GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials (Security 401)"
at http://www.giac.org/registration/gsec

Building a Security Lab with Virtual Machines

By: Baker Hart
(GSEC V1.4 Option 1)

© SANS Institute 2000 - 2002, Author retains full rights.

1 Summary

This paper will introduce Virtual Machines to new security professionals and show how they can be used as a powerful yet simple security tool. Before the advent of Virtual Machines a security lab required the configuration of multiple physical machines to replicate a network environment for testing. This complex and difficult to manage environment was a significant barrier for new security professionals in developing their skills. However, Virtual Machine technology enables multiple simultaneous operating systems to run on a single Intel Pentium-Class machine. Using this technology, security professionals can understand how operating systems are affected by different (malicious) environments without physically building the environment. Using a workstation and Virtual Machine software an entire test lab can be configured virtually on a single machine.

Although there are many Virtual Machine solutions specifically for the mainframe market, there are only a few software products for the Intel platform. VMware Workstation 3.1, one of the most popular, creates Virtual Machines which allow the security professional to easily test many security issues that would otherwise require several computers, and thus, a much more complex environment. After gaining an understanding of the networking features of VMware Workstation 3.1 and completing the included security exercises, new security professionals are able to replicate and study most networking environments using only one machine.

2 Virtual Machines

IBM originally developed Virtual Machine technology in the 1960's for their mainframe computers. The ability to have Virtual Machines was built into the mainframe platform, unlike the current Intel platform that does not support Virtual Machines natively.

A Virtual Machine is basically a computer inside a computer. This technology allows an operating system and programs to operate separately from the original operating system. As far as the operating system running in the Virtual Machine is concerned, it is running on its own computer.

The two most common methods of creating Virtual Machines on the Intel platform are software emulators and virtual machine monitors.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

A software emulator creates a Virtual Machine totally in software. An emulator intercepts and interprets every instruction for the target Virtual Machine. This method uses a large amount of system overhead and can cause Virtual Machines to run very slowly. Further, Virtual Machine emulation can become unstable if the emulated CPU and hardware do not behave exactly as expected.[1]

A virtual machine monitor creates a Virtual Machine by passing most instructions directly to the host machine's hardware. The virtual machine monitor emulates only certain privileged CPU instructions and peripheral hardware providing a significant increase in instruction execution speed compared to total emulation. In this case the stability of the Virtual Machine is also improved because almost all instructions are performed by the real CPU and hardware reducing the chance of emulation errors.

Virtual Machine technology has increased in popularity over the last few years because production environments have had success using Virtual Machines to reduce many types of operational costs (e.g., equipment, configuration and management time).[6] If this trend continues, knowledge of Virtual Machine technology strengths and vulnerabilities will be required of the well-rounded security professional.

3 VMware Workstation 3.1

SANS has set the precedence of teaching security professionals how to accomplish most basic security tasks using open source or free software. However, it is also important to be practical about the hardware and time resources needed to learn basic security skills. For these reasons VMware Workstation 3.1 (VMware) was chosen to demonstrate an alternative to purchasing several test machines and investing the time to configure and manage the operating systems on each machine. The cost savings from reducing hardware and time requirements can be used to justify the cost of VMware.

VMware uses a virtual machine monitor to create Virtual Machines. The efficiency of a virtual machine monitor enables VMware to provide other important features such as:

- Up to eight Virtual Machines at one time in one computer.
- A networking environment that allows almost any network configuration to be replicated.
- Hardware mapping from virtual hardware directly to real hardware enabling device disconnection and reconnection.
- Virtual hard drives up to 128 GB in size.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

- Seamless integration between the host desktop and a Virtual Machine desktop enabling easy interaction with active Virtual Machines.

A fully functional Linux or Windows version of VMware Workstation 3.1 can be obtained in four forms:

- A 30 day free evaluation copy (electronic download only).
- An academic license for $129.95.
- A full license from electronic download for $299.00.
- A full license from packaged distribution for $329.00.

All of these options are available from the VMware site at: http://www.vmware.com/.

There are other versions of VMware dedicated to the production server environment that are not covered in this paper.

VMware is unique in how it implements Virtual Machine technology but not in its ability to create Virtual Machines. A product called "Virtual PC" can also be used to create Virtual Machines and is available at: http://www.connectix.com/. This product is similar to VMware but does not have as many features especially in the networking area.

4 Lab Configuration and VMware

VMware is available for Linux and Windows NT/2000/XP platforms. However, to simplify this section it is assumed that the VMware *host system* is running Windows 2000. The Linux and Windows versions are almost identical. To reduce configuration issues, install VMware on the host operating system you are most comfortable with. Be sure the version purchased is the one you need, inasmuch as a Windows license is not transferable to Linux and vice versa.

To make the discussion easer:

- **Host System:** This is the system running VMware.
- **Guest Operating System:** An operating system running inside a VMware Virtual Machine.
- **OS Template:** A base Guest Operating System installation.
- **Lab Template:** An OS Template with all necessary tools and applications installed to perform a given task. (The reason for differentiating between images and templates is that during the process of using images in the labs the images may be altered or damaged especially if a malicious exploit is being tested. Using

© SANS Institute 2000 - 2002, Author retains full rights.

new images will also enable you to get consistent lab results from a given exercise.)

- **Lab Image:** A working copy of a Lab Template.

