Chapter 5

Clustering and Security

The VMware Virtual Environment supports several forms of VMware ESX or ESXi clusters. Some are the strictly traditional shared disk clusters, and others are more advanced, enabling the movement of running virtual machines from host to host or storage device to storage device. Each of these is a form of cluster. This chapter discusses how the use of these clusters can and will affect and be affected by security. However, before we launch into the security aspects of clustering we need to consider the different types of clusters involved.

Clusters bring into the fore combined security issues and encompass many other issues covered within this book. For example, you cannot talk about clusters without delving into storage issues, which are covered in Chapter 4, “Storage and Security,” or virtual networking, which is covered in Chapter 9, “Virtual Networking Security.” But in order to think of these issues, it is important to realize that ESX without a cluster is missing quite a bit of important functionality. How this functionality is used will impact your security decisions for these other issues. Because this is the case, it seems logical to place this chapter early in the book.

Most, if not all, virtual environments are composed of clusters, as well. I know of a few single virtualization server installations used within a data center. But they are few and far between.

Types of Clusters

In reality, four types of clusters are possible when using VMware ESX or ESXi. A **cluster** is defined as the sharing of computer, network, and storage resources.

The clusters of concern are shared storage clusters, hardware level clustering, vmware clusters, and virtual machine clusters. Many of these share the same features but are presented within the virtual environment at different levels. Because virtualization server clusters are prevalent within the data center, we'll delve into each type of cluster.

Standard Shared Storage

The first cluster type is a standard shared storage cluster, shown in Figure 5.1. To share storage, some form of communication must occur between the nodes of the cluster, so that they know when the shared storage metadata (definition of the files on the shared storage) has been updated. This is usually accomplished by using a cluster-aware file system within the operating system, but it can be achieved using specialized daemons. In the case of VMware ESX or ESXi, this is achieved using two distinct file systems. The first is VMFS, which is a cluster-aware file system, and the other is NFS, which is a network-aware file system that uses daemons to notify the NFS Server that data has been modified. We discuss these file systems in detail in Chapter 4. A shared file system for a virtualization server makes available two important virtualization features. These features do not require a cluster to be defined within the management tools to be of use, but they do require shared storage of some form in order to work.

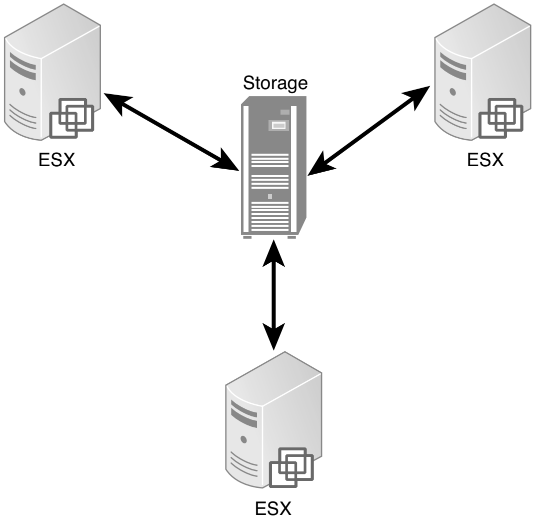


Figure 5.1 Shared storage cluster

VMotion

VMotion is one of the more useful features of a virtual environment. VMotion enables the movement of a VM from node to node without the need to power down the VM, as illustrated by Figure 5.2. The underlying virtual disk does not change its location, but the VM will now execute on a new host. VMotion copies the virtual machines in use memory footprint from one node to another over a network connection that should be dedicated to VMotion.

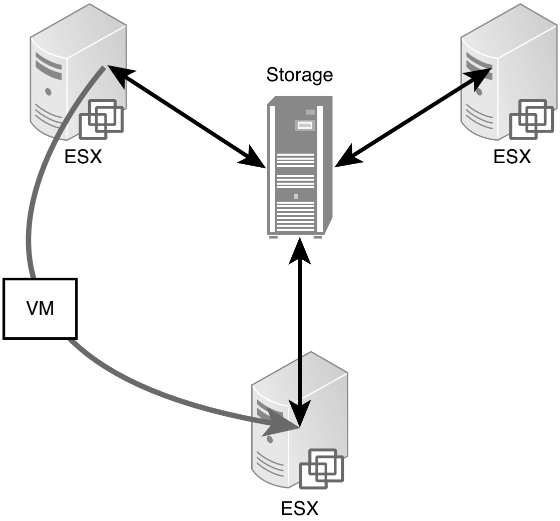


Figure 5.2 VMotion within cluster

As of VMware Virtual Infrastructure 3, VMotion can be routed. This clear-text protocol can now be routed between distinct networks using a VMotion specific network. In fact, VMotion requires the creation of a vmkernel portgroup on a vSwitch, and this portgroup must be assigned to VMotion. Although routing is possible, note that you still want VMotion to finish as fast as possible because not every guest operating system can handle delays due to network latency.

VMotion can be performed between virtualization servers that are not within the same processor family as long as the VM is defined as running a 32-bit guest OS. They achieve this by masking off portions of the CPU to increase the compatibility between the nodes for VMotion. However, if the mask is set up improperly, the VMotion will not be allowed to happen. Other protections built in to VMotion will keep a VM running when a VMotion could cause a failure. Possible points of failure include the following:

- CPU masks do not match.
- CD-ROM is connected.
- Target server does not contain the proper networks.
- Source vSwitch has no active pNIC.
- Floppy is connected.
- Serial port pass through is in use.
- SCSI device pass through is in use.
- In the case of a 64-bit VM, the target system does not share the same processor type and family.

A number of warnings also can be produced when VMotion is attempted. It is very important to review each warning in case one of these warnings could cause VMotion to fail. To reduce the number of SCSI reservation conflicts, VMware has a default limit of four simultaneous VMotions per VC Datacenter.

VMotion with Private vSwitches

There is a way to alleviate the issue of not being able to VMotion, if the source vSwitch does not contain an active pNIC, which is referred to as a private vSwitch. This is often required if you employ virtual firewalls to add security to the virtual machines on the network. When a vFW is employed, it is placed between two vSwitches, where the outer vSwitch is connected to a wire, and the inner vSwitch may not be, as is shown in Figure 5.3. In Figure 5.3 we can see that the inner vSwitch, the one with the dot box, has no connection to the external pNIC; it has no external interface and therefore is a private vSwitch.

If you modify the file `C:\Document and Settings\All Users\Application Data\VMware\VMware VirtualCenter\vpxd.cfg` on your Virtual Center server with the following additions, you can enable vMotion for these virtual networks. Note that these changes should be made before the closing `</config>` tag within the XML file.

```
<migrate>
  <test>
    <CompatibleNetworks>
       <VMOnVirtualIntranet>false</VMOnVirtualIntranet>
    </CompatibleNetworks>
  </test>
</migrate>
```

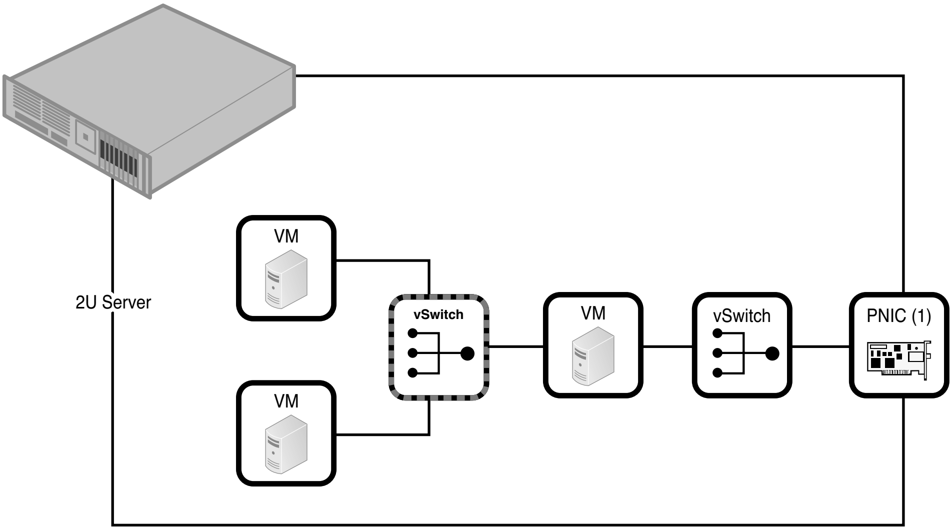


Figure 5.3 vFW implementation

Storage VMotion

Storage VMotion (SVM) is just like VMotion, but it will move the virtual disk from one storage device to another without powering down the VM. As shown in Figure 5.4, the target storage device must be seen by at least one VMware ESX or ESXi host over a network connection. In this case a device could be a LUN presented by the same SAN or iSCSI server or one presented by a different SAN or iSCSI Server to the host.

