

multiplicity.bsd.lv: The NetBSD Xen Guide

1 - The NetBSD Xen Guide

1.1 - Overview

This guide is intended for users who would like to install and operate a NetBSD Xen 3 Dom0 host with unprivileged and privileged guests but have never used NetBSD. It is designed to work in conjunction with the [NetBSD 3.1 and 4.0 Installation Guide](#). This guide will assume that NetBSD 4.0 i386 is installed and has Internet access.

Additional NetBSD-oriented Xen information:

- [NetBSD/xen HOWTO](#)
- [How to set up a guest OS using xen3](#)
- [Xen Garden aka: Running multiple NetBSD virtual machines under Xen](#)
- [Virtual LAN with Xen3](#)
- [Port-xen mailing list archives](#)
- [The Xen Users' Manual v3.0](#)

A few things you should know:

- The NetBSD bootloader does not support the Xen kernel and you will need to use [GNU GRUB](#).
- The GRUB documentation, particularly with regards to NetBSD is both dated and inconsistent.
- The reported problem of GRUB requiring a root partition of less than 504M appear to be resolved by uncompressing the Xen kernel.

Various packages will display useful information upon installation and that information will be included as appropriate.

1.2 - Required packages and kernels

See the related [NetBSD 3.1 and 4.0 Installation Guide](#) for basic package management information.

- The GRUB package: [sysutils/grub](#).
- The Xen 3 kernel: [sysutils/xenkernel3](#) (requires python24).
- A Dom0 host without HVM support requires xentools3: [sysutils/xentools3](#) (requires expat, readline, python24, py24-readline, and py24-xml).
- A Dom0 host with HVM support requires xentools3-hvm: [sysutils/xentools3-hvm](#) (requires xentools3 and devel/SDL, which in turn requires audio/arts, audio/esound, audio/nas, graphics/aalib, graphics/glu, graphics/MesaLib which may in turn depend on others).

Finally, you will need retrieve, install and uncompress the NetBSD Dom0 kernel:

```
# ftp ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-4.0/i386/binary/kernel/netbsd-XEN3_DOM0.gz
# cp netbsd-XEN3_DOM0.gz /
# gunzip /netbsd-XEN3_DOM0.gz
```

1.3 - GRUB configuration, installation and testing

Be sure to test GRUB> with the NetBSD kernel before attempting to configure and use it with a Xen kernel. The GRUB package includes the `grub-install` script which will create and populate the `/grub` directory but not create a configuration file. Note that you may want to create the `/grub/` directory and `/grub/menu.lst` before running `grub-install` as not to run the risk of having an unbootable system during the configuration.

`grub-install` can be run as either:

```
# grub-install /dev/wd0d

-or-

# grub-install '(hd0)'
```

Optionally, the same steps can be performed manually:

```
# mkdir /grub
# cp -p /usr/pkg/lib/grub/i386-/* /grub
# grub --no-floppy
grub> root (hd0,a)
grub> setup (hd0)
grub> quit
```

The GRUB configuration file will vary in name and location on various operating systems and the NetBSD package specifies `/grub/menu.lst`.

Note that this reads ".LST" and not ".1ST". Create this file and add the following:

```
default=0
timeout=5
title NetBSD 4.0
root(hd0,0,a)
kernel /netbsd
title chainloader
root(hd0,0,a)
```

```
chainloader +1
```

This file is read at boot time and any modifications will be interpreted automatically. GRUB need not be re-installed for changes to take effect.

The first option specifies the NetBSD kernel while the `chainloader +1` option instructs grub to attempt booting from the boot sector (one sector from the start) of the specified partition.

Reboot your system and verify that both options successfully boot the system.

Note: The grub binary is dynamically-linked, should you ever attempt to use it to repair a misconfigured system.

NB! The `/usr/pkg/lib/grub/i386-/stage2` in the [sysutils/grub](#) package [does not support](#) Apple x86 hardware. This can be resolved by retrieving a `stage2` binary from a Linux live CD that works on given hardware and copying it to both `/usr/pkg/lib/grub/i386-/` and `/grub/`. A compatible `stage2` can be found [here](#).