VMware Virtual Machines can support almost all Microsoft operating systems from DOS to the new ".NET" servers and most Linux and BSD distributions. However, to simplify this section on creating *Virtual Machines,* it is assumed that we will focus on Windows 2000 Server (We will use the server version because it can used in both server and workstation roles.) and Red Hat Linux 7.2. The issues faced in constructing a Virtual Machine security lab for these operating systems will translate easily to many other operating systems. You are strongly encouraged to setup templates and experiment with all operating systems that exist in your environment or that you are curious about.

Note that all guest operating systems installed in Virtual Machines must be licensed. Trial versions of almost all Microsoft products are available by downloading or ordering media online. Linux distributions and applications are available from many sources at little or no cost.

One useful feature of a Linux host install of VMware versus a Windows install is that you can use a utility called *vmware-mount.pl* (executable) to mount VMware virtual drives not currently in use. This feature is not available on the Windows platform.

4.1 Host System Hardware Guidelines

VMware is a resource intensive application. Use the following guidelines only as a relative guide to the resources needed by VMware when running multiple Virtual Machines on a single host system. These guidelines are for a Windows 2000 host system and Windows 2000 Virtual Machines. If Linux is used for the host operating system and most of the guest operating systems then the hardware resources needed on the host system can be reduced depending on configuration of the guest operating systems.

The maximum memory available to Virtual Machines in VMware is limited to 1GB. System memory in excess of 1GB can only be used by the host system. Virtual Machine performance will degrade quickly if at least 128MB of memory is not reserved for the host system. The amount of host system memory needed will further increase as the number of Virtual Machines increase.

The most important single hardware resource needed by Virtual Machines is memory. To ensure reasonable performance, each guest operating system must have dedicated system memory of at least 128MB for Windows 2000 and 32-128MB (depending on the configuration) for Linux installations. Experimenting with memory settings on Virtual Machines is the best way to optimize performance. Most Virtual Machines will also have better performance on a slower CPU host system with plenty of memory than on a faster CPU host system with memory constraints.[1]

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Host system drive space requirements can vary greatly depending on the guest operating system configuration but at least 2GB for each template and image is reasonable.

Here are some general hardware guidelines for host systems:

- **Basic Workstation**

The recommended hardware for running two to three Virtual Machines:
CPU (1) 800+ MHz
RAM 512 MB
HD 7200 RPM
Drive Space 20+ GB
RAM available to Virtual Machines: 384MB

- **Intermediate Workstation**

The recommended hardware for running three to five Virtual Machines:
CPU (1) 1.5+ GHz
RAM 1 GB
HD (2) 7200 RPM RAID 0 stripe
Drive Space 40+ GB
RAM available to Virtual Machines: 768MB

- **Advanced Configuration**

The recommended hardware for running five to eight Virtual Machines:
CPU (2) 1+ GHz or (1) 2.0+ GHz
RAM 1.5 GB
HD (2) 7200 RPM RAID 0 stripe
Drive Space 60+ GB
RAM available to Virtual Machines: 1024MB

4.2 VMware Networking

Prior to connecting Virtual Machines and building a security lab, networking Virtual Machines in VMware is one of the most important concepts to understand.

The networking features of VMware allow the user to do almost anything with Virtual Machines that can be done on a normal network. VMware uses virtual network hubs to create network connections for Virtual Machines. A virtual hub works the same way a standard ethernet hub works but exists inside the VMware software. Network connections in VMware are defined by how a virtual hub is connected to the host system and how the Virtual Machines are connected to the hubs.

© SANS Institute 2000 - 2002, Author retains full rights.

VMware creates each network connections using one of nine virtual hubs. These virtual hubs are labeled as **VMnet0** through **VMnet8** and each can support up to eight Virtual Machine connections.

VMware has four different types of connections between a virtual hub and the host operating system:

- **Bridged networking** –This virtual hub connection enables Virtual Machine network interfaces to function as if they were on the same network as the host machine's physical network interface. The Virtual Machine network interfaces are given virtual MAC addresses and can have their IP addresses assigned or pull them from an outside DHCP server. VMnet0 defaults to a bridged connection on the host system primary network interface and VMnet2-7 can optionally be bridged to a second host system network interface.

- **Host-only networking** – This connection type supports connections between Virtual Machine network interfaces and a special non routing virtual network interface on the host system. **VMnet1** is the only virtual hub with this type of connection and also provides DHCP services.

- **Isolated –** This virtual hub configuration has no network connection to the host system or the outside world. Virtual Machine network interfaces connected to a hub configured in this way can only communicate with other Virtual Machine network interfaces that are also connected to this hub. **(default for VMnet2–7)**

- **NAT networking** –This connection works like the host–only networking except the host system acts like a proxy server enabling the Virtual Machines to connect to the outside world using the IP address of the host system. **VMnet8** is the only virtual hub with this type of connection and also provides DHCP services.

© SANS Institute 2000 - 2002 Author retains full rights.