SVM will fail if the VM has snapshots, non-independent mode virtual disks, or raw disk maps. In addition, enough resources must be available on the host to run two images of the same VM at least temporarily. If there are not enough resources, SVM will fail.

There are a combined default maximum of four simultaneous SVM and VMotion instances possible per VC Datacenter.

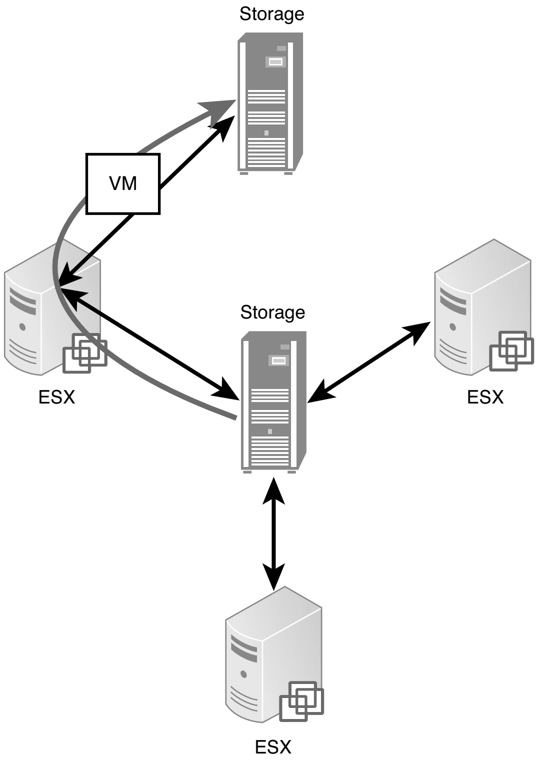


Figure 5.4 Storage VMotion

RAID Blade

Some blade enclosures support the concept of a RAID blade. When the blades share the same enclosure, if one blade fails the other blade can boot from the same disks (usually within a disk blade). This is very powerful single enclosure failover. The HP C-Class blade enclosures share this capability. However, the boot disks for the blades need to be in a special disk blade and not using the disks within the blades. This is depicted in Figure 5.5, where both C-Class blades have their boot disks on the disk blade; when one fails, the other automatically boots using the same LUN on the disk blade.

A less-automatic version of this would be virtualization hosts that "boot-from-SAN." If the virtualization host hardware dies, it is possible to manually put a new host in its place, enable boot from SAN, boot the host, and start running VMs once more. However, to do this the new host must be identical to the existing host, or you'll need to do quite a bit of fix up.

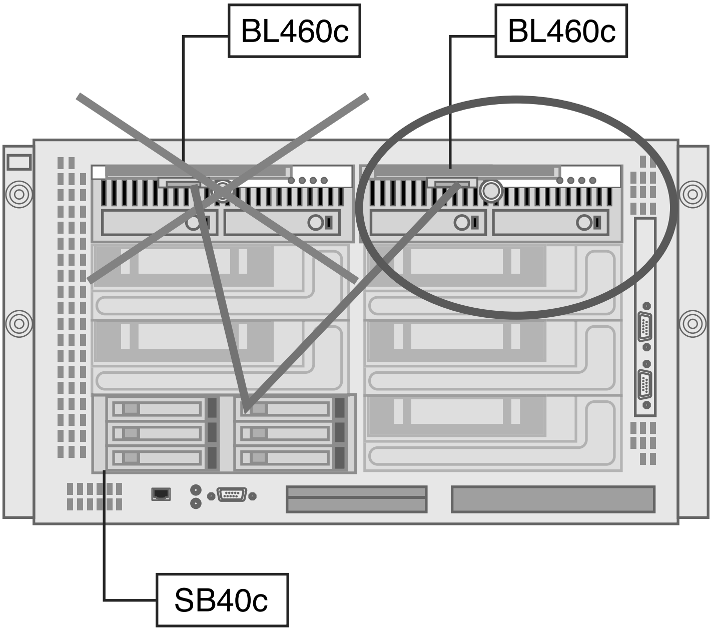


Figure 5.5 RAID blade

VMware Cluster

Add-on products are available from VMware that form what they refer to as a VMware Cluster. Without VMware Clusters, all resources are considered to end at the host boundaries. A VMware cluster enables resources to be pooled together between the member hosts. The pooled resources are CPU and Memory. Disk and network resources are still bound by the host. The pooling of resources does not yet enable VMs to straddle hosts, but they can be used to automatically move VMs around the cluster, restart VMs as needed, and dynamically power off host instances during nonpeak hours.

High Availability (HA)

VMware HA detects when a host or individual VM fails. Failed individual VMs are restarted on the same host. Yet if a host fails, VMware HA will by default boot the failed host's VMs on another running host. This is the most common use of a VMware Cluster, and it protects against unexpected node failures.

Dynamic Resource Scheduling (DRS)

VMware DRS is another part of a VMware Cluster that will alleviate CPU and memory contention on your hosts by automatically VMotioning VMs between nodes within a cluster. If there is contention for CPU and memory resources on one node, the VM can automatically be moved to another underutilized node using VMotion. This requires a standalone VMotion License or an Enterprise License.

Distributed Power Management (DPM)

The experimental VMware DPM will allow nodes within a VMware Cluster to migrate their VMs and power down during off hours. During peak hours they can be powered on and again become active members of the VMware cluster when they are needed. DPM requires wake on LAN functionality on the VMware ESX service console pNIC (VMware ESXi management pNIC) in order to be used. DPM is actually a feature of DRS, expanding its functionality to the machine level.

Enhanced VMotion Capability (EVC)

VMware EVC ties into the Intel FlexMigration and AMD-V Extended Migration capabilities to present to the VMware Cluster members the same CPU feature sets to the VMs running on the EVC enhanced nodes. Each CPU in use on a system contains a set of enhanced features; Intel-VT is one of these. In addition, instructions available to one chipset may be interpreted differently on another chipset. For VMotion to work, these feature sets must match. To do this, a VM set of CPU masks can be set to match up feature sets between disparate CPUs and chipsets. EVC does this at the host level, instead of the per VM level. Unfortunately, EVC will work only between Intel CPUs that support Intel Flex Migration or between AMD CPUs that support Extended Migration. You cannot use EVC to move VMs between AMD and the Intel family of processors. Enabling EVC can be a bit tricky. For example, on the HP DLx80 hardware you must enable Intel-VT and No eXecute (NX) Bits within the BIOS. Each vendor will have its own settings that are needed to enable this feature of VMware Clusters. To enable EVC, you do not need any special licensing; however, you must create a VMware Cluster. After a host is part of that cluster, you must power cycle (not just reboot) all VMs on the virtualization host.

Fault Tolerance (FT)

VMware vSphere 4 Fault Tolerance creates a shadow copy of a VM whose virtual CPUs are kept in lock step with the master CPU employing the VMware vLockStep functionality. VMware FT depends on the VM residing on storage that VMware ESX or ESXi hosts can access, as well as other restrictions on the type and components of the VM (for example, support exists for only one vCPU VMs). In addition to shared datastores the hosts involved in FT must participate in the same VMotion network.

