions of ESX Server that you have in use. These are published for download periodically, as well as on an as-needed basis for security fixes. You can receive notifications for security-related patches by subscribing to an RSS feed here: *vmware.simplefeed.net/subscription/*.

The service console also should not be managed like a traditional Linux host. The usual `redhat-config-*` commands are not present, nor are other components such as the X server. Instead, you manage the ESX Server host using a series of purpose-built commands, such as `vmkfstools` and the `esxcfg-*` commands. Many of these commands should be used only upon instruction from VMware Technical Support, or not invoked manually at all, but a few provide functionality that is not available via the VI Client, such as authentication management and advanced storage configuration.

Because ESX Server runs a customized, locked-down version of Linux, there is much less likelihood of security exploits than in a standard Linux distribution. If you follow the best practice of isolating the network for the service console, there is no reason to run any antivirus or other such security agents, and their use is not recommended. However, if your environment requires that such agents be used, then use a version designed to run on Red Hat Enterprise Linux 3, Update 6.

For more information on the special administrative commands in the service console, see Appendices A and B of the *Server Configuration Guide*.

Establish and Maintain File System Integrity

The service console has several files that specify service console configurations:

- `/etc/profile`
- `/etc/ssh/sshd_config`
- `/etc/pam.d/system_auth`
- `/etc/ntp`
- `/etc/ntp.conf`
- `/etc/passwd`
- `/etc/group`
- `/etc/sudoers`
- `/etc/shadow`

In addition, ESX Server configuration files located in the `/etc/vmware` directory store all the VMkernel information.

All of these files should be monitored for integrity and unauthorized tampering, using a tool such as Tripwire, or by using a checksum tool such as `sha1sum`, which is included in the service console. These files should also be backed up regularly, either using backup agents or by doing backups based on file copying.

Maintain Proper Logging

Proper and thorough logging allows you to keep track of any unusual activity that might be a precursor to an attack and also allows you to do a postmortem on any compromised systems and learn how to prevent attacks from happening in the future.

The syslog daemon performs the system logging in ESX Server. You can access the log files in the service console by going to the `/var/log/` directory. Several types of log files generated by ESX Server are shown in the following table:

Component	Location	Purpose
VMkernel	`/var/log/vmkernel`	Records activities related to the virtual machines and ESX Server
VMkernel warnings	`/var/log/vmkwarning`	Records activities with the virtual machines
VMkernel summary	`/var/log/vmksummary`	Used to determine uptime and availability statistics for ESX Server; human-readable summary found in `/var/log/vmksummary.txt`
ESX Server host agent log	`/var/log/vmware/hostd.log`	Contains information on the agent that manages and configures the ESX Server host and its virtual machines
Virtual machines	The same directory as the affected virtual machine's configuration files; named `vmware.log`	Contain information when a virtual machine crashes or ends abnormally
VirtualCenter agent	`/var/log/vmware/vpx`	Contains information on the agent that communicates with VirtualCenter
Web access	`/var/log/vmware/webAccess`	Records information on Web-based access to ESX Server
Service console	`/var/log/messages`	Contain all general log messages used to troubleshoot virtual machines or ESX Server
Authentication log	`/var/log/secure`	Contains records of connections that require authentication, such as VMware daemons and actions initiated by the xinetd daemon.

The log files provide an important tool for diagnosing security breaches as well as other system issues. They also provide key sources of audit information. In addition to storing log information in files on the local file system, you can send this log information to a remote system. The syslog program is typically used for computer system management and security auditing, and it can serve these purposes well for ESX Server hosts. You can select individual service console components for which you want the logs sent to a remote system.

The following tips provide best practices for logging:

- Ensure accurate time-keeping.

  By ensuring that all systems use the same relative time source (including the relevant localization offset), and that the relative time source can be correlated to an agreed-upon time standard (such as Coordinated Universal Time — UTC), you can make it simpler to track and correlate an intruder's actions when reviewing the relevant log files. In the service console, you set the time source using the NTP (Network Time Protocol) system. For instructions on how to configure NTP, see VMware knowledge base article 1339 (*kb.vmware.com/kb/1339*).

- Control growth of log files.