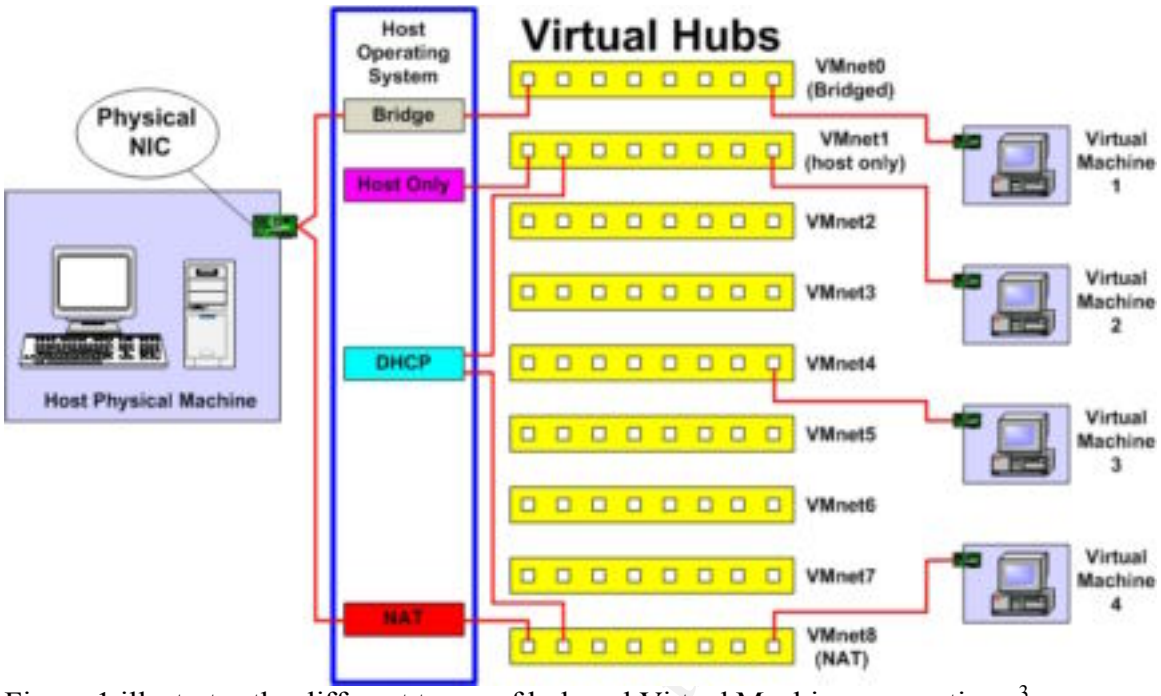


Figure 1 illustrates the different types of hub and Virtual Machine connections.[3]

Figure 1 Elaboration:

- **Virtual Machine 1:** Virtual hub VMnet0 is providing a **bridged connection** to Virtual Machine 1. This configuration enables Virtual Machine 1 to operate as if it had a physical connection to the host systems network. The Virtual Machines network interface could pull an address from a DHCP server operating on the host systems network or it could be configured manually.

- **Virtual Machine 2:** Virtual hub VMnet1 is providing a **host only** network connection to Virtual Machine 2. In this configuration the Virtual Machine can pull a DHCP address from the VMware DHCP server and connect to the host system and other Virtual Machines connected to VMnet1. No outside connections can be made by Virtual Machine 2.

- **Virtual Machine 3:** Virtual hub VMnet4 is providing an **isolated** hub connection to Virtual Machine 3. In this configuration the Virtual Machine can only connect to other Virtual Machines connected to VMnet4. The Virtual Machines network interface must be manually configured to make network connections on this hub. No outside connections can be made by Virtual Machine 3.

- **Virtual Machine 4:** Virtual hub VMnet8 is providing a **NAT** or proxy server type connection to Virtual Machine 4. Using this configuration the Virtual Machine can pull a DHCP address from the VMware DHCP server and make outside network connections using the host systems IP address.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

VMware supports up to three network interfaces per Virtual Machine. Each network interface can be configured independently and connected to a separate virtual hub as shown in the configuration editor screen in Figure 2. Network adapters or other hardware can be added to a Virtual Machine by clicking the "add…" button at the bottom of the screen.

