Installing a Virtual Honeywall using VMware

Spanish Honeynet Project http://www.honeynet.org.es

Diego González Gómez

<dggomez -at- honeynet.org.es>

15 September, 2004. *Last updated:* 14 November, 2004

Abstract

The Honeywall CDROM makes the implementation of a GenII Honeynet Gateway easier. Furthermore, if it is installed on a virtual machine, it will also include the many advantages that a virtual machine environment offers. This paper therefore, explains how to go about configuring **`VMware`** *to deploy a Honeywall.*

1. Introduction

The Honeywall CDROM is a bootable CD with a set of open source tools configured by the Honeynet Project to make the implementation of a GenII Honeynet Gateway easier. Using this document as an installation guide, we are going to implement the Honeywall using the commercial software, **`VMware`** . This document makes a few assumptions, one of them is that you have read and understood the papers Know Your Enemy: Virtual Honeynets, Know Your Enemy: Learning with VMware, Know Your Enemy: Honeywall CDROM.

`VMware` is virtualization software that allows the running of multiple operating systems at the same time on Intel x86 architectures. It was and is developed by VMware Inc. and it has three product lines, namely Workstation, GSX, and ESX. We will be using Workstation. You can download a free evaluation version here.

Several tools included in the CD are only available for GNU/Linux platforms. One of the advantages of using **VMware** is that it permits the implementation of the Honeywall under any operating system supported by this program. Up until the CD was released, the only way to install a complete Honeywall under Windows (with traffic limiting capabilities using iptables, for example) was to first install **VMware** and then configure a GNU/Linux distribution under it. But now, with the introduction of the Honeywall CDROM this task is very straightforward.

Another reason for implementing the Honeywall using **VMware** is that by default, Honeywall uses all resources of the machine you install it on. If a virtual environment is used then this is restricts the Honeywall to use only the resources inside the virtual machine. The advantage is that it is not necessary to consume all the resources of a machine to install the Honeywall CDROM.

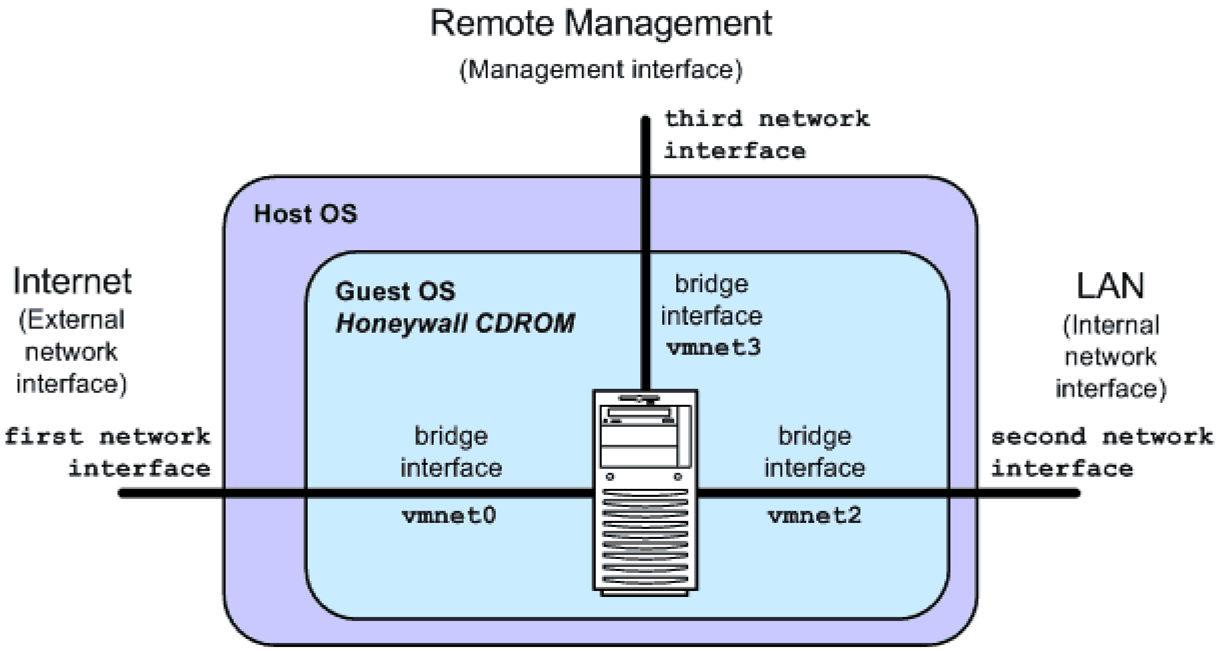
Finally, **VMware** is a good tool for testing purposes. It is an excellent option to develop and experiment with multiple customized Honeywall CDROMs in a controlled environment.