This file is likely located at `/usr/lib/grub/i386-pc/stage2`.

1.4 - Xen Dom0 Configuration

Begin by verifying that [mount_kernfs\(8\)](#) is mounted as it will be required by the Xen tools. It is mounted by default in NetBSD 4.0.

```
# mount
/dev/wd0a on / type ffs (NFS exported, local)
kernfs on /kern type kernfs (local)
```

Create the Xen devices:

```
# cd /dev && sh MAKEDEV xen
```

Install the Xen daemons `xenbackendd`, `xend` and `xendomains`:

```
# cp /usr/pkg/share/examples/rc.d/xen* /etc/rc.d/
```

Enable the Xen daemons by adding the following lines to `/etc/rc.conf`:

```
xend=YES
xenbackendd=YES
xendomains=YES
```

Enable bridged networking by creating `/etc/ifconfig.bridge0` containing:

```
create
!brconfig $int add fxp0 up
```

Replace `fxp0` with the name of the appropriate network device on your system.

Install and uncompress the Xen kernel:

```
# cp /usr/pkg/xen3-kernel/xen.gz /
# gunzip /xen.gz
```

Finally, add the following lines to `/grub/menu.lst`:

```
title Xen 3.0 / NetBSD 4.0
root(hd0,0)
kernel (hd0,a)/xen dom0_mem=65536
module (hd0,a)/netbsd-XEN3_DOM0 root=/dev/wd0a ro console=tty0
```

NB! The [NetBSD/xen HOWTO](#) specifies `bootdev=/dev/wd0a` in place of `root=/dev/wd0a` and this may not work.

Finally, reboot and test the `Xen 3.0 / NetBSD 4.0` boot option. This can be changed to be the default by either changing the `default=` option in `/grub/menu.lst` or placing it above the `NetBSD` entry. The final `/grub/menu.lst` looks like:

```
default=0
timeout=5
title Xen 3.0 / NetBSD 4.0
root(hd0,0)
kernel (hd0,a)/xen dom0_mem=65536
module (hd0,a)/netbsd-XEN3_DOM0 root=/dev/wd0a ro console=tty0
title NetBSD 4.0
root(hd0,0,a)
kernel /netbsd
title chainloader
root(hd0,0,a)
chainloader +1
```

Verify that the Xen kernel is booted:

```
# xm list
Name                                ID    Mem VCPUs    State    Time(s)
Domain-0                            0     64      1    r-----    9.9
```

To determine if HVM support is enabled on Intel hardware for use with unprivileged DomU's:

```
# xm dmesg | grep VMX
(XEN) HVM: VMX enabled
(XEN) VMX: MSR intercept bitmap enabled
```

To determine if HVM support is enabled on AMD hardware for use with unprivileged DomU's:

```
# xm dmesg | grep SVM
(XEN) HVM: SVM enabled
```

You will need to enable SVM in BIOS if you receive:

```
(XEN) AMD SVM Extension is disabled in BIOS.
```

1.5 - Unprivileged Xen DomU Configuration

The `xentools3` environment consists of these files and scripts: `/usr/pkg/etc/xen/scripts`, `/usr/pkg/etc/xen/xend-config.sxp` and `/usr/pkg/share/examples/xen/`.

The daemons: `xenbackendd`, `xend`, and `xendomains` in `/usr/pkg/share/examples/rc.d`

These utilities with man pages: `/usr/pkg/sbin/xm` and `/usr/pkg/sbin/xentop`

These configuration man pages: `xend-config.sxp.5` and `xmdomain.cfg.5`

And these support utilities in `/usr/pkg/bin/`:

```
xc_restore
xc_save
xencons
xenconsole
xenperf
xenstore-chmod
xenstore-control
xenstore-exists
xenstore-list
xenstore-ls
xenstore-read
xenstore-rm
xenstore-write
xen-detect
```