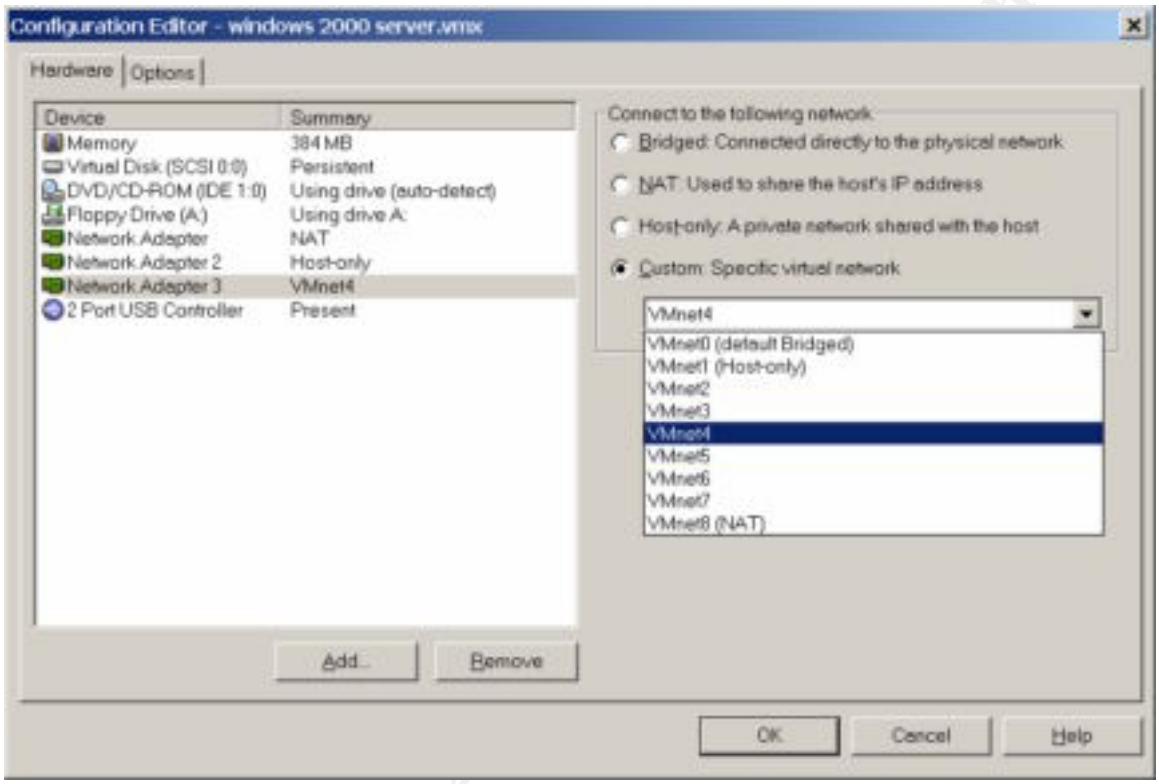


Figure 2 shows the configuration editor for a Windows 2000 Server Virtual Machine with three network interfaces connected to three different virtual hubs.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

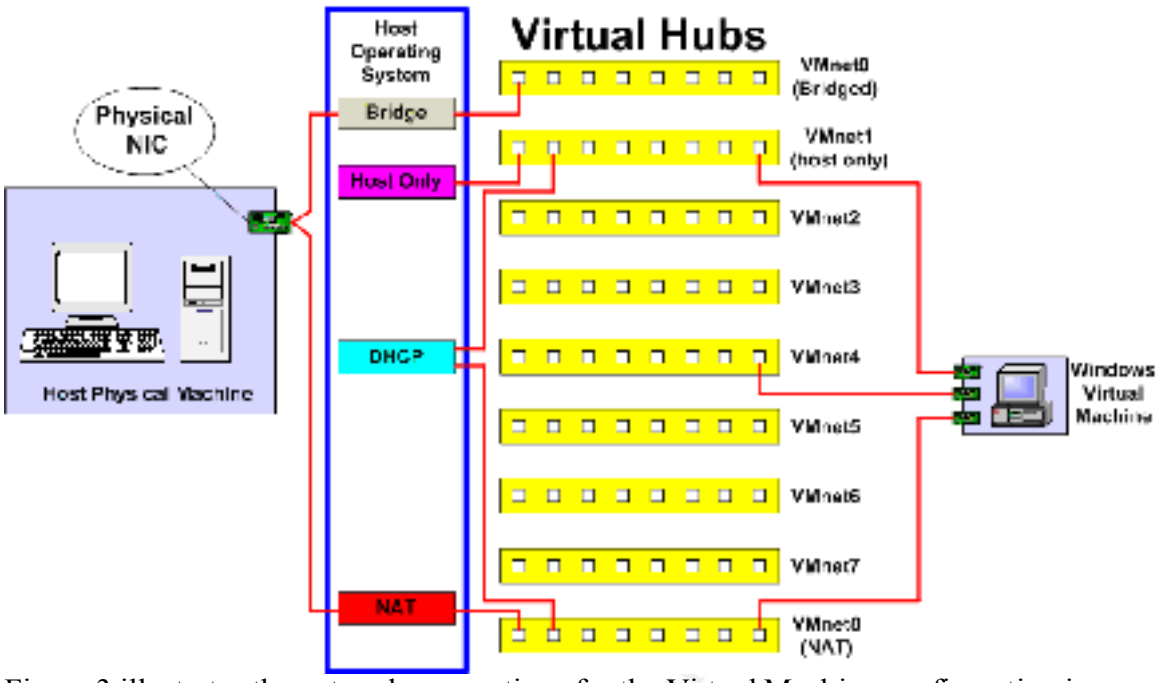


Figure 3 illustrates the network connections for the Virtual Machine configuration in Figure 2.[3]

Using the networking features described in this section almost any network configuration can be replicated in VMware.

4.3 Creating Basic OS Templates

The main purpose of creating an operating system template is to reduce to work involved in building several working Virtual Machine images. The process of installing a base operating system, applying the latest patches, installing the VMware video driver tools, and other basic system prep tasks, is laborious. The advantage of using templates is that after a guest operating system is configured a duplicate can be created by simply copying the Virtual Machine files and editing the Virtual Machine configuration file to point to the new directory.

After VMware has been installed on the host system using the default installation options, the guest operating systems needed for the lab will be installed onto Virtual Machines. These initial guest operating system installs are the operating system templates that will be used to build the remaining lab templates for the exercises.

During a Windows 2000 host system installation of VMware you will be prompted to disable the AutoRun feature on the host system. The AutoRun feature automatically starts programs on CDs as soon as the CD is placed in your computer. Turning this feature off will eliminate several CDROM access issues for Virtual Machines.[1]

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

During the operating system template section for Windows and Linux the steps outlined say to configure a Virtual Machine for NAT networking and configure the guest operating systems network interface for DHCP. This will make connecting the guest operating system to the internet easier provided the host system has internet access. This setting can be changed after the templates are created.