2. Requirements and design considerations

The hardware requirements needed to install the Honeynet Gateway are specified at the Honeywall CDROM page. For proper operation in a Virtual Machine environment, these hardware requirements should be specified in the **VMware** guest OS panel. The Honeywall CDROM version 0.68 does not support SCSI hard disks; therefore IDE devices must be used. The hard disk size depends a lot on the network traffic and the period of time in which the Honeywall is going to be used. The Honeywall uses a minimum of 500 MB of disk space for swap. The remaining space is used for storing the log files.

The overall architecture of the Virtual Honeywall is illustrated in the figure 1. Remember that the Host OS could be any of the supported by **VMware**, based either on Linux or on Windows.

Figure 1. General Architecture



The host machine has three physical network interfaces. The third interface is optional and it is used for remote management. The Guest OS, naturally, is the Honeywall CDROM (based on a modified version of William Salusky's FIRE CD, with Linux kernel version 2.4). Each network adapter of the Guest OS is bridged to its respective physical network interface.

The Honeywall works in bridge mode, since it is the safest and stealthiest option. Therefore, only the network management interface of the Guest OS has an IP stack. However, if deemed necessary the Host OS can define IP addresses for any of its network interfaces without problems. If so, proper security measures must be defined. Naturally, if the Host OS is compromised, the Guest OS is too.

The latest version of the Honeywall CDROM can be configured to periodically send captured information and summary activity reports to a remote system. If the Host OS has an IP address it can be used for Data Collecting, storing the data sent by the Honeywall, and for Data Analysis, allowing the opportunity to carefully examine the collected data.

The IP stack for each network adapter can be disabled following the instructions below:

- In Linux, execute the command '`ifconfig <interface name> 0.0.0.0`'.
- In Windows, open the properties of the network connection to be modified and disable the 'Internet Protocol TCP/IP' option.

One common problem when installing several network interfaces is determining which network interface corresponds to which physical port. The following method is as good as any other:

1. Plug the network cable to only one network interface.
2. Disable all the network interfaces and then, enable and configure with a valid IP address one of them.
3. Try to probe another host in the LAN using the IP destination address, not the DNS name. For example, making a PING to the gateway.
4. If no response is received, unplug the network cable and plug it into consecutive network ports until you are successful. Also make sure that the connections being tried are not blocked by any intermediate network device.
5. Repeat steps 2 to 4 to identify the rest of the interfaces.

3. Virtual honeynets

Depending on the way the virtualization software is used, the Virtual Honeynets can be classified as Self-contained or Hybrid.

Self-contained Virtual Honeynets integrate the whole network into only one physical system. You can read a complete guide for installing a Self-contained Virtual Honeynet, using **`VMware`** and the Honeywall CDROM at the Pakistan Honeynet Project site. These kinds of Virtual Honeynets have several advantages:

- Central management.

- Consolidated Honeynet system.
- Low cost as only one machine is required.
- Portable, if installed for example on a laptop.
- Easier deployment. Only one system needs to be implemented and connected.

However, important limitations should also be kept into account:

- `VMware` only supports x86 platforms. This limits the software that can be used in the Honeynet.
- Any problem with the hardware would affect the entire honeynet.
- A powerful system is needed, depending of the kind and number of Guest OS' used (watch for memory and processor consumption).
- The virtualization software, as in any other software, is susceptible to being compromised by an attacker that could take control over the entire Honeynet.
- An attacker inside a compromised honeypot can easily determine if it is a virtual environment.

Hybrid Virtual Honeynets combine real and virtual systems. Data Capture and Data Control are deployed on an isolated system, and the honeypots are virtual systems run on a single box. In their advantages, they are:

- More secure. The Honeywall is in a separate system, and if it is running in bridge mode the attacker can only view and attack the honeypots.
- More flexible. There are fewer limitations with the software that can be used either in the Honeywall or in the honeypots.