In order to prevent the log file from filling up the disk partition on which it resides, configure log file rotation. This automatically creates a backup of the log file after it reaches a certain specified size and keeps only a specified number of older backup files before automatically deleting them, thus limiting the total disk usage for logging. The log rotation behavior is specified for each component in configuration files located in the directory `/etc/logrotate.d` as well as in the file `/etc/logrotate.conf`.

For the three files in `/etc/logrotate.d` — `vmkernel`, `vmksummary`, and `vmkwarning` — it is recommend that the configuration be modified to:

- Increase the size of the log file to `4096k`.
- Enable compression by setting the line `compress` instead of `nocompress`.

This allows greater logging in the same file system space. For more information on configuring log file rotation, see `man logrotate`.

- Use remote syslog logging.

  Remote logging to a central host provides a way to greatly increase administration capabilities. By gathering log files onto a central host, you can easily monitor all hosts with a single tool as well as do aggregate analysis and searching to look for things like coordinated attacks on multiple hosts.

  An important point to consider is that the log messages are not encrypted when sent to the remote host, so it is important that the network for the service console be strictly isolated from other networks.

  Syslog behavior is controlled by the configuration file `/etc/syslog.conf`. For logs you want to send to a remote log host, add the string `@<loghost.company.com>` after the log name, where `<loghost.company.com>` is the name of a host configured to record remote log files. Make sure that this host name can be properly resolved, putting an entry in `/etc/hosts` if needed.

  Example:

  ```
  local6.warning   /var/log/vmkwarning   @loghost.company.com
  ```

  After modifying the file, tell the syslog daemon to reread it by issuing the following command:

  ```
  kill -SIGHUP `cat /var/run/syslogd.pid`
  ```

- Display different log level messages on different screens.

  An option for syslog is to log to an alternate console, which can be displayed from the terminal of the ESX Server host. ESX Server has the capability at the console to display a number of virtual terminals. This gives you the capability to have critical, error, and warning messages displayed on different screens, enabling you to quickly differentiate types of errors.

  To enable this separation of log message display, add the following lines to the `/etc/syslog.conf` file:

  ```
  *.crit /dev/tty11
  ```

  All log items at the critical level or higher are logged to the virtual terminal at tty11. Press Alt-F11 at the ESX Server console to view these logs.

  ```
  *.err /dev/tty10
  ```

  All log items at the error level are logged to the virtual terminal at tty10. Press Alt-F10 at the ESX Server console to view these logs

```
*.warning /dev/tty9
```

All log items at the warning level are logged to the virtual terminal at tty9. Press Alt-F9 at the ESX Server console to view these logs.

When you are finished, issue the command for rereading the configuration file:

```
kill -SIGHUP `cat /var/run/syslogd.pid`
```

- Use local and remote sudo logging.

  If you have configured sudo to enable controlled execution of privileged commands, you can benefit from using syslog to audit use of these commands. The following instructions show how to log all privileged command executions using syslog. You can then benefit from the other syslog features such as remote logging and log file rotation.

  - Configure sudo to use syslog to record all occurrences of its use. Use the visudo command, which allows you to make changes to the /etc/sudoers configuration file, and add the following line (in the Defaults section, for clarity):

    ```
    Defaults syslog=local2.info
    ```

    The changes in this file take effect immediately

  - Add an entry to /etc/syslog.conf to send the logging information to a file and, optionally, to a remote host.

    ```
    local2.info    /var/log/sudolog
    ```

    or

    ```
    local2.info    /var/log/sudolog    @loghost.company.com
    ```

    Issue the command for rereading the configuration file.

    ```
    kill -SIGHUP `cat /var/run/syslogd.pid`
    ```

- Secure the SNMP configuration

  ESX Server may be configured to act as a client for SNMP. When the SNMP service is enabled in ESX Server, network management tools query it to gather information about the configuration of virtual machines, information about the physical server, etc. In ESX Server 3, only SNMPv1 is supported, and only for queries.

  The ESX Server SNMP package takes the simplest approach to SNMP security in the default configuration. It sets up a single community with read-only access. This is denoted by the ro community configuration parameter in the configuration file for the master snmpd daemon, snmpd.conf. Here are some other changes to make the operation of SNMP more secure.