- **Windows Operating System Template**

There are several ways to install Windows 2000 Server in a Virtual Machine, but the easiest way is from a bootable CD. If you have a licensed copy of Windows 2000 Server available, use the bootable CD that came with the license, otherwise order a 120 day evaluation copy from Microsoft for $7.95 at URL: http://www.microsoft.com/windows2000/edk/

To create a Windows 2000 operating system template do the following:

1. Open VMware and select "***New Virtual Machine***" from the ***File*** menu.

2. Select "***Typical installation***" and click "***Next***".

3. Set the guest operating system to ***"Windows 2000 Server"*** and click "***Next***".
4. Name the Virtual Machine "***Windows 2000 Server Template***", Select the location of the Virtual Machine files and click "***Next***".

5. Select "***Use network address translation (NAT)***" and click Finish. This option will be covered in the networking section.

6. Insert the bootable CD into the host systems CD drive.

7. Select the "***Power On***" option from the "***Power"*** menu. At this point the Virtual Machine BIOS will appear and begin booting from the installation CD normally.

    a. If the Windows 2000 CD does not boot you may need to open the ***"Configuration Editor"*** from the "***Settings"*** menu and manually select the drive containing the CD

8. Follow the same Windows 2000 Server installation steps for a normal install. During the network configuration set the network interface to use ***DHCP***.

9. After the Virtual Machine reboots normally select the "***Ctrl-Alt-Del***" option from the "***Power"*** menu to logon.

10. Select the "***Install VMware Tools***" option from the "***Settings***" menu to install the VMware video driver. This is an optional step but the VMware Tools greatly increase the video resolutions the Virtual Machine can support and the

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

mouse will move seamlessly between the Virtual Machine and the host systems desktop.

11. The desktop resolution can now be increased from the default 640 X 480 to a more useable setting.

12. The ***"sysprep.exe"*** utility located on the distribution CD under ***"\support\tools\deploy.cab"*** should be run to strip the unique security identifier (SID) from the template. After this utility is run, the Virtual Machine will shutdown. Do not boot the original copy of your Windows 2000 Server template. Always make a copy and then boot the copy and run through the startup wizard (the wizard that creates a new SID) that way each copy will have a unique SID. This step is optional but recommended because two copies of Windows 2000 with the same SID will have problems communicating.

We now have a Windows 2000 template that can be used to build the lab templates needed for the exercises.

- **Linux Operating System Template**

Just like Windows there are many ways to install Linux in a Virtual Machine but the easiest is from a bootable CD. Most distributions come with bootable CD media including the Red Hat 7.2 distribution we will use for our Linux Virtual Machines.

To create a Red Hat 7.2 operating system template complete the following:

1. Open VMware and select ***"New Virtual Machine"*** from the ***File*** menu.

2. Select ***"Typical installation"*** and click ***"Next"***.

3. Set the guest operating system to ***"Linux"*** and click ***"Next"***.

4. Name the Virtual Machine ***"Linux Template"***, Select the location of the Virtual Machine files and click ***"Next"***.

5. Select ***"Use network address translation (NAT)"*** and click Finish. This option will be covered in the networking section.

6. Insert the bootable CD into the host systems CD drive.

7. Select the ***"Power On"*** option from the ***"Power"*** menu. At this point the Virtual Machine BIOS will appear and begin booting from the installation CD. (See step 7a of the Windows template section if the CD does not boot.)

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

8. The initial Red Hat screen should appear and prompt you to choose the install mode. At the command prompt type ***"text"*** and hit Enter.

From this point select the installer defaults to get a basic system up and running, with the following notes:

- Using the default DHCP setting on the Network Configuration screen will make initial setup easier.

- The Package Group Selection screen is where tools and network services can be selected. Installing the "Software Development" tools will make it easier to install some security tools.

- On the "Video Card Configuration" screen select "Generic SVGA" and "4096" video RAM.

- Don't set the default login to "Graphical" until after you have installed the VMware Tools for the video driver.

After the Virtual Machine reboots use the following steps to install the VMware tools video driver before trying to run X windows.

1. login as root.
2. Select "install VMware Tools" from the Settings menu.
3. mount -t iso9660 /dev/cdrom /mnt
4. cp /mnt/vmware-linux-tools.tar.gz /tmp
5. umount /dev/cdrom
6. Untar the VMware Tools tar file in /tmp, and install it.
7. cd /tmp
8. tar zxf vmware-linux-tools.tar.gz
9. cd vmware-linux-tools
10. ./install.pl
11. Start X and your graphical environment if they are not started yet.
12. In an X terminal, launch the VMware Tools background application.
13. vmware-toolbox &

We now have a Linux template that can be used to create the lab templates needed for the exercises.

4.4 **Lab Templates**

The level of detail provided in this section is intended to provide enough information to get you started. A great reference for more information on installing and configuring the security tools covered in the following sections is the "Security Essentials Toolkit"

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

provided with the SANS Security Essentials Track and listed in the references section of this paper.

Now that the OS templates are configured they can be used to build the lab templates for the exercises. Keep in mind that any Virtual Machine can be compromised just like any other operating system running on any other machine, so be sure that all templates and images are appropriately patched. This is especially true if the Virtual Machines will be used to access or expose any ports to the internet.

If you are concerned about having enough system resources to perform some of the exercises that require several Virtual Machines you may want to consider using your host system to perform some of the scanning or intrusion detection tasks. This will involve installing tools on the host operating system and will require reserving a little more memory for the host operating system but much less than an additional Virtual Machine would need.