To these, the `xentools3-hvm` packages adds:

```
/usr/pkg/lib/xen/boot/hvmloder
/usr/pkg/libexec/qemu-dm
/usr/pkg/share/examples/xen/hvm
/usr/pkg/share/examples/xen/qemu-ifup
/usr/pkg/share/xen/qemu/keymaps/...
```

Paravirtualized DomU domains typically consist of a kernel, a configuration file, and a disk image containing the filesystem for the domain. Of these, only the configuration files have a default location of:

```
/usr/pkg/etc/xen/
```

Pre-compiled NetBSD 3.1 and 4.0 Xen 3 DomU kernels are available at:

```
ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-3.1/i386/binary/kernel/
ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-4.0/i386/binary/kernel/
```

The NetBSD 3.1 and 4.0 Xen DomU kernels share the same name and include a special installation kernel with the behavior of booting from an installation CD:

```
netbsd-INSTALL_XEN3_DOMU.gz
netbsd-XEN3_DOMU.gz
```

If you plan to use a mix of NetBSD 3.1 and 4.0 DomU domains, you may want to distinguish the kernels by name:

```
netbsd-3.1-INSTALL_XEN3_DOMU.gz
netbsd-4.0-INSTALL_XEN3_DOMU.gz
netbsd-3.1-XEN3_DOMU.gz
netbsd-4.0-XEN3_DOMU.gz
```

For this guide we will locate them in the custom directories:

```
/var/xen/kernels/
```

And disk images in:

```
/var/xen/images/
```

To create a 2048M image for use with a Dom0:

```
# dd if=/dev/zero of=/var/xen/images/xen-2GB.img bs=1024 count=1 seek=2000000
```

Finally, to create a NetBSD 3.1 DomU instance using the image we just created, the configuration file `/usr/pkg/etc/xen/netbsd-3.1.cfg` should contain:

```
kernel = "/var/xen/kernels/netbsd-3.1-INSTALL_XEN3_DOMU.gz"
#kernel = "/var/xen/kernels/netbsd-3.1-XEN3_DOMU.gz"
memory = 64
name = 'NetBSD-3.1'
vif = [ 'mac=00:16:3e:00:00:11, bridge=bridge0' ]
disk = [ 'file:/var/xen/images/xen-2GB.img,0x1,w' ]
root = "xbd0"
# Optional boot parameters such as "-s" for single user mode
#extra = ""
```

The disk image in the above example will appear as `/dev/xbd0a`

Note: Most documentation specifies `root = "/dev/wd0d"` and this will also work.

NB! Verify your disk image paths or DomU kernel will stop at:

```
xennet0: using event channel 6
unknown type console at xenbus0 id 0 not configured
unknown type console at xenbus0 id 0 not configured
backend_changed backend
```

Specifying more than the available memory will return the error:

```
Error: [Errno 2] No such file or directory: '/proc/xen/balloon'
```

Additional disk entries:

A disk image plus the physical CD-ROM device according to some:

```
disk = [ 'file:/var/xen/images/xen-2GB.img,0x1,w' , 'phy:/dev/cd0a,0x2,r' ]
```

A disk image plus the physical CD-ROM device according to others:

```
disk = [ 'file:/var/xen/images/xen-2GB.img,0x1,w' , 'phy:/dev/cd0a,0x04,r' ]
```

A disk image plus the physical CD-ROM device according to yet others:

```
disk = [ 'file:/var/xen/images/xen-2GB.img,0x1,w' , 'phy:/dev/cd0a,cd0a,r' ]
```

A disk image plus a CD-ROM image:

```
disk = [ 'file:/var/xen/images/xen-2GB.img,0x1,w' , 'file:/var/xen/images/i386cd-3.1.iso,0x04,r' ]
```

The same with an alternative CD-ROM parameter:

```
disk = [ 'file:/var/xen/images/xen-2GB.img,0x1,w' , 'file:/var/xen/images/i386cd-3.1.iso,cd0a,r' ]
```

The resulting DomU domain can be created and attached to the console with:

```
# xm create -c /usr/pkg/etc/xen/netbsd-3.1.cfg
```