Host Profiles and Distributed Virtual Switch

VMware vSphere 4 provides Host Profiles and Distributed Virtual Switches to aid in configuration management by allowing vSphere host configurations and virtual switch configurations to be the same across all hosts within the cluster. However, this does mean that it is also simpler to catastrophically affect all nodes on the cluster. Use of these tools should be tied into your change management process.

Virtual Machine Clusters

It is also possible to cluster virtual machines within the virtualization environment. The virtual machines can be clustered between virtualization server nodes, within a single node, and between a virtual node and a physical node. Multiple types of clusters are also supported, including network load balanced (NLB) clusters and shared disk clusters. Microsoft Cluster Services (MSCS) and Redhat Clusters are just two examples of virtual machine clusters available.

Security Concerns

Now that we have reviewed the basics that define what composes a cluster within the virtual environment, we need to look further into the security of the cluster elements. In our definitions of threat, vulnerability, and fault from Chapter 1, “What Is a Security Threat?” we know that any failure of a node within a cluster should be considered from a security perspective. Although some failures are easy to track to the root cause, that is not always the case. That is when a security analysis of a fault should be performed in conjunction with normal fault determination.

For example, a recent crash of a system was easy to spot after we opened up the system and determined that a heat sink was not properly attached. However, if we did not have access to the box, or if the heat sink looked attached, would we have automatically assumed the failure was due to hardware? In many cases, we would have, but not always. Could it have been a malicious attack? Yet this unexpected failure did not force VMware HA to fail the VMs over to the other nodes in the cluster as we expected. What was the root cause of this VMware HA failure? Could this have been a security issue? Although we will give the answer to this question further on as we explore the parts of the cluster from a security perspective, the general answer is to correlate events within networking, storage, operational, hardware, and VMware log files to find the culprit.

Clusters are one way to mitigate possible failures by either rapidly booting virtual machines or transferring the load from busy systems to less used systems. Business continuity and failover are part of any security architecture because they are employed to mitigate the unknown problems that occur within the data center. The goal is to keep systems running.

If failover does occur for some reason, this is when we may have to look at things from a security perspective. Why a node of a cluster crashed, a VM was moved from node to node, or a VM was using more resources than normal could be security concerns and point to a more severe problem. This is not always the case, but it could be the start of an attack.

Process accounting has always been just one part of security research and should remain so within the virtual world. Process accounting is the gathering of data about all processes running within your VMs and virtualization hosts (which include the VMs). Such data would be the length of time a process took to run, which CPUs and other devices were in use, and so on. With clusters of virtualization servers, process accounting needs to now include full virtual machine data and not just the single process running. The performance data stored by the virtual center could be an invaluable research tool that could lead to recognizing a security issue. This illustrates the importance of gathering baseline data. The tool often used to gather this data will be the `vm-support` command for each virtualization host, or you can export diagnostic data when using the VIC.

Clifford Stoll wrote about his research into computer espionage within the book *Cuckoo's Egg* (New York: Pocket Books, 1990). In this real-life story, a $0.75 accounting discrepancy on a time-share system led to the capture of a worldwide computer espionage ring. This one discrepancy shows that something apparently minor could be the tip of the iceberg. This is an important point, and a good illustration. If you don't have an idea of what your baseline is and how this compares with current data, you will never know there was a security problem.

We will delve into some specific issues about clusters that could be cause for security concern. It is important to realize that many subsystems are secured by other means as part of our strategy of defense in depth. However, knowing about the threats to clusters will give your security measures more importance. Although this is not an exhaustive list covering clustering, it does cover some of the more prevalent concerns.

Heartbeats

A major part of any cluster is the way it communicates between its member nodes. This communication will control what happens within the cluster. It is also an attack point for cluster Denial of Service (DoS) attacks as well as possibly an unauthenticated way into the system. When dealing with VMware clusters, you should be concerned with two heartbeats: SCSI reservations and service console or management appliance network connectivity.

SCSI Reservations

SCSI reservations are looked at in detail within Chapter 4. Because the VMFS is a cluster-aware file system, SCSI reservations are used to protect VMware ESX or ESXi hosts from colliding on changes to the file system metadata. SCSI reservation releases are also a means of notifying a virtualization host that an update to the metadata has been made. The current revisions of VMware ESX limit per LUN SCSI actions so that there is minimal impact on the SCSI reservation subsystem. However, when it comes to clusters of VMware ESX or ESXi hosts, either as a VMware Cluster or just sharing drives, SCSI reservations become quite important.

Even with the protections against too many SCSI reservations within the current versions of VMware ESX and ESXi hosts, it is still possible to inundate the all important storage subsystem with more requests than it can handle. This is dependent on the SAN, iSCSI Server, and actions taken. If SCSI reservation conflicts occur, they are generally caused by direct action by a script or user attempting to manipulate the metadata of the clustered file system. However, it is also possible for the storage subsystem to be overloaded, and this will also produce SCSI reservation errors as well as other failure errors. With the introduction of NPIV, it is now possible for VMs to generate SCSI reservations requests. However, without NPIV, SCSI reservation requests cannot occur from within VMs. They can be issued only via the management tools or from within the VMware ESX or ESXi consoles when administrative or other necessary actions take place that change a VMFS's metadata.

To see SCSI reservation conflicts, look in the /var/log/vmkernel log file for lines similar to the following. The source of these errors could be a form of a Denial of Service because of a script that is running out of control, either accidentally or perhaps for more nefarious reasons. To get this information out of VMware ESXi, you will either have to redirect logging to a remote host or use the VIC to export diagnostic data from the VMware ESXi host. If you go the route of using the VIC, you will then need to unpack the diagnostic data to get to the proper logfile.

```
May 10 02:01:25 aurora02 vmkernel: 1:12:11:30.946 cpu7:1078)SCSI: 4782:
path vmhba1:0:1: Passing device status RESERVATION_CONFLICT (18) through
May 10 02:02:03 aurora02 vmkernel: 1:12:12:09.517 cpu6:1091)SCSI: 4782:
path vmhba1:0:1: Passing device status RESERVATION_CONFLICT (18) through
May 10 02:02:15 aurora02 vmkernel: 1:12:12:21.237 cpu7:1081)SCSI: 4782:
path vmhba1:0:1: Passing device status RESERVATION_CONFLICT (18) through
```

I have found that many of these errors occur because of users doing too many management items that change the VMFS metadata at once, a misunderstanding of what constitutes a change to the metadata via some scripting mechanism, a script that uses incorrect assumptions, impractical configurations, or too many nodes attached to any single LUN within the storage fabric. Yet with increased functionality within the VM such as N_port ID Virtualization (NPIV), it is possible for the VM to impact the storage subsystem. These impacts could be caused by malicious activity or normal activity. Because it could be malicious, I consider all VMs to be a hostile environment with respect to the virtualization host.