- **Scanning Template**

This template is used to scan systems for vulnerabilities, test exploits, or assuming a "hacker" role. Scanning templates can be built in Linux or Windows, however there are more tools for Linux and they are more mature. This also gives someone unfamiliar with Linux an opportunity to gain useful experience.

Example tools to install:

- Nmap
- SuperScan
- Nessus
- Legion
- L0pht Crack (LC3)

- **Workstation Template**

The purpose is to configure a template that functions like workstations in your environment.

Example installations:

- Office productivity suites
- Anti-virus software
- Personal Firewalls

- **Intrusion Detection Template**

This template will have your ID tools installed and depending on how sophisticated you want to get may have two or three network interfaces configured to enable monitoring of multiple virtual hubs or subnets. Again the Linux ID tools are more mature than the Windows versions.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Example Tools to install:

- Tcpdump
- Snort

- **Server Templates**

These are very specific templates because there are so many roles servers can have in an environment. The purpose here is to create templates that can provide all network services needed for testing.

Configuring templates that provide multiple network services will conserve system resources. The trade-off depends on how accurately you need to replicate a given environment versus how fast all the Virtual Machines can run at once. The idea is to reduce the number of Virtual Machines because a single Virtual Machine running two compatible network services will need less system resources than two Virtual Machines running the services separately.

Example network services:

- Windows Domain Controller
- Email
- Data Base
- Virtual Private Network
- Web and SSL
- File and Print

- **Firewall Template**

This template is used to connect two virtual hubs one representing the outer network and the other the secure network containing the protected servers and workstations. The firewall template is one of the more difficult templates to configure because it needs two network interfaces configured for separate IP networks and a firewall application to function correctly.

Example applications:

- Microsoft ISA Server for Windows.
  - A 120 day evaluation copy of ISA server is available at: http://www.microsoft.com/isaserver/evaluation/trial/default.asp
- ipchains for Linux
  - A free copy of ipchains is available from: http://www.netfilter.org/ipchains/

© SANS Institute 2000 - 2002, Author retains full rights.

4.5 Managing Virtual Machine Images

VMware is a powerful tool for creating Virtual Machines but managing the large virtual hard drive files needed for each image poses some issues. After several Virtual Machine templates and images have been created, the amount of hard drive space needed becomes apparent. A Windows 2000 Server Template is almost 1 GB with no tools or applications installed and the size of a working image can be much larger.

Here are some ways to make managing images and templates easer:

- Use large fast hard drives to make storing, copying, and running these large images much easier. If you can afford it, a RAID 0 stripe will also help reduce the copy and backup times.

- Use a file compression utility to help reduce the space needed for backups and make moving the images around much easier. Virtual Machine files compress very well and usually reduce in size by more than half.

- Use a network file share mapped as a local drive on your Virtual Machines. The file share can contain common files and application installation files needed by multiple Virtual Machines. This will help reduce the size of images and make it easer to share files between Virtual Machines. The file share could be provided by the host system or a Virtual Machine dedicated to this purpose.

5 Lab Exercises

This section outlines several exercises using the lab templates created in the previous section to complete several common security tasks. The exercises will start off simple and increase in complexity as progress is made.

The figures provided in this section illustrate one possible network configuration of many for completing the exercises. It is assumed that all Virtual Machine network interfaces that are connected to the same virtual hub are configured to the same IP subnet.

5.1 Basic System Scan (2 Virtual Machines)

This exercise is very straightforward. The objective is to use a scanning image to scan a workstation or server image. Each image must have a network interface connected to the same virtual hub. This exercise may be simple but two Virtual Machines interacting on the same virtual hub defines most of the security experiments you will probably conduct in the process of learning about computer security.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