  - Change the permissions of the snmpd.conf file to 700. Maintain the default ownership of this file: user=root and group=root.
  - Change the SNMP community strings from default to something that is difficult to guess
  - Ensure that SNMP access is restricted to an authorized IP address on your administrative network
  - Do not change the mode from read-only unless you have a specific need and are aware of the implications.

  Consult the snmpd.conf man page for more information on securing SNMP.

ESX Server Host

The following recommendations apply to the way that the ESX Server host itself is configured. Many of the recommendations apply to the configuration of the networks to which virtual machines are attached, because most security attacks occur through network connections. Others pertain to the operation of the ESX Server software itself.

Label Virtual Networks Clearly

Label all your virtual networks appropriately to prevent confusion or security compromises. This labeling prevents operator error due to a virtual machine being attached to a network it is not authorized for or to a network that could allow the leakage of sensitive information.

Do Not Create a Default Port Group

During ESX Server installation, there is an option to create a default virtual machine port. However, this option creates a virtual machine port group on the same network interface as the service console. If this setting is left unchanged, it could allow virtual machines to detect sensitive and often unencrypted information. Since the service console should always be on a separate, private network, this option should never be used except in a test environment.

Use a Dedicated, Isolated Network for VMotion and iSCSI

Because VMotion information is not encrypted, the entire state of a virtual machine could potentially be snooped on the network used for VMotion. Therefore, it is critical that this network be isolated from any other use. If you want to encrypt VMotion traffic, you have the option of using hardware-based SSL encryption. Encryption is not available for iSCSI disk I/O, so you must keep this network strictly controlled, too.

Do Not Use Promiscuous Mode on Network Interfaces

ESX Server has the ability to run virtual network adapters in promiscuous mode. Promiscuous mode may be enabled on virtual switches that are bound to a physical network adapter (vmnic) and virtual switches that do not bind to a physical network adapter (vmnet). When promiscuous mode is enabled for a vmnic switch, all virtual machines connected to the virtual switch have the potential of reading all packets sent across that network, from other virtual machines as well as any physical machines or other network devices. When promiscuous mode is enabled for a vmnet switch, all virtual machines connected to the vmnet switch have the potential of reading all packets across that network — that is, traffic among the virtual machines connected to that vmnet switch.

While promiscuous mode can be useful for tracking network activity, it is an insecure mode of operation because any adapter in promiscuous mode has access to the packets regardless of whether some of the packets should be received only by a particular network adapter. This means that an administrator or root user within a virtual machine can potentially view traffic destined for other guest operating systems. Promiscuous mode should only be used for security monitoring, debugging, or troubleshooting.

By default, promiscuous mode is set to Reject. This option is changed by modifying virtual switch policy on a virtual switch, as described in the section "Layer 2 Security Policy" in chapter 3 of the *Server Configuration Guide*.

Protect against MAC Address Spoofing

Each virtual network adapter in a virtual machine has its own initial MAC address assigned when the adapter is created. In addition, each adapter has an effective MAC address that filters out incoming network traffic with a destination MAC address different from the effective MAC address.

Upon creation, a network adapter's effective MAC address and initial MAC address are the same. However, the virtual machine's operating system can alter the effective MAC address to another value at any time. If an operating system changes the effective MAC address, its network adapter then receives network traffic destined for the new MAC address. The operating system can send frames with an impersonated source MAC address at any time. Thus, an operating system can stage malicious attacks on the devices in a network by impersonating a network adapter authorized by the receiving network. You can use virtual switch security profiles on ESX Server hosts to protect against this type of attack by setting two options, which are set for each virtual switch:

- **MAC address changes** — By default, this option is set to Accept, meaning that ESX Server accepts requests to change the effective MAC address to other than the initial MAC address. The MAC Address Changes option setting affects traffic received by a virtual machine.

  To protect against MAC impersonation, you can set this option to Reject. If you do, ESX Server does not honor requests to change the effective MAC address to anything other than the initial MAC address. Instead, the port that the virtual adapter used to send the request is disabled. As a result, the virtual adapter does not receive any more frames until it changes the effective MAC address to match the initial MAC address. The guest operating system does not detect that the MAC address change has not been honored.

- **Forged transmissions** — By default, this option is set to Accept, meaning ESX Server does not compare source and effective MAC addresses. The Forged Transmits option setting affects traffic transmitted from a virtual machine.

