

## FreeBSD PXE SERVER HOWTO

---

---

First of all after a freebsd fresh install I install the dhcp server port :

```
cd /usr/ports/net/dhcp30-server  
make install clean
```

Later I configure dhcpd.conf like :

```
allow booting;  
allow bootp;  
authoritative;  
  
option domain-name "freebsdpxe.ramattack.net";  
option subnet-mask 255.255.255.0;  
default-lease-time 600;  
max-lease-time 7200;  
ddns-update-style ad-hoc;  
log-facility local7;  
local-address 10.0.0.1;  
  
subnet 10.0.0.0 netmask 255.255.255.0 {  
    range 10.0.0.70 10.0.0.80;  
    next-server 10.0.0.1;  
    filename "freebsd8/boot/pxeboot";  
    option root-path "/expert/netboot/freebsd8";  
}
```

I will use the tftp-server that comes with freebsd... so in inetd.conf :

Uncomment this :

```
tftp    dgram    udp        wait       root       /usr/libexec/tftpd      tftpd -l -s  
/expert/netboot
```

In /etc/exports :

```
/expert -alldirs,ro -network 10.0.0 -mask 255.255.255.0
```

Yes readonly... nothing should be written to our nfs exported content.

Now let's populate /expert/netboot/freebsd8 with freebsd disc 1 :

```
tar -C /expert/netboot/freebsd8 -pxvf 8.0-RELEASE-amd64-disc1.iso
```

Now from sources let's build pxeboot and later boot stages for booting with nfs support.

NOTE : It's important to use the sources available at the installation medium of the current machine in which we are working on. For avoiding problems basically and doing things properly.

```
# cd /usr/src/sys/boot
# make
```

Now let's copy files that are relevant for booting :

```
# cp i386/pxeldr/pxeboot /expert/netboot/freebsd8/boot/
# cp i386/boot0/boot0 /expert/netboot/freebsd8/boot/
# cp i386/boot2/boot1 /expert/netboot/freebsd8/boot/
# cp i386/boot2/boot2 /expert/netboot/freebsd8/boot/
# cp i386/mbr/mbr /expert/netboot/freebsd8/boot/
```

We copy pxeboot, boot0, boot1, boot2 and mbr... I think really boot0 it's not necessary because we go from pxeboot (which is just previous to boot1 which loads BIOS loader for later loading loader in boot2).