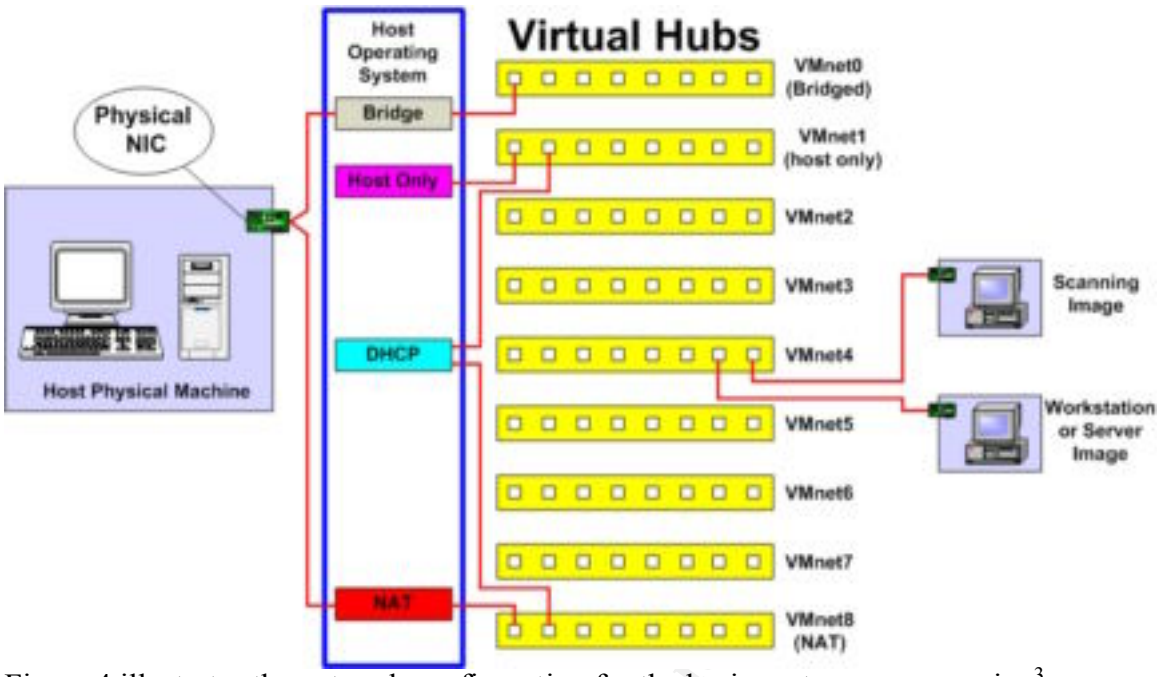


Figure 4 illustrates the network configuration for the basic system scan exercise.[3]

After you have completed your first successful Virtual Machine scan you may want to conduct a scan of all your server templates to provide a baseline for future scans.

5.2 Network Based Intrusion Detection (3 Virtual Machines)

The objective here is to listen to the all the traffic created from a system scan. This lab is a repeat of the previous exercise with the addition of an intrusion detection image used to capture network traffic. All three images must be connected to the same virtual hub and alternately this exercise could be preformed on VMnet1 if DHCP is preferred. Capturing traffic on an isolated hub insures that all traffic was created from the connected images.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

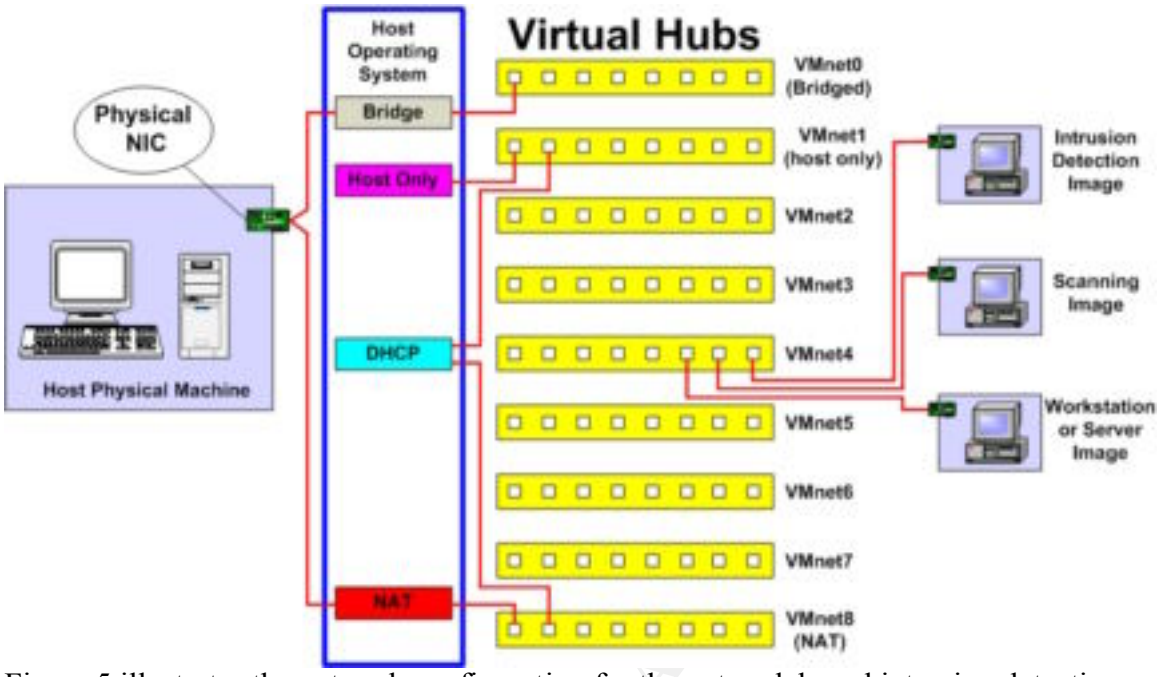


Figure 5 illustrates the network configuration for the network based intrusion detection exercise.[3]

5.3 Capturing Windows Logons (3 Virtual Machines)

The objective of this lab is to use a scanning image with L0pht Crack installed to sniff passwords during Windows client authentication. This exercise requires a Windows domain controller image, Windows workstation image, and a scanning image with the L0pht Crack application installed. The network configuration looks just like Figure 5 with all three images connected to the same virtual hub.

5.4 Connecting Virtual Hubs Using a Firewall Image (3-5 Virtual Machines)

The goal of the exercise is to understand how to use the firewall image to bridge two virtual hubs. This exercise can be very challenging because an understanding of IP routing and your firewall software is needed to configure the images. This exercise is also important because after completion, most networking environments can be replicated.

Set the outer network or unsecured interface on the firewall image toVMnet8 virtual hub and the inside or secure interface to the isolated virtual hub VMnet1. This will make it easer to confirm that the firewall image is configured correctly when the Virtual Machines connected to the secure hub can access the internet. Note that all Virtual

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Machine network interfaces connected to the inside or secure hub must use the firewall images local network interface address as the default gateway. Optionally, you can let all network interfaces on VMnet8 use DHCP if you do not want to configure the addresses manually.

If host system resources are an issue, the intrusion detection and the workstation images are optional.