  If you set this option to Reject, ESX Server compares the source MAC address being transmitted by the operating system with the effective MAC address for its adapter to see if they match. If the addresses do not match, ESX Server drops the packet. The guest operating system does not detect that its virtual network adapter cannot send packets using the impersonated MAC address. ESX Server intercepts any packets with impersonated addresses before they are delivered, and the guest operating system might assume that the packets have been dropped.

It is recommended that both of these options be set to Reject for maximal security. To learn how these options are configured, see the section "Layer 2 Security Policy" in chapter 3 of the *Server Configuration Guide*.

Secure the ESX Server Console

Even if you have locked down ESX Server to protect it from attacks that arrive over the network, anyone with access to the console of the host might still cause problems. Although physical harm to the host cannot be prevented, it still might be possible, for example, to influence the host so that it behaves improperly, perhaps in a manner that is hard to detect.

One way to guard against this is to use grub passwords to prevent users from passing options to the kernel during boot. Unless the password is known, the server boots only the kernel with the default options. For more information on grub passwords, see the GNU Grub Manual at *www.gnu.org/software/grub/manual/html_node/index.html*.

Mask and Zone SAN Resources Appropriately.

Zoning provides access control in a SAN topology; it defines which host bus adapters (HBAs) can connect to which SAN device service processors. When a SAN is configured using zoning, the devices outside a zone are not visible to the devices inside the zone. In addition, SAN traffic within each zone is isolated from the other zones. Within a complex SAN environment, SAN switches provide zoning, which defines and configures the necessary security and access rights for the entire SAN.

LUN masking is commonly used for permission management. LUN masking is also referred to as selective storage presentation, access control, and partitioning, depending on the vendor. LUN masking is performed at the storage processor or server level; it makes a LUN invisible when a target is scanned. The administrator configures the disk array so each server or group of servers can see only certain LUNs. Masking capabilities for each disk array are vendor specific, as are the tools for managing LUN masking.

You should use zoning and LUN masking to segregate SAN activity. For example, you manage zones defined for testing independently within the SAN so they do not interfere with activity in the production zones. Similarly, you could set up different zones for different departments. Note that zoning must take into account any host groups that have been set up on the SAN device.

Protect against the Root File System Filling Up

When you install ESX Server 3, you should accept the recommended disk partitioning for the most effective installation. If you choose to partition the disk manually, you should ensure that you have created separate partitions for the directories `/home`, `/tmp`, and `/var/log`. These are all directories that have the potential to fill up, and if they are not isolated from the root partition, you could experience a denial of service if the root partition is full and unable to accept any more writes. Appendix B of the *Installation and Upgrade Guide* covers disk partitions in more detail.

VirtualCenter

VirtualCenter provides a powerful way to manage and control your VMware Infrastructure from a central point and enables more sophisticated operations through tools that work through its SDK. It is extremely powerful and therefore should be subject to the strictest security standards.

Set Up the Windows Host for VirtualCenter with Proper Security

Because VirtualCenter runs on a Windows host, it is especially critical to protect this host against vulnerabilities and attacks. The standard set of recommendations applies, as it would for any host: install antivirus agents, spyware filters, intrusion detection systems, and any other security measures. Make sure to keep all security measures up-to-date, including application of patches.

Limit Administrative Access

VirtualCenter runs as a user that requires local administrator privilege and must be installed by a local administrative user. To limit the scope of administrative access, avoid using the Windows Administrator user to run VirtualCenter after you install it. Instead, use a dedicated VirtualCenter administrator account.

- Create a local VirtualCenter administrator account as an ordinary user that will be used to manage VirtualCenter.
- In VirtualCenter, log on as the Windows Administrator, then grant VirtualCenter root administrator access to the newly-created account

- Log out of VirtualCenter, then make sure you can log in to VirtualCenter as the new user and that this user is able to perform all tasks available to a VirtualCenter administrator
- Remove the permissions in VirtualCenter for the local Administrators group.

By configuring accounts in this way, you avoid automatically giving administrative access to domain administrators, who typically belong to the local Administrators group. You also provides a way of getting into VirtualCenter when the domain controller is down, because the local VirtualCenter administrator account does not require remote authentication.