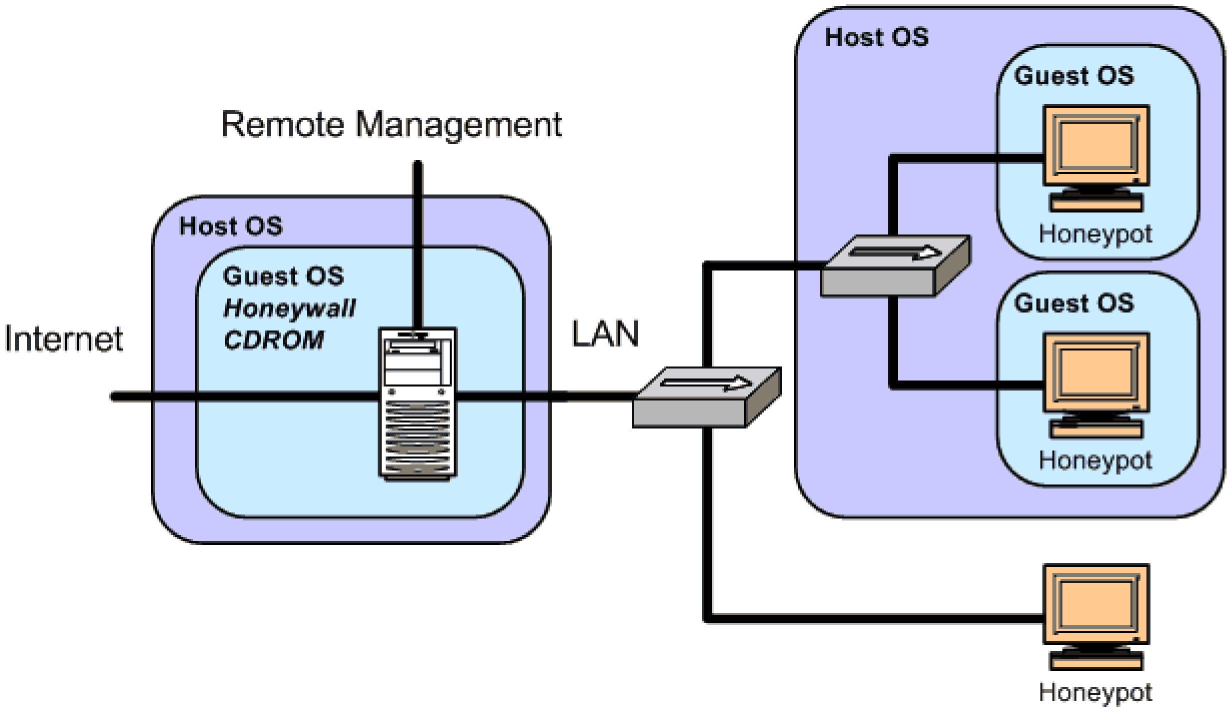
Some disadvantages are:

- They are not very portable, since due to its nature the number of physical systems increases.
- They are more expensive in time, money and space, since it involves the deployment of several devices in the network.

The solution proposed here uses `VMware` just to install the Honeywall. On top of that, the honeypots can be installed in one of the following ways:

1. On another system (using other virtualization software such as `VMware` or User Mode Linux). This would be a Hybrid Virtual Network.
2. A better option, install a honeypot(s) on a set of real systems, to avoid remote fingerprinting issues.
3. A third and last option would consist in a combination of the previous two. See figure 2.

Figure 2. Virtual Honeywall with virtual and real Honeypots



Using **`VMware`** to deploy only the Honeywall may be considered a special case of a Hybrid Virtual Honeynet, because it uses virtualization software for the Honeywall and the honeypots can be virtual or otherwise. It has the following advantages:

- OS independent. The Honeywall can be installed (and tested) on any Host OS supported by the virtualization software, and the equipment used does not lose any existing data.
- More secure. As with the Hybrid Virtual Honeynets, the Honeywall is in a separate system and is running in bridge mode. This permits the attacker to only see and attack the honeypots.
- Flexibility. The honeypots can be installed in a virtual environment or as real systems.

And also these disadvantages:

- It is less portable, since several machines are needed.
- It is more expensive in time, money and space to assign and configure several devices.

4. Configuring VMware

The configuration of **`VMware`** is very straightforward. Once installed, before creating the new Virtual Host, it is necessary to set up the virtual networks.

4.1. Virtual Network configuration

This subsection explains how to configure the network interfaces to work as illustrated in figure 1. The instructions depend on the OS used.