Service Console vswif (ESXi Management Console NIC)

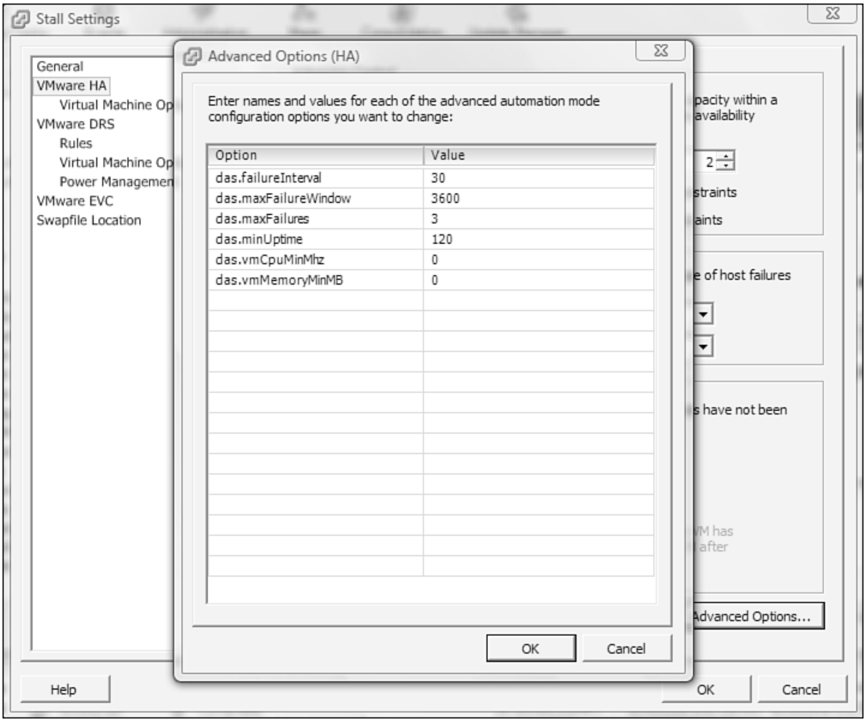
The service console network interface—vswif0—or the VMware ESX and ESXi console NIC can also be a source of heartbeat communication for VMware Clusters. These network links are used to transmit the heartbeat used by VMware HA, which is composed of EMC Automated Availability Manager (Legato AAM). If and when these links go down, VMware HA will kick in and boot the VMs on the other hosts per the rules you set. In addition, it is possible to use VMware HA to monitor and reboot individual VMs on a host.

A couple of issues could affect this heartbeat and prematurely kick off a reboot of the VMs on new nodes.

The first occurs when the service console VM dies on a VMware ESX host. The service console can crash independently of the VMs as the service console is a VM. When this occurs, VMware HA loses heartbeat, and the VMs that are already running on the existing host are booted on a new node. This could cause two identical VMs to be running at the same time and cause IP confusion at the very least; generally, however file locking prevents this from occurring. The service console can crash for various reasons, but the most general cases are due to hardware issues that somehow do not affect the vmkernel. Other ways could force the service console to crash. If the service console was compromised, it is extremely easy to force a crash of this VM. If that happens there is no method to manage the vmkernel, which causes the rare case of requiring some form or remote access to the VMs to safely power them off, so you can reboot the virtualization host. This is not the case with VMware ESXi; if the management appliance for VMware ESXi crashes, the entire server also crashes, including the VMs.

The second occurs when the network connection to the console dies either through switching fabric issues (legitimate downtime, MAC Address spoofing within the physical network, ARP cache poisoning attacks, and so on), or an unexpected switching fabric failure (bad cables, bad pNICs, switch failures, and the like). It is possible for this to occur if VMware HA is configured to be too sensitive. It was not possible prior to VMware ESX v3.5.0 Update 1 to even change the sensitivity of VMware HA. The default is to look for three missing heartbeats, which takes roughly 54 seconds (3 seconds for each heartbeat + seconds to wait before declaring the other node dead). After the three missing heartbeats, VMware HA will do as you directed it to do and either shut down the running VMs on the now isolated host and boot the VMs on other hosts within the VMware Cluster, or keep the VMs running but still attempt to boot them on the other host.

It is important to manage this setting so that your requirements for failover are covered. If you know that network activity will take more than 20 seconds, you may want to increase the timeout value. If this is overlooked, failover will occur. To set this value to something higher, modify the advanced options of the VMware HA configuration (see Figure 5.6) by adding or modifying `das.maxFailures` from the default of three failures. Another option would be to set failure interval to be greater than its default of 30 seconds (`das.failureInterval`).



Enter names and values for each of the advanced automation mode configuration options you want to change:

Option	Value
das.failureInterval	30
das.maxFailureWindow	3600
das.maxFailures	3
das.minUptime	120
das.vmCpuMinMhz	0
das.vmMemoryMinMB	0

Figure 5.6 VMware HA advanced configuration

Other possible options to set are shown in Table 5.1.

Table 5.1

VMware HA Advanced Options

Option	Value
das.failuredetecttime	Number of milliseconds (default 15000) to wait before declaring the other host dead.
das.failuredetectioninterval	Number of milliseconds (default 1000) to use as the heartbeat interval.
das.poweroffonisolation	If set to false will leave all VMs powered on the host in the case of isolation.
das.usedefaultisolationaddress	If set to false then only das.isolationaddress will be used to determine isolation.

Option	Value
das.isolationaddress	Set the address to ping if a host is isolated from the network. This is useful if you have more than one service console port that is often used for iSCSI Storage. You can alternatively add a number to the end of the option to specify multiple isolation response addresses, 1–10 are the numbers allowed. If all these addresses are unreachable, the host would be considered isolated.
das.vmMemoryMinMB	The default setting of the reservation size (256MB by default) to use in admission control calculations. Higher values will reserve more space for failovers.
das.vmCPUMinMhz	The default setting of the reservation size (256MHz by default) to use in admission control calculations. Higher values will reserve more space for failovers.
das.defaultfailoverhost	Use as the first host to try when failing over a VM to another host.
das.allowVmotionNetworks	Set to True to allow the NICs for VMotion networks to be considered for VMware HA usage.
das.allowNetworkX	Enable the use of portgroup names to be used to control the networks used for VMware HA. Note, X is a number starting at 0.
das.isolationShutdownTimeout	Time, defaults to 300 seconds, to wait before VM is forcibly powered off if there is no clean shutdown due to isolation response setting.
das.bypassNetCompatCheck	False by default, set to true to use the enhanced network compliance check that increases cluster reliability.
das.failureInterval	If there is no heartbeat for XX seconds (30 seconds by default) declare VM dead.
das.minuptime	Minimum amount of time a VM has to be up for HA to be considered for this VM. Default is 120 seconds.
das.maxFailures	Maximum number of HA failures and automated resets within das.maxFailureWindow time frame; the default is 3.
das.maxFailureWindow	The default is one day set in seconds, 86400. If set to –1, the das.maxFailures is an absolute number of resets allowed.

For these options to take effect, VMware HA must be disabled and then reenabled on the cluster. You should verify that the AAM daemons are also not running on the VMware ESX host in question, as well. If one host is stuck, for some reason, the change will not take place. To make this verification, run the following command from the console of participating VMware ESX hosts.

```
ps ax ¦ grep ft ¦ grep -v grep
```

Any output implies that AAM (represented by several processes beginning the characters `ft`) did not stop, and you will have to investigate the reason why. For example, the processes could be in a defunct state, which is impossible to kill and requires the system to be rebooted. That they are in a defunct state is very important to note. This could be due to down level device drivers within the service console of a VMware ESX host or something else entirely. After you discover that they are defunct, your investigation begins. In general, you use the `pstree` command to get a process tree to determine the exact owner of the default process and then correlate this to the various VMware HA log files within the directory `/var/log/vmware/aam`.

It is possible that these values can also be modified to increase the sensitivity of your cluster. Given that, then it is possible that failures will be more prevalent with respect to VMware HA.

Resource Contention

Many people misunderstand the reason behind VMware DRS. VMware DRS is not a load-balancing service, yet it may appear that way. It is, instead, a service that will alleviate resource contention on a node. It looks at memory and CPU utilization and determines if there is any resource contention on the node. Then it uses that information to either recommend or automatically move the VM to a node that has no memory or CPU contention.

However, it is possible that a VM could land on a node and be forced to vacate once more, if the automated migration threshold is set too aggressively. This could mean that in rare cases VMs constantly shuffle around the network as they experience contention in CPU and memory resources on each node. This could be an esoteric form of attack by forcing a VM to use all memory and CPU assigned to it when the VM was created. Aggressive automation and movement of the same VM could act like a DoS attack, because the VM spends more time moving than processing. Granted during VMotion, processing continues except when the vCPU state and in use memory is copied.

In this case, monitoring of system accounting information could be used to detect this type of attack and behavior. It is also a case against overly aggressive automation of the movement of VMs from node to node. After the nodes are balanced, still by hand and by judicious use of DRS, it is possible then to place DRS in a partial automation mode that will ask or recommend whether your VMs should move from node to node.