If planning to install from physical CD or ISO image, you will need to manually select the CD-ROM device to:

```
device:      cd0a
set directory: /i386/binary/sets
```

to:

```
device:      xbd1d
set directory: /i386/binary/sets
```

If that fails, try:

```
device:      xbd1a
set directory: /i386/binary/sets
```

You can detach from the DomU's console with `CTRL-]`

NetBSD can be installed into an unprivileged DomU by using the `netbsd-3.1-INSTALL_XEN3_DOMU.gz` kernel and either a CD-ROM or network option for the distribution sets.

Once the OS is installed to the DomU, you can specify normal booting by changing the `kernel` entry in `/usr/pkg/etc/xen/netbsd-3.1.cfg` from `netbsd-3.1-INSTALL_XEN3_DOMU.gz` to `netbsd-3.1-XEN3_DOMU.gz`

DomU can be fully shut down from its console with:

```
# shutdown -p now
```

Useful general `xm(1)` commands are:

```
# xm list
# xm info
# xm top
# xm help
```

Useful DomU `xm(1)` management commands that take a specific DomU Name or ID are:

```
# xm shutdown
# xm destroy
# xm reboot
# xm pause
# xm unpause
# xm console
```

Note that `xm reboot` does not appear to re-read the configuration file.

If you want a domain to be created at host system startup, verify that its configuration file is `/usr/pkg/etc/xen/` and add it's name to `/etc/rc.conf` after the `xendomains=YES` directive as:

```
xendomains=YES
xendomains='netbsd-3.1.cfg'
```

Multiple domains can be separated with spaces as: `xendomains='netbsd-3.1.cfg netbsd-4.0.cfg'`

It is recommended that you add `power=YES` to the DomU's `/etc/rc.conf` allow proper shutdown if `xm shutdown -R` Or `xm shutdown -H` are executed from the Dom0.

To suppress `wscons` warnings, disable all of the DomU's tty's by changing `/etc/ttys`:

```
console "/usr/libexec/getty Pc"      vt100  on secure
ttyE0   "/usr/libexec/getty Pc"      vt220  off secure
ttyE1   "/usr/libexec/getty Pc"      vt220  on secure
ttyE2   "/usr/libexec/getty Pc"      vt220  on secure
ttyE3   "/usr/libexec/getty Pc"      vt220  on secure
...
```

to:

```
console "/usr/libexec/getty Pc"      vt100  on secure
ttyE0   "/usr/libexec/getty Pc"      vt220  off secure
ttyE1   "/usr/libexec/getty Pc"      vt220  off secure
ttyE2   "/usr/libexec/getty Pc"      vt220  off secure
ttyE3   "/usr/libexec/getty Pc"      vt220  off secure
...
```

Also comment out all of the DomU's screens in `/etc/wscons.conf` by changing:

```
...
#screen 0      -      vt100
screen 1      -      vt100
screen 2      -      vt100
screen 3      -      vt100
screen 4      -      -
#screen 4      80x25bf vt100
...
```

to:

```
...
#screen 0      -      vt100
#screen 1      -      vt100
#screen 2      -      vt100
#screen 3      -      vt100
#screen 4      -      -
#screen 4      80x25bf vt100
...
```

Thanks to bridged networking, network configuration on the DomU is identical to a normal system.