4.1.1. Linux

The `vmware-config.pl` command is used to define the virtual networks. The following output was captured when configuring **VMware** under Linux.

```
[root@hwall vmware]# vmware-config.pl
Making sure services for VMware Workstation are stopped.

[...]

Would you like to skip networking setup and keep your old setting
are?
(yes/no) [no]

Do you want networking for your virtual machines? (yes/no/help) [y

Would you prefer to modify your existing networking configuration
wizard or the editor? (wizard/editor/help) [editor]

The following virtual networks have been defined:

. vmnet0 is bridged to eth0
. vmnet1 is a host-only network on private subnet 10.10.10.0.
. vmnet8 is a NAT network on private subnet 10.10.10.0.

Do you wish to make any changes to the current virtual networks se
(yes/no) [no] yes

Which virtual network do you wish to configure? (0-99) 1

The network vmnet1 has been reserved for a host-only network.  Yo
it, but it is highly recommended that you use it as a host-only ne
you sure you want to modify it? (yes/no) [no] yes

What type of virtual network do you wish to set vmnet1?
(bridged,hostonly,nat,none) [hostonly] none

Removing a host-only network for vmnet1.

The following virtual networks have been defined:

. vmnet0 is bridged to eth0
. vmnet8 is a NAT network on private subnet 10.10.10.0.

Do you wish to make additional changes to the current virtual net
settings?(yes/no) [yes]
```

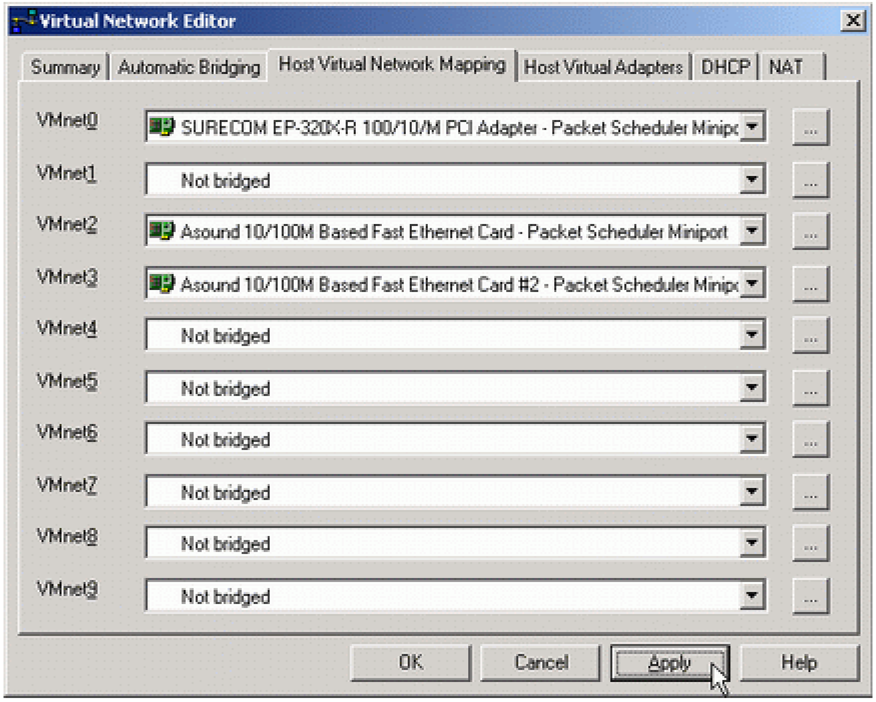
```
Which virtual network do you wish to configure? (0-99) 8

The network vmnet8 has been reserved for a NAT network.  You may c
but
it is highly recommended that you use it as a NAT network.  Are yo
want to modify it? (yes/no) [no] yes

What type of virtual network do you wish to set vmnet8?
(bridged,hostonly,nat,none) [nat] none

Removing a NAT network for vmnet8.

The following virtual networks have been defined:

. vmnet0 is bridged to eth0

Do you wish to make additional changes to the current virtual netw
settings?(yes/no) [yes]

Which virtual network do you wish to configure? (0-99) 2

What type of virtual network do you wish to set vmnet2?
(bridged,hostonly,nat,none) [none] bridged

Configuring a bridged network for vmnet2.

Your computer has multiple ethernet network interfaces available:
Which one do you want to bridge to vmnet2? [eth0] eth1

The following virtual networks have been defined:

. vmnet0 is bridged to eth0
. vmnet2 is bridged to eth1

Do you wish to make additional changes to the current virtual netw
settings?(yes/no) [yes]

Which virtual network do you wish to configure? (0-99) 3

What type of virtual network do you wish to set vmnet3?
(bridged,hostonly,nat,none) [none] bridged

Configuring a bridged network for vmnet3.

The following virtual networks have been defined:

. vmnet0 is bridged to eth0
. vmnet2 is bridged to eth1
. vmnet3 is bridged to eth2
```

```
Do you wish to make additional changes to the current virtual netw
settings?(yes/no) [yes] no

Do you want this program to automatically configure your system to
virtual machines to access the host's filesystem? (yes/no/help) [n

Starting VMware services:
   Virtual machine monitor                                     [  OK
   Virtual ethernet                                            [  OK
   Bridged networking on /dev/vmnet0                           [  OK
   Bridged networking on /dev/vmnet2                           [  OK
   Bridged networking on /dev/vmnet3                           [  OK

The configuration of VMware Workstation 4.5.2 build-8848 for Linux
running kernel completed successfully.

You can now run VMware Workstation by invoking the following comma
"/usr/bin/vmware".

[...]
```