As stated at the beginning of this section of the book, monitoring performance and accounting information is a good start for detecting possible security issues.

The process accounting logs to monitor depend on the process accounting software installed as part of your hardening steps. This is covered in Chapter 11, "Security and Virtual Infrastructure."

Isolation

An oft overlooked concept is what happens when nodes of your cluster become isolated through software means, the experimental VMware DPM, or a crash, hopefully not caused by a security issue. Do you have the VMs set up to ensure that there is no isolation of networks, or if enough nodes fail do you now have data commingling on the virtual network and storage fabrics? Such isolation behavior is not due only to clustering issues, but could also be due to disaster recovery response. Consider Figure 5.7, which contains DMZ, production, and dev/test clustered virtualization servers. Not shown on the diagram is the shared

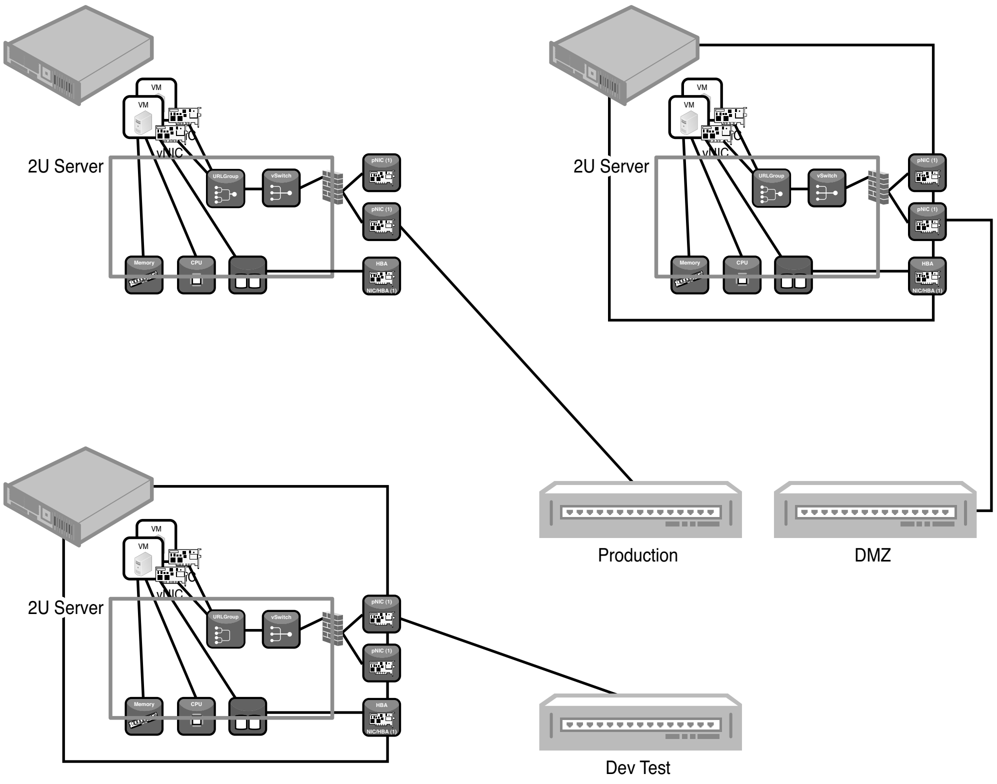


Figure 5.7 Sample data center

storage used by each of these virtualization hosts. Each host is used for a specific reason, which could be because of who bought the resource, but they are set up to employ VMware HA. Note that this example is **not** a recommended solution. In the enterprise, this virtual network would not exist; it violates every best practice. I have presented it to demonstrate what could happen if isolation response is not considered at design time.

Now in a failure mode, the datacenter is represented by Figure 5.8, where the DMZ VMs have been brought up on either the production or dev/test nodes cluster nodes. This is one reason why it is always recommended that you place a DMZ on its own pair of clustered machines. Unfortunately, not everyone can afford to do so. During normal operations, a DMZ VM would never be on these hosts, but a DMZ portgroup would already exist that would use a VLAN to allow communication to the DMZ switch through the network switch fabric. Normally, traffic would not be directed to these hosts because there would be no DMZ VMs hanging off the portgroup. Broadcast traffic, if allowed within a DMZ, would still exist, however.

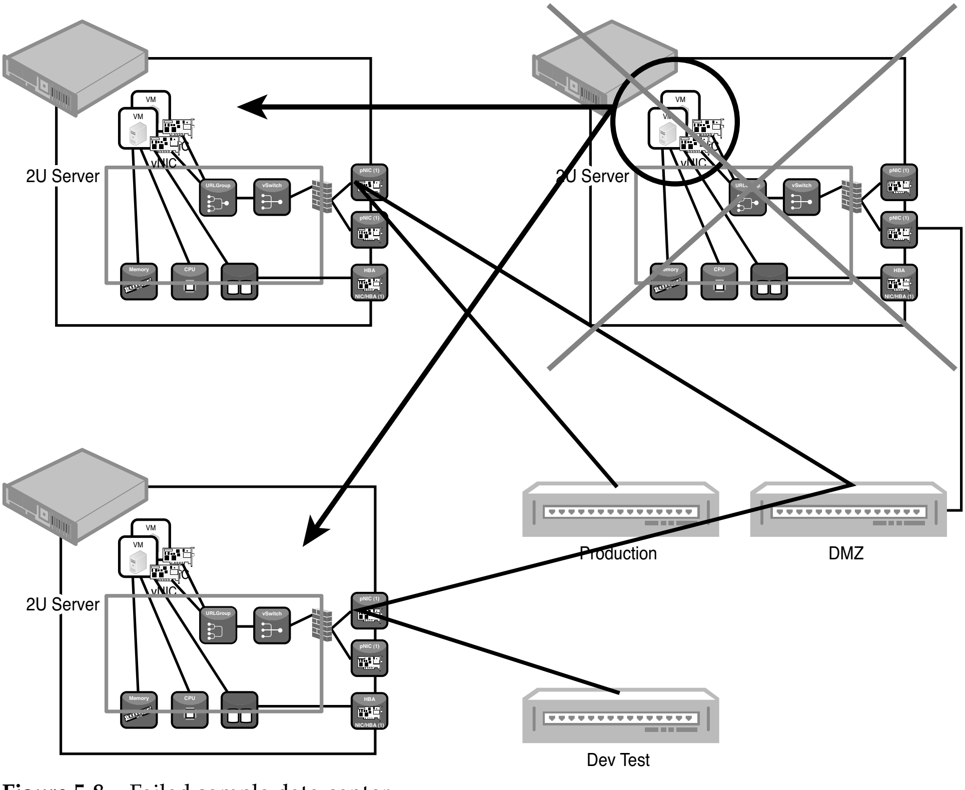


Figure 5.8 Failed sample data center

It is important to note that the isolation response was to combine the DMZ and production networks on the same virtual switch with a minimal number of necessary systems to be in use. However, now the virtual switch is commingling all network data from the DMZ and production networks onto the same cable. Although we will discuss this further in Chapter 9, "Virtual Network Security," VLANs do not necessarily protect you. The 802.1q RFC does not contain anything that will guarantee security. It is often assumed that it does, but it does not. The data within the virtual network is protected from all VLAN attacks listed in Chapter 2, but if the attacker targets a physical switch for VLAN attacks, there may not be any protection. Figure 5.9 displays what the packets could look like when 802.1q VLAN tagging is in use, which leads to data commingling on the wire.