Limit Network Connectivity to VirtualCenter

The only network connection VirtualCenter requires is to the ESX Server service console and to a network on which instances of VI Client are running. You should avoid putting the VirtualCenter server on any other network, such as your production or storage networks. Specifically, VirtualCenter does not need access to the network on which VMotion takes place. By limiting the network connectivity, you cut down on the possible avenues of attack.

Use the following guidelines to limit network connectivity:

- Firewalls

  You should protect the VirtualCenter server using a firewall. This firewall may sit between the clients and the VirtualCenter server, or both the VirtualCenter Server and the clients may sit behind the firewall, depending on your deployment. The main consideration is ensuring that a firewall is present at what you consider to be an entry point for the system as a whole.

  For more information on the possible locations for firewalls used with VirtualCenter, see the section “Firewalls for Configurations with a VirtualCenter Server” in chapter 10 of the *Server Configuration Guide*.

- TCP and UDP ports for management access

  Networks configured with a VirtualCenter server can receive communications from several types of clients: the VI Client, VI Web Access, or third-party network management clients that use the SDK to interact with the host. During normal operation, VirtualCenter listens on designated ports for data from the hosts it is managing and from clients. VirtualCenter also assumes that the hosts it is managing listen for data from VirtualCenter on designated ports. If a firewall is present between any of these components, you must ensure that the appropriate ports are open to support data transfer through the firewall.

  The section “TCP and UDP Ports for Management Access” in chapter 10 of the *Server Configuration Guide* lists all the predetermined TCP and UDP ports used for management access to your VirtualCenter server, ESX Server hosts, and other network components. Study this section carefully to determine how to configure your firewalls to maintain maximum security while still allowing required management operations.

  Note: You might not be able to open a VI Client remote console when your network is configured such that a firewall using NAT stands between the ESX Server host and the computer running VI Client. See VMware knowledge base article 749640 (*kb.vmware.com/kb/749640*) for a workaround for this issue.

Ensure Proper Security Measures Are Used when Configuring the Database for VirtualCenter

You should install the VirtualCenter database on a separate server or virtual machine and subject it to the same security measures as any production database. You should also carefully configure the permissions used for access to the database to the minimum necessary. Use the guidelines appropriate to your database:

- SQL Server

  During installation and upgrade, the VirtualCenter account must have the DB Owner role. During normal operations, you may further restrict permissions to the following:

  - Invoke/Execute Stored Procedures
  - Select Update, Insert
  - Drop

- Oracle

  During installation, upgrade, and normal operation, the VirtualCenter account needs CONNECT and RESOURCE privileges only.

  Caution: The *Installation and Upgrade Guide* states that CONNECT and DBA privileges are needed. That statement is incorrect.

Note: The ODBC database password is stored in cleartext on the VirtualCenter server.

Enable Full and Secure Use of Certificate-based Encryption

All versions of VMware products, including all releases of VirtualCenter Server use X.509 certificates to encrypt session information sent over SSL (secure sockets layer protocol) connections between server and client components. However, in earlier versions of VirtualCenter, the client did not verify the authenticity of the server certificate presented during the SSL handshake phase (prior to encryption), thus leaving clients vulnerable to man-in-the-middle attacks. VirtualCenter 2.0.1 Patch 1, VirtualCenter 1.4.1 Patch 1, VirtualCenter 1.3.1 Patch 2, and subsequent releases resolve this issue for Windows clients, adding support for the proper client behavior during the SSL handshake. Therefore, it is critical that you upgrade VirtualCenter to the latest patch level in order to use certificate-based encryption.

During the installation of VMware products, default, self-signed certificates are automatically generated. However, the default certificates generated by VirtualCenter up to and including version 2.0.1 Patch 1 are defective and should not be used. By contrast, the default certificates generated by ESX Server hosts are valid and can be used as-is. This requires that any VI Client that wishes to connect to ESX Server directly (that is., without going through VirtualCenter), must pre-trust the default certificates.

For environments that require strong security, VMware recommends that administrators replace all default self-signed certificates generated at installation time with legitimate certificates signed by their local root certificate authority or public, third-party certificates available from multiple public certificate authorities. You should also enable server-certificate verification on all VI Client installations and the VirtualCenter host. This involves a modification to the Windows registry on all client hosts.