1.6 - Privileged Xen DomU Configuration

If your hardware supports `vmx` or `svm` extensions, the above configuration example with a CD-ROM image can be modified for privileged use as follows:

```
kernel = "/usr/pkg/lib/xen/boot/hvmloader"
builder = "hvm"
memory = 64
name = "NetBSD-3.1"
#vif = [ 'mac=00:16:3e:00:00:17, bridge=bridge0, type=ioemu, model=ne2k_pci' ]
#vif = [ 'mac=00:16:3e:00:00:13, bridge=bridge0, type=ioemu, model=pcnet' ]
vif = [ 'mac=00:16:3e:00:00:13, bridge=bridge0, type=ioemu' ]
device_model = '/usr/pkg/libexec/qemu-dm'
disk = [ 'file:/var/xen/images/xen-2GB.img,ioemu:hda,w' , 'file:/var/xen/images/i386cd-3.1.iso,ioemu:hdc:cdrom,r' ]
# Optional floppy image
#fda = '/var/xen/images/boot1.fs'
# Boot device: a = Floppy, c = hard Drive, d = CD-ROM
boot = "d"
# Attach to an XWindow via SDL
sdl = 1
# Alternatively attach to VNC
#vnc = 1
#vncdisplay = 0
#vncunused = 0
# Allow use of a USB pointing device (tablet is recommended)
usb = 1
usbdevice = "tablet"
#usbdevice = "mouse"
```

Note that you may need to specify `boot = "c"` if you only specify a CD-ROM in `disk =` entry. Placing the CD-ROM entry before the hard drive entry may not allow the hard drive to be selected for booting with either "c" or "d".

See the example `/usr/pkg/share/examples/xen/hvm` for expanded explanations.

This example should work from an XWindow session with:

```
xm create /usr/pkg/etc/xen/netbsd-3.1.cfg
```

Type `CTRL-ALT` to release the mouse if captured by the display window.

1.7 - For the adventuresome

NetBSD 4.0 and OpenBSD 4.3 will work as a privileged DomU's. You may experience some display issues, X session termination with the error `mmapbatch: remap error 14!` if you change virtual consoles, plus the [network error](#) `re0: watchdog timeout`. One fix is to use the NE2000 network interface which require additional CPU load on Dom0:

```
vif = [ 'mac=00:16:3e:00:00:11, bridge=bridge0, type=ioemu, model=ne2k_pci' ]
```

Alternatively, disabling NetBSD's `re(4)` network device on the DomU's kernel with `userconf(4)` or `boot -c` at startup should force use of the preferable `rtk(4)` network device. OpenBSD does not appear to include the `rtk(4)` device. The `pcn(4)` driver appears to work under OpenBSD but has been reported to be problematic in the past.

An OpenBSD DomU can be fully shut down with `shutdown -ph now`

FreeBSD 6.3 and 7.0 displayed an [infinite reboot](#) at the bootloader.

The following privileged configuration should theoretically boot a live CD but produced system freezes on a Mac mini Core Duo:

```
kernel = "/usr/pkg/lib/xen/boot/hvmloader"
builder = "hvm"
memory = 128
name = "livecd"

vif = [ 'mac=00:16:3e:00:00:17, bridge=bridge0, type=ioemu, model=ne2k_pci' ]
#vif = [ 'mac=00:16:3e:00:00:13, bridge=bridge0, type=ioemu, model=pcnet' ]
#vif = [ 'mac=00:16:3e:00:00:13, bridge=bridge0, type=ioemu' ]
device_model = '/usr/pkg/libexec/qemu-dm'

disk = [ 'phy:/dev/cd0a,ioemu:hdc:cdrom,r' ]

boot = "d"

sdl = 1
usb = 1
usbdevice='tablet'
```

The OpenBSD unprivileged Xen DomU instructions specify:

Install the `devel/mercurial` port.

Retrieve the OpenBSD Xen DomU sources with:

```
# hg clone http://hg.recoil.org/openbsd-xen-sys.hg
```

They can be updated with:

```
# hg pull -u
```

And built with:

```
# cd openbsd-xen-sys.hg/arch/xen
# ln -s conf.i386 conf
# cd conf
# config <kernel-file>
# cd ../compile/<kernel-file>
# make depend
# make
```

FreeBSD as a Guest OS

<http://www.freebsd.org/doc/en/books/handbook/virtualization-guest.html>

FreeBSD 5.3 Xen DomU Install HOWTO

<http://txrx.org/xen/>

multiplicity.bsd.lv: The NetBSD Xen Guide

Copyright © 2008 [Maikls Deksters](#) Last modified: \$Date: 2008/06/10 14:09:32 \$