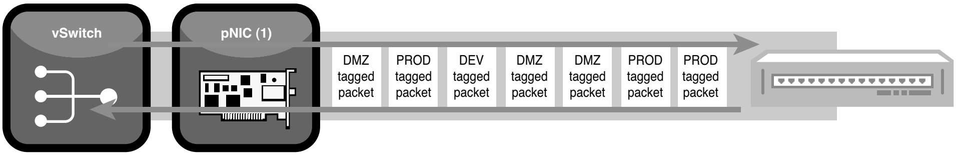


Figure 5.9 Packet depiction with 802.1q

Data Commingling

Data commingling ends up being a networking trust issue to many people; they trust that 802.1q and the networking hardware involved will protect their data from leaking between their defined security zones, even though the data is traveling over the same wire, ports, and other hardware as it travels from virtual machine to server and back again. There are many Layer 2 VLAN attacks available and more being researched. These attacks will continue to be developed. VLAN attacks and some network switch failures allow data to leak across the VLAN boundaries and possibly stolen or false data to be injected into the data stream.

Whether you can legitimately commingle data depends entirely on the data to be commingled. In some cases the privacy laws of the country in which the data is warehoused, as well as how you interpret the privacy standards to which you must be compliant, come into play for some types of personal data. Consider the value of the data when assessing the risk from a breach of security regarding data commingling. How valuable would this be to a hacker? How much would it cost the organization if the data was stolen?

In general, however, data commingling between security zones should be avoided because of the possibility of data leaking across the VLAN boundaries. Some government organizations will explicitly forbid network data commingling.

Data commingling is possible because as nodes fail, VMs are brought up on new systems that are looking for network labels and not specific devices. So if node A has four network labels all using four separate vSwitches, but on node B six network labels are on four vSwitches, data could be commingling. See Figure 5.10 for a pictorial view of this possibility.

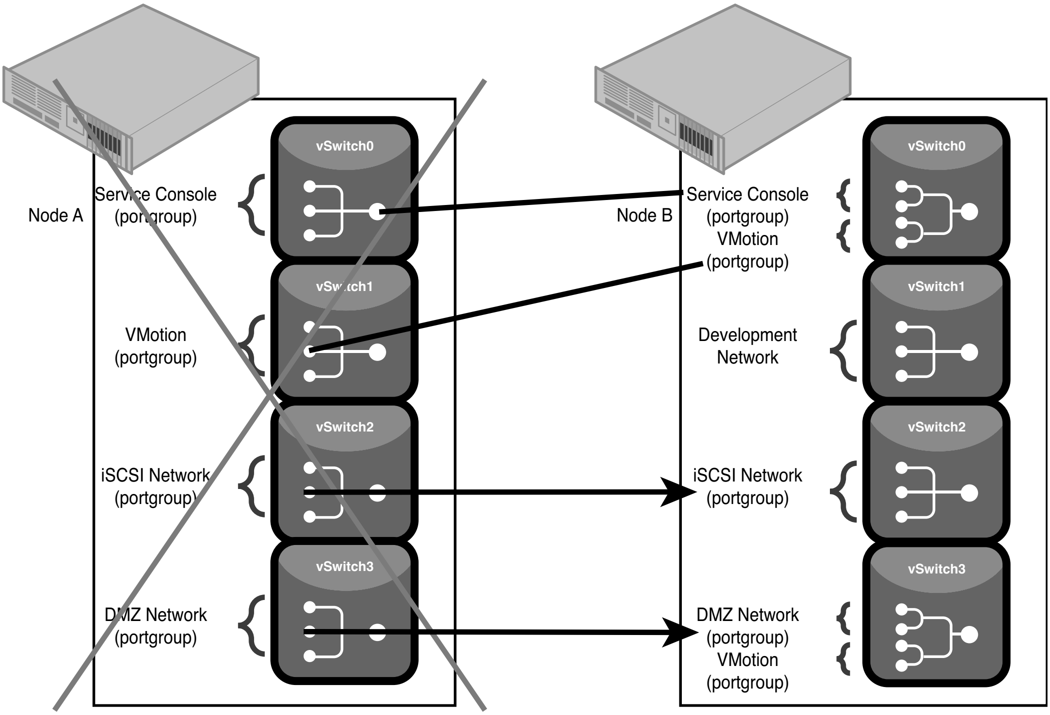


Figure 5.10 Network Labels used for failover

When using a Distributed Virtual Switch and Host Profiles with VMware vSphere 4, the chance of missing, misspelled, or different network labels between different nodes in a cluster will decrease significantly. However, with vSphere 4 it will still be possible to add special purpose, per node virtual switches.

In Figure 5.10 we notice that data commingling exists on vSwitch0 as well as vSwitch3 because of multiple networks being compressed from single vSwitches to portgroups on these vSwitches. If each vSwitch had multiple physical NICs (pNIC) associated with it, and each was assigned to a separate portgroup, data commingling would be mitigated except in the case when one of those pNICs failed. Then we would be back to data commingling on the wire. This will be discussed extensively in Chapter 9.

For DMZ VMs, or any VM that has a different classification than other VMs on the same node, it is recommended that these VMs be on their own cluster of virtualization hosts. This will minimize data commingling within the virtualization server. Note that your external physical network could still contain such instances, unless there is isolation at this level as well. To ensure such data commingling does not exist; you should at the very least follow the steps in the following Security Best Practice. Note that you can substitute "classification level" for "DMZ" in the practice.

Security Best Practice

If there is a virtualized DMZ environment, place all your DMZ VMs on nodes with their own LUNs.

Present these LUNs to DMZ-specific VMware ESX or ESXi hosts.

Do not commingle DMZ VMs with non-DMZ VMs on the same node or LUN.

This implies that during by-hand or automated VMotion, the VMs will always stay on the nodes assigned to the DMZ.

Do not use SVMotion to move VMs from DMZ-specific LUNs to non-DMZ specific LUNs.

In addition to the network implications of an isolation response, there are also storage considerations. In a normal isolation case of a host failure, this is most likely not going to happen because the VMs will not move between datastores automatically, but commingling storage could be the result in the case of disaster recovery. Often, rules exist that govern the handling of network and storage data of different classifications. However, the solution is often to spend more money to get more hardware to alleviate the possibility of data commingling.

Outside of data commingling, it is important to realize that VMware HA is controlled by EMC, formerly Legato AAM. AAM is very sensitive on VMware ESX and ESXi hosts and could lead to inadvertent failures in the HA service. This could cause failover not to occur. The most common failures of VMware HA deal mostly with its configuration. However HA can be adversely affected by domain name servers used on the network as well. VMware ESX servers need to be added to VMware Virtual Center by a fully qualified domain name (FQDN); otherwise, AAM can have issues in proper detection of other nodes within the cluster. Each FQDN of the hosts within the cluster needs to be resolvable by VC and all virtualization hosts through either DNS or a hosts file (preferably DNS).

One solution to FQDN resolution is to add a local hosts table on each ESX server, which contains the FQDN and short name of the other hosts within the cluster, the license server, as well as the virtual center server. This is also a solution to the issue of DNS servers being lost for some reason because VMware HA depends on solid name services being run.

Why the license server as well? VMware HA will not restart VMs unless the license server is also available.

Isolation response also deals with the order in which VMs are restarted, as well as the grouping of VMs on various hosts. These are per VM isolation responses associated with VMware HA and DRS. An example of this failure could be booting the database client VM before the database has completed its boot. Another example would be having VMs on private networks grouped across different hosts where no interconnect exists, but the network label exists. Figure 5.11 depicts these failures.

In Figure 5.11, neither of the rebooted VMs can talk to each other because the Private Network has no physical NIC attached to it, so there is no interconnect for that vSwitch between the new nodes. Because VMware HA boots VMs and places them on networks using portgroup labels, the isolation response will cause quite a bit of havoc until fixed. Systems may not be able to communicate with one another.

The last concern for isolation response is whether the target host(s) has enough resources available to start the VMs in question. If it does not, VMware HA will fail to boot the VMs. In some cases this will happen even if you told the cluster to avoid availability constraints. This is due to HA reserving some resources just for failover. In rare cases, when all nodes but one fail, there may not be enough resources. You may be able to force a boot of these VMs by setting the following values to 0 within the VMware HA advanced settings. The defaults are 256MHz or MBs, respectively. These values are used to reserve resources just for VMware HA, and if set, represent the minimum amount of resources required for any virtual machine within a cluster. If this minimum amount of resources is not available, the VM will not be able to boot and participate in the cluster because there will be insufficient resources.