Note: You need to replace the default VirtualCenter Server certificate before enabling server-certificate verification.

For background and information on replacing VirtualCenter Server certificates, see the technical note at *www.vmware.com/vmtn/resources/658*. For information on enabling server-certificate verification for VI Client installations, including how to pre-trust certificates and how to modify the Windows registry for client hosts, see VMware knowledge base article 4646606 (*kb.vmware.com/kb/4646606*).

Make Use of VirtualCenter Custom Roles

VirtualCenter 2.0 added a sophisticated system of roles and permissions, to allow fine-grained determination of authorization for administrative and user tasks, based on user or group and inventory item, such as clusters, resource pools, and hosts. You should take advantage of this system to assure that only the minimum necessary privileges are assigned to people in order to prevent unauthorized access or modification. Some recommendations are:

- Create roles that enable only the necessary tasks. For example, a user who is only going to make use of an assigned virtual machine assigned might need permission only to power the machine on or off, and not necessarily to attach a CD or floppy device.
- Assign roles to as limited a scope as necessary. For example, you can give a user certain permissions on a resource pool instead of a discrete host, and you can use folders to contain the scope of a privilege.

For more information on VirtualCenter roles, see the *Basic Administration Guide*.

Document and Monitor Changes to the Configuration

Although most of a VMware Infrastructure environment is defined by information contained in the VirtualCenter database, certain important configuration information resides only on the VirtualCenter Server host's local file system. This includes the main configuration file `vpxd.cfg`, various log files, and, implicitly, the Windows registry settings that pertain to VirtualCenter.

For compliance and auditing, it is important that you have a record of these configurations over time. One convenient way to capture everything in one place is to use the **Generate VirtualCenter Server log bundle** command, in the **VMware program file** menu on the VirtualCenter host. This tool was designed to capture information to be used for troubleshooting and debugging, but the resulting archive file serves as a convenient way to maintain a historical record.

The resulting ZIP archive includes:

- `licmgr_reg.txt`, `odbc_reg.txt`, `vmware_reg.txt` — all the relevant Windows registry entries
- `vpxd.cfg` — the main VirtualCenter Server configuration file (in XML format)
- `vpxd-*.log` — log files for VirtualCenter Server
- `lmgrd.log` — log file for the license server (if present)

By performing this task on a regular basis, you can keep track of all changes that affect your VirtualCenter installation.

References

Server Configuration Guide
www.vmware.com/pdf/vi3_server_config.pdf

Basic System Administration
www.vmware.com/pdf/vi3_admin_guide.pdf

Installation and Upgrade Guide
www.vmware.com/pdf/vi3_installation_guide.pdf

Enabling Server-Certificate Verification for Virtual Infrastructure Clients
kb.vmware.com/kb/4646606

Replacing VirtualCenter Certificates
www.vmware.com/vmtn/resources/658

VMware ESX Server: Third-Party Software in the Service Console
www.vmware.com/vmtn/resources/516

Sudo Main Page
www.gratisoft.us/sudo

GNU Grub Manual
www.gnu.org/software/grub/manual/html_node/index.html

Enabling Active Directory Authentication with ESX Server
www.vmware.com/vmtn/resources/582

Installing and Configuring NTP on VMware ESX Server
kb.vmware.com/kb/1339

VI Client Cannot Open Remote Console Session on ESX Server 3.0
kb.vmware.com/kb/749640

About the Author

Charu Chaubal is technical marketing manager at VMware, where he specializes in enterprise datacenter management. Previously, he worked at Sun Microsystems, where he had more than seven years' experience designing and developing distributed resource management and grid infrastructure software solutions. He has also developed and delivered training courses on grid computing to a variety of customers and partners in the United States and abroad. Chaubal received a Bachelor of Science in Engineering from the University of Pennsylvania, and a Ph.D. from the University of California at Santa Barbara, where he studied the numerical modeling of complex fluids. He is the author of numerous publications and several patents in the fields of datacenter automation and numerical price optimization.

Acknowledgements

The author would like to thank Kirk Larsen, Jim Weingarten, and Jason Mills for their valuable technical knowledge, which made this paper possible.

VMware, Inc. 3145 Porter Drive Palo Alto, CA 94304 www.vmware.com