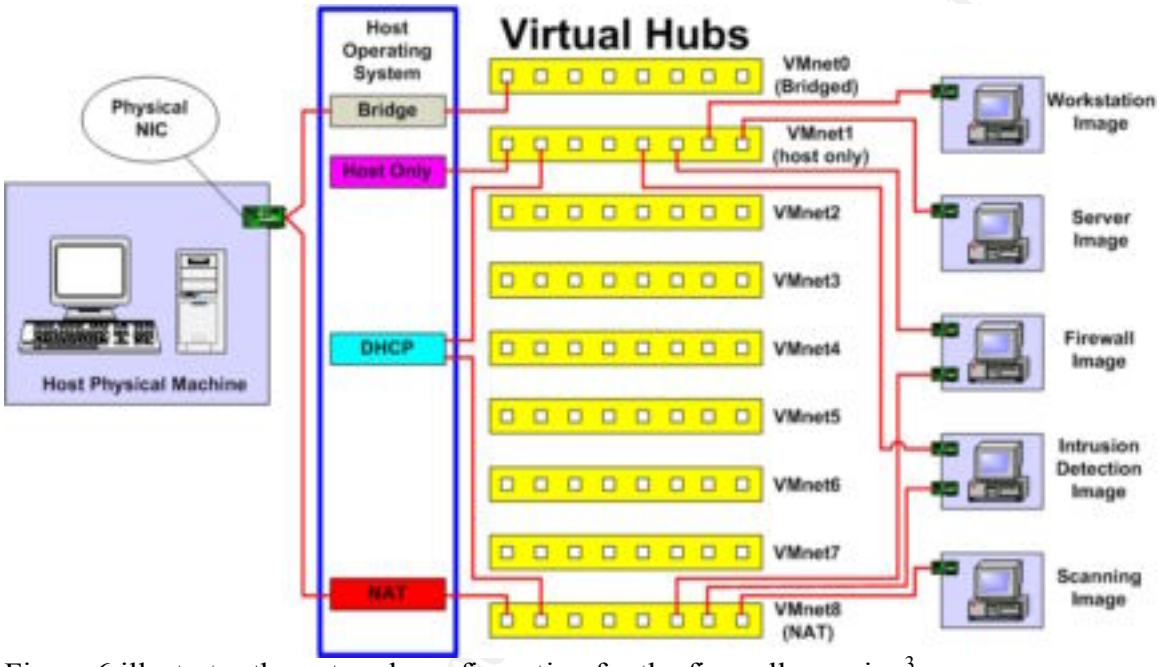


Figure 6 illustrates the network configuration for the firewall exercise.[3]

5.5 Other Exercises

The first four exercises have covered most of the networking features of VMware and different types of Virtual Machine connections that can be made. With this understanding in place you will be able to replicate many other networking environments and conduct further exercises to help improve your understanding of security tools and the environments you are trying to protect.

Here are a few more exercises you may want to try:

- **Honeypots**
  A honeypot is a system that is left vulnerable in some way and then exposed to the internet and closely monitored to gain an understanding of how systems are compromised by attackers. VMware can be a very useful tool if you need to setup a honeypot but be careful this exercise can be very dangerous and should be thoroughly researched before continuing.[8] For example, it is very easy for an attacker to discover that a system is running in a Virtual Machine because the

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

hardware foot print for a given operating system is unique to VMware and always the same.

- **VPN**
  Virtual Private Networks provide a way to create secure connections across the internet and after you have completed the firewall exercise you can add VPN services to a firewall template and configure a workstation image to create a VPN connection across the unsecured virtual hub. After the connection is made an intrusion detection image can be used to observe the traffic or a scanning image used to scan for vulnerabilities.

- **Virus or Attack labs**
  These environments are conducted to gain an understanding of how really vicious programs such as viruses and worms compromise systems. This is a very challenging and very dangerous area. Extensive research should be conducted before attempting to setup this type of environment.[5]

© SANS Institute 2000 - 2002, Author retains full rights.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

List of References

1. Ward, Brian. The Book of VMware: The Complete Guide to VMware Workstation. San Francisco: No Starch Press, 2001.

2. Munro, Jay. "Virtual Machines & VmMare, Part I." 21 December, 2001 URL:http://www.extremetech.com/article/0,3396,s%253D1027%2526a%253D20322,00.asp

3. Munro, Jay. "Virtual Machines & VMware, Part II." 28 December, 2001 URL:http://www.extremetech.com/article/0,3396,s=1027&a=20455,00.asp

4. Cole, Eric. Security Essentials Toolkit. Indiana: Que Publishing, 2002.

5. Bailey, Don. "Attack Lab Design & Security Mini How-To." 6 April, 2002 URL:http://ruff.cs.jmu.edu/~beetle/download/attacklab.html#1

6. Millman, Howard. "Virtual Recovery Via Virtual Servers." 4 March, 2002 URL:http://www.computerworld.com/databasetopics/data/story/0,10801,68722,00.html

7. McDougall, Paul. "Finding Efficiency In The Guts Of A Single Server." 14 January, 2002 URL:http://www.informationweek.com/story/IWK20020111S0034

8. Seifried, Kurt. "Honeypotting with VMware – basics." 15 February, 2002 URL:http://www.seifried.org/security/ids/20020107-honeypot-vmware-basics.html

9. Shankland, Stephen. "Intel servers slice and dice Linux." 23 February. 2002 URL:http://www.zdnetindia.com/techzone/linuxcentre/stories/51131.html

© SANS Institute 2000 - 2002, Author retains full rights.