- das.vmCpuMinMhz 0
- das.vmMemoryMinMB 0

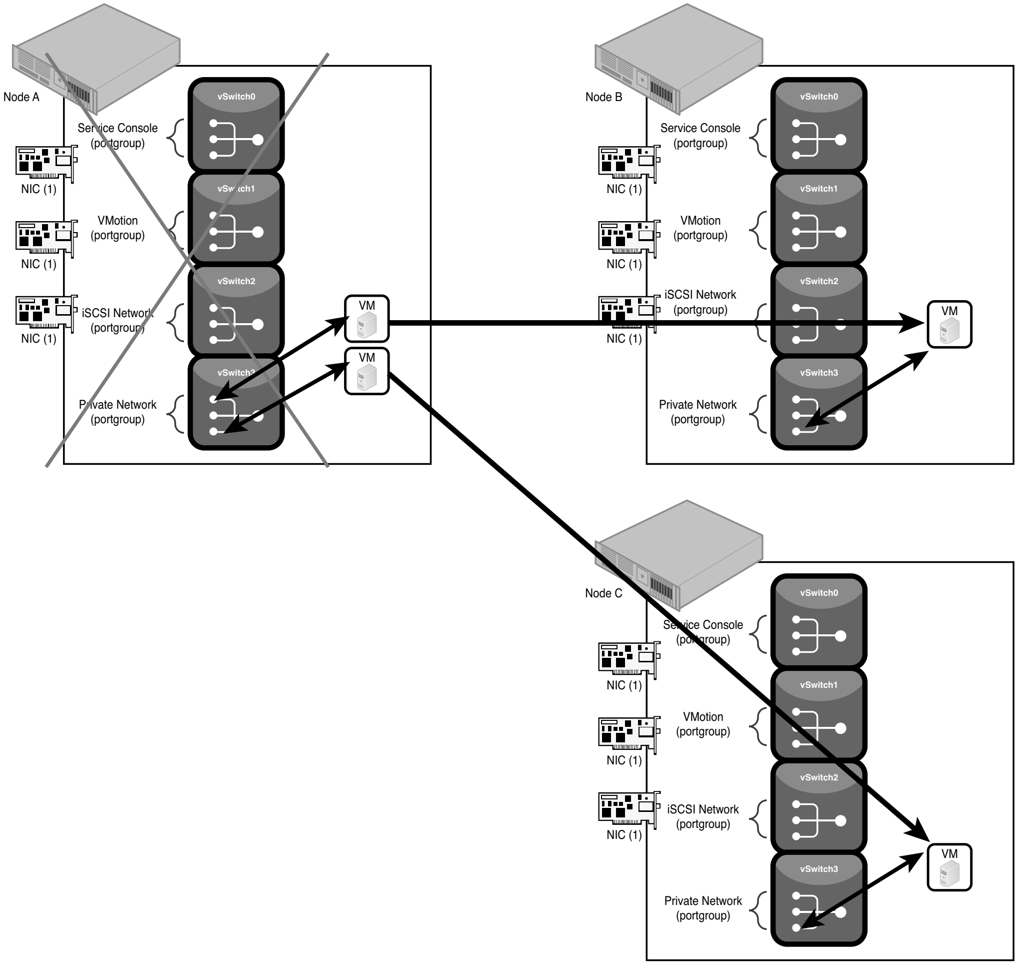


Figure 5.11 Boot order/location failures

However, if you do overload the host(s) within your cluster because of failure of other nodes, other problems could occur that have nothing to do with clustering. Overcommitting the four primary resources will cause performance problems that could be severe and cause applications to fail, networks to be unavailable, and even storage subsystems to be overloaded.

Given these possibilities, it is important to maintain and audit your VMware HA and DRS configurations to determine if they are valid, so that the proper isolation response occurs. All the VMware HA settings are very easy to modify, either accidentally or purposefully.

VMware Cluster Protocols

All VMware Cluster protocols are unencrypted. The protocols travel over the network as clear text and are unprotected with no encryption or authentication of the source of the packets that initiate the clustering events. This implies that if the proper packets are sent to a server along the management network or direct to the service console VMware HA could fire and failover VMs, VMware FT could failover VMs, VMware DPM could power off nodes and failover VMs, or VMware DRS could move VMs from node to node.

This could be used to create an attack against your cluster resources whose impact could range from merely annoying to possibly destructive. We cover how to protect these networks in Chapter 9, but you should understand why this is necessary. All communication from management tools flows over the management virtual network on each of the VMware ESX/ESXi nodes. In some cases, the data does as well; those cases would include VMware Cluster heartbeat and responses to the management commands. However, not all protocols used by VMware ESX and ESXi are encrypted, such as VMotion and Storage VMotion protocols. The most important data from a hacker's perspective is the memory image of the VM (provided by VMotion) or the disk image of the VM (provided by SVMotion). These two technologies make use of the VMotion network, which is suggested to be separate from any other network. This data is 100% clear text and provides the most valuable, and therefore the most dangerous, data to leave unprotected.

Given that the protocols used, except for authentication, are clear text, the entire virtual environment could be at risk when the cluster protocols are in use. Specifically there is now a best practice that management communication and VMotion communication be isolated from normal networks. Administrators, specifically, have unparalleled access to the virtual environment, but they may not be the VM owner or even have rights to log in to the VM in question. Therefore they do not need access to the VMotion and SVMotion data. Therefore, it may also be important to isolate VMotion and SVMotion traffic from the administrators, hence the management network, as well as the other networks involved.

VMware Hot Migration Failures

Even with all the built-in protections within VMware that prevent the VMotion or SVMotion of VMs that do not meet our previously discussed criteria (CD-ROMs connected, incompatible CPUs, and so on), it is still possible that the migration will fail without notice. Intel FlexMigration and AMD Enhanced Migration are supposed to alleviate migration failure, but it can still happen. If the migration fails, the VM and hosts should be examined for the root cause of the failure. One tool that will help in this analysis is Tripwire Opscheck (www.vwire.com/free-tools/opscheck).

Migration failures can be caused by overloaded storage fabric, SCSI reservation conflicts, temporary loss of connectivity to the storage device, or complete storage failure. Migration failures involving storage often lead VMs to crash without warning. However, what caused these issues in the first place with regard to storage? Could it have been an attack?

One issue that does come up from time to time is that Microsoft Windows VMs appear to be more resilient to VMotion between disparate CPUs than Linux VMs. For example, I migrated a Windows VM from an Intel Quad Core CPU to an Intel Single Core CPU that was several generations back by masking off the list of registers in use by the quad core so that the VM could safely be migrated. However using the same masks to the VM made to a Linux VM, the migration caused the VM to crash.

Setting the mask bits for each VM will not always guarantee a successful migration; therefore, it is very important to test these changes on nonproduction VMs similar to production hardware. Actually, it should be identical to production hardware because even the slightest change to the feature set used on the systems can affect VMotion mask bits.

Many very easy-to-change aspects of virtual environment management could have undesired effects, and setting CPU masks is one of these items. If the mask is set incorrectly, VMware ESX or ESXi will not allow the migration, which could force the VMware DRS to fail for the VM, or worse, the migration is allowed but the VM crashes. It is important to know what will happen in these cases.

You may be asking how this is security related; it could be that the mask was changed accidentally or even purposefully so that the failure does not occur until you absolutely need to use VMotion or VMware DRS. The questions would be, how did this information change and from what source did the VMotion request originate? Because these settings will affect the uptime of a VM, they could be security related. This is where a baseline of all VM configurations come in very handy as a comparison base to determine what actually has changed. Next, you would correlate these changes to the log files for VC and the virtualization host to determine who did what when.

