Installing VMWare Server on Debian

The initial version of this article was written in spring 2006 to cover installing VMware Server on Debian GNU/Linux 3.1 (sarge).

In the meanwhile, Debian 4.0 (etch) has been released, making it worth to review the procedure with the current VMware Server and Debian releases.

While for the initial testing a SuperMicro SuperServer 5014C-T with 2 Gigs of RAM, a Pentium 4 CPU @ 2.8 GHz and two 250 GB SATA hard drives (ICH6R, md software raid) was used, the current hardware looks quiet a bit different. Today a more powerful SuperMicro 6014 series server with 4 GiB RAM, two DualCore Xeon CPU @ 2.66GHz (VT enabled) and a 3ware 9000 storage controller (RAID10) was used.

Again Debian was bootstrapped via PXE/NetInstall, downloading everything through FTP.

After the basic install the following steps were taken to get VMware Server installed properly.

#1 Software Dependencies: X

Even if you are installing VMware Server on a host system without X, some dependencies must be fulfilled to ensure proper installation. So this is the list of packages required:

libx11-6
libxtst6
libxt6
libxrender1
libxi6

#2 Software Dependencies: db2

For the MUI package, another dependency must be installed:

libdb2 (for Debian 3.1)
libdb3 (for Debian 4.0)

This provides the libdb3.so.2 shared library.

#3 Software Dependencies: Kernel Source

Since VMware Server uses some kernel modules to provide monitoring and networking functions, you will need the kernel sources or the kernel-headers at hand.

Originally this article covered two methods, however for the sake of simplicity only the straight forward method will be lined out. This is because the second method proved to be too error prone and is in fact only required if you happen to use a custom kernel.

Since I asked Debian to use a 2.6 kernel, I had to install the kernel headers and the kernel build environment.
The steps I took are similar if you use a 2.4 kernel on Debian 3.1.

So first I looked for the 'linux-image' package. I found two package names to be installed:

linux-image

linux-image-2.6

In Debian 3.1 these will be named 'kernel-image' instead, however they serve the same purpose.

To get the kernel headers and the build environment I looked for the 'linux-headers-2.6.18-4', 'linux-headers-2.6.18-4-486' and 'linux-kbuild-2.6.18' packages and installed them.
Debian 3.1 users will look for the similar packages by the name of 'kernel-headers-xyz' and 'kernel-kbuild-xyz'.

Make sure you also install these packages: 'binutils', 'gcc', 'make', 'libc6-dev'

After installing these packages, look at your modules directory at /lib/modules/`uname -r`.
There should be a symlink directory called 'build' pointing to '/usr/src/kernel-headers-2.6.18-4-686'.
If it would not exist or point to a different location, recreate the symlink to the proper header directory.

The kernel header directory (/usr/src/kernel-headers-2.6.18-4-686 or /lib/modules/`uname -r`/build/include) will be required during VMware installation later on.

#4 Software Dependencies: perl + CPAN

A fresh Debian installation is missing pretty much of everything. So you need to make sure that Perl is installed to including the whole base package.

Just check for the 'perl' and 'perl-modules' package and install them if required (vanilla Debian usually has only 'perl-base' installed).

You may find it valuable to also install some additional dependencies for CPAN first:

wget
lynx
unzip
gnupg
ncftp
ftp
curl

Now I would strongly recommend to configure CPAN on your host:

```
perl -MCPAN -e shell
```

Follow the configuration dialogs for CPAN (you *may* just press ENTER on every question as they provide you with typical default values -- however it ain't wrong to check out CPAN documentation first).

When you're done with that, try installing the CPAN bundle. Type this in the CPAN shell:

```
install Bundle::CPAN
```

This will make sure you have the current CPAN modules at hand. It will ask you to 'follow the dependency' chain. Just confirm that to download and install all required modules.

As soon as it finished, try to install the MakeMaker package:

install ExtUtils::MakeMaker

This particular module is required for the VMware Perl API.

#5 Getting and Installing VMware Server

Now download the VMware packages. Use the tar files and take both the 'VMware-server' and the 'VMware-mui' package.

Right, you *can* leave the latter one away as it only provides the web-based frontend system. The essentials for running VMware are in the server package.

Untar the VMware-server tarball to a temporary location and run the vmware-install.pl script as stated by the VMware documentation.

NB: One Debian 4.0 I encountered a situation where vmware-install.pl mourned about 'killall' being missing. Simply installing it by means of 'apt-get install psmisc' solved this easily.

When asked for the 'kernel headers' simply provide the directory containing the kernel source or the kernel headers. You should have come along this in step #3 already.

It will now try to compile and load the kernel modules. If this step fails with an error message about failing to load the module into the kernel, double-check your kernel header and kernel source to match your currently installed kernel.
If you have met all the requirements this step should complete smoothly.

Using '/etc/init.d/vmware' you can start/stop the server, though I my experience was that wouldn't start until I rebooted the host at least once.

To install the MUI package repeat the above steps according to the setup instructions for the MUI package.

To start/stop the web ui use '/etc/init.d/httpd.vmware'.

#6 Accessing VMware Server

If you installed the MUI package, you should find your host system serving a page at https://IP:8333.

You need to use your 'root' login to access the management interface.

There you can also download the VMware console package for Linux or Windows.
You will need this to manage your VMware server as the web frontend does only allow starting/shuttind down/rebooting vm's and status monitoring.

To connect using the VMware console just provide the server IP and the root login. In case it wouldn't work, make sure no firewall blocks traffic from/to port 902.

#7 Securing your Host

A plain Debian installation is pretty much secure as it doesn't run lots of fuzzy services as other distros do.

So after installing VMware you should only have port 902/tcp to listen, additionally 8222/tcp and 8333/tcp when the MUI was also installed.

If you didn't pay attention to the mta during installation, you may end up with exim listening on port 25/tcp leaving your host as an open relay.

I usually configure it for local delivery so it will listen to port 25/tcp on localhost only.

For remote administration, you may want to install the 'ssh' package (Debian 3.1) or 'openssh-server' and 'openssh-client' (Debian 4.0) respectively.

So all in all, you will only have a few ports open. I would stronly recommend to protect your ssh, smtp and vmware ports through iptables from unwanted access.
You may want to use the 'ferm' toolkit for easy but powerful creation of firewall rules.

If you are using VMware using bridged networking instead of host-only or nat, you can even think of bringing the physical host interface (eg. eth0) up without an ip address. This will reduce the risk of attacks to the host system itself. Just plugin a second NIC and attach it to a dedicated LAN (eg. non-public) for management purposes.

#8 Observations

It's been some time that VMware Server has become a production grade product.

It has definitely matured since the beta releases and all the debugging code has gone.

Still with the availability of VT or similiar virtualization extensions native to the CPU performance impacts have become less critical than before.