4.1.2. Windows

In the Windows version of **`VMware`**, the virtual network properties are configured with the '**`Virtual Network Editor`**'. Go to '**Host Virtual Adapters**' and remove (if they exist) the network adapters vmnet1 (host-only) and vmnet8 (NAT). Then, go to '**Host Virtual Network Mapping**' and bridge the three network interfaces to `vmnet0`, `vmnet2` and `vmnet3` respectively. The configuration should be similar to the illustrated in figure 3.

Figure 3. Virtual Network Editor - Host Virtual Network Mapping



VMnet0 SURECOM EP-320X-R 100/10/M PCI Adapter - Packet Scheduler Minipc
VMnet1 Not bridged
VMnet2 Asound 10/100M Based Fast Ethernet Card - Packet Scheduler Miniport
VMnet3 Asound 10/100M Based Fast Ethernet Card #2 - Packet Scheduler Minipc
VMnet4 Not bridged
VMnet5 Not bridged
VMnet6 Not bridged
VMnet7 Not bridged
VMnet8 Not bridged
VMnet9 Not bridged

Close the '`Virtual Network Editor`' and open the Windows Services editor to stop and disable '`VMware DHCP service`' and '`VMware NAT service`'.

4.2. Virtual Machine configuration

Create a New Virtual Machine. In order to specify the kind of hard disk used, select the '**Custom**' option during the creation process (remember that the Honeywall CDROM version 0.68 only supports IDE). Check '**Linux**' as the Guest OS, and '**Other Linux 2.4.x kernel**' as the version. At least 256 MB of memory should be specified. Under Network Connection, choose '**Use bridged networking**'. The recommended size of the IDE hard disk is 30 GB, but a lot depends on the amount of traffic expected to be recorded.

Once the new Virtual Machine is created, edit it's settings to add the Ethernet interfaces. First task is to edit the existing network adapter: Under '**Network connection**' choose '**Custom**' and choose `vmnet0` device. Then, we add an additional Ethernet Adapter, choosing `vmnet2` device. Repeat this operation and create another Ethernet Adapter, `vmnet3`. `Vmnet1` adapter is not chosen because it is used by default for host-only networks.

At this point the Guest OS should be ready. If an ISO image is used, instead of a recorded CDROM, the Guest OS will run faster. Lastly, to avoid boot errors, it is highly recommended to change the devices boot order in the BIOS configuration of the Virtual Machine. Open it by pressing F2 at boot time and put the CDROM drive as primary boot device. There is further useful information regarding the implementation of the Honeywall CDROM at it's homepage.

Here are snapshots of the Virtual Honeywall running under both Red Hat Linux and Windows XP.

If the Host OS is Linux and the Honeywall CDROM does not detect the Ethernet Adapters, change the '**Use virtual device**' configuration of the network adapters from '`vlance`' to '`vmxnet`' option.

5. Conclusion

Deploying a Virtual Honeywall combines the advantages offered by the Honeywall CDROM and the virtual environments.

It is very straightforward, it is safe, it can be installed on any OS (supported by **`VMware`**), and it does not need to remove any information previously stored on the host machine. In fact, it leaves the opportunity to use the Host OS to perform Data Collection and Data Analysis functions.