One thing to note is that there is no way to set CPU masks on 64-bit VMs, so you must depend on EVC to protect you from such failures, as well as the BIOS settings of the VMware ESX/ESXi target host. The 64-bit VMs require that the Intel VT or AMD-V options within the processor be enabled within the BIOS. Most servers these days do not have this set by default. To enable EVC you may also need to set other BIOS bits.

Virtual Machine Clusters

We have covered hardware clustering (RAID Blade) and VMware cluster capabilities, but another business continuity tool is clustering VMs. VMs can be clustered using any shared disk clustering technology supported by the guest operating systems within the VM and also supported by VMware ESX and ESXi. A discussion of how to set this up is outside the scope of the book because it is well documented within a VMware white paper on the subject (www.vmware.com/pdf/vi3_35/esx_3/vi3_35_25_u1_mscs.pdf). You should realize that where you set this up could affect your cluster in a major way.

Cluster in a Box (CiB) clusters VMs within the same VMware ESX or ESXi host. This may provide a small level of redundancy for the VM, but if the host fails for some reason, the cluster also fails. This is the risk when using CiB. CiB also supports only up to two shared disk cluster node VMs.

Clusters between VMware ESX or ESXi hosts can also be achieved and add more redundancy to the VMs, because a single node failure can now be absorbed by the VMs themselves. VMware supports up to eight shared disk cluster node VMs in this guest clustering form. This may not be desirable as well, because the isolation response for each VM on the multiple hosts will need to be managed. Nor can these VMs be migrated using VMotion, SVMotion, or via VMware DRS.

The last cluster is when a VM participates in a cluster with physical nodes. This has the added advantage of the VM being a relatively inexpensive option to use when failure occurs on the physical nodes. The same limitations, with respect to VMotion, exist in this VM clustering mode as well.

All these clustering modes have security issues, but they are mostly within the guest OS and not the underlying virtualization layer. The major concerns are placement of the VMs within the VMware Cluster and how isolation responses are set up so that HA and DRS (VMotion) respond appropriately and do not adversely affect the guest cluster.

Fault Tolerance

VMware vSphere 4 introduces the concept of VM Fault Tolerance, where a shadow copy of a VM is kept in CPU lockstep with the original VM. This could replace the need for Virtual Machine Clustering except in cases where load balancing is desired instead of high availability. However, because the VM and the shadow VM are kept in vCPU lockstep, a security issue in the original VM still exists in the shadow copy. Therefore, a forced failure of the original VM could also cause a forced failure of the shadow copy, and Fault Tolerance would kick in and try to switch over to an infected or inoperable shadow copy. Fault Tolerance will not protect against security issues because a shadow copy is running the same VM. If, for example, a rootkit is installed into the original VM, the shadow copy also has that rootkit installed.

However, if you use traditional clustering, you are running multiple independent virtual machines, so the infection or security breach in one VM does not necessarily imply a breach to other nodes in the cluster.

Management

It is not possible to discuss clustering without discussing how clustering is configured and the possible issues with not having the proper management tools in place at use time. It is also important from a security discussion to understand how each of the management tools fits into clustering so that you know from where each action starts. We know they all will eventually end at the VMware ESX or ESXi host. Knowing this also brings to light possible attack points in which changes could be a source of an issue.

To properly configure VMware HA, you need access to a VMware VirtualCenter (VC) Server. Although it is possible to configure this from the command line of every node in question if you know EMC AAM, it is very inconvenient, and a simple mistake could cause VMware HA to not work. It is therefore recommended that you manage this through the virtual infrastructure client accessing VC. Auditing for changes in VMware HA configuration files is an important step. These files should be part of your baseline and live within

/etc/opt/vmware/aam. VC is also the place where you define the isolation response for each VM within the cluster should they be migrated, powered off, stick with other VMs, and so on.

Another important aspect of VMware HA is that access to the VMware License Manager is required when VMs are rebooted by VMware HA or when VMware HA is configured. So you could have a chicken and the egg situation if the VM containing your License Manager fails; then HA cannot do its job because the VMs may not be able to boot. This is one of the main reasons to have your License Manager on a clustered setup using either physical or VM clustering across nodes.

VMware DRS, and therefore VMotion, require that VC also be available. It is impossible to launch VMotion without VC assistance. Even third-party tools, such as HP Systems Insight Manager (HPSIM), which claim to do VMotion within their interfaces require VC to be available to initiate VMotion. This is an important item to realize; if VC is no longer available for some reason, there is no way to use VMotion to move the VMs to other hosts.

VMware SVMotion in VI3 requires VC as well as the Remote Command Line Interface (RCLI). Without the RCLI, SVMotion cannot occur (as of VMware ESX version 3.5.0 Update 4). This toolkit of command-line tools adds another management method into the mix that can configure nearly everything that the VIC can be used to configure. RCLI was introduced with ESXi because no easily accessible console exists for the VMware ESXi. However, it also works as a way to access and manage VMware ESX.

One other important aspect of managing a VMware Cluster is the deployment of the VMs. It is suggested that there be a deployment node on which you deploy all your VMs; after testing and installing, move them onto the production environment before powering them on. We discuss this more in Chapter 6, "Deployment and Management." The main gain of using a deployment node is that SCSI reservation conflicts for initial deployments are limited in scope.

To configure enhanced vmotion capability (EVC), you must also use VC. You create an EVC-enabled cluster using vCenter and add your nodes into the cluster one by one. EVC also only does away with the setting of CPU masks when using like families of processors, such as Intel or AMD. However, to use VMotion between Intel and AMD systems, CPU masks are still required.

The experimental Dynamic Power Management (DPM) tool also required VC, but furthermore requires the capability to wake the server on LAN using the first pNIC associated with the management appliance of VMware ESXi or the service console of VMware ESX, which is yet another BIOS setting. If Wake-on-LAN is not enabled, DPM cannot power on the VM. However, this also leads to the question that any network traffic sent to the Wake-on-LAN port could also force a boot of the host, even without VMware DPM firing: yet another reason to protect the management network. VMware DPM works with VMware DRS. As resource contentions are recognized during normal operating times, VMware DPM powers on the VMware ESX or ESXi hosts, giving more choices for VM placement by VMware Dynamic Resource Scheduling (DRS).

Overactive VMware DRS with DPM enabled could lead to higher power costs. This is a direct to the bottom line possibility and perhaps the goal of a hacker.

Although we did not delve into this section like we have with the others, it will give you food for thought as we discuss further management issues. However, you should now know the basics of the data flow around the management network for each of the types of cluster functionality. Even so we will delve into this data flow more in Chapter 6.

Conclusion

VMware clustering employs five distinct technologies, each of which will affect security in different ways. This chapter looked at security from the perspective of VMware High Availability (HA), VMware Dynamic Resource Scheduling (DRS), VMware Fault Tolerance, VMware VMotion, and VMware Storage VMotion. Building on Chapter 4, where we specified that storage networks should be isolated, we now state that VMotion/SVMotion, console, and management networks should also be isolated from all other networks. Chapter 9 will go into this in detail now that we have the reasons for this recommendation.

Although most of the items discussed in this chapter are mitigated using proper network configuration and security, it is important to consider the listed possibilities when discussing security of the virtual environment and performing risk analysis.

We posed a question in the second section of this chapter concerning why VMware HA did not fail the VMs over from the failed machine. Have you thought of any reasons that could have caused this, given the preceding text? In this case, it was a bug in VMware HA for VMware Virtual Infrastructure 3.5.0 Update 2 that caused VMware HA to fail to configure properly. This was fixed in VMware VirtualCenter 2.5.0 update 3 and the 10/3/08 patches for VMware ESX Update 2, but the same thing could happen if someone accidentally or purposefully modified the VMware HA configuration. The problem was that VMware HA reserved too many resources for clustering—so many that VMs could not boot because HA reserved all the resources when only two nodes existed. You could not lower the amount of resources required until the patch was made available.

In the next chapter we will further delve into deployment and management of the virtual environment. We started a simple discussion here, but we shall advance this to cover all the different management and deployment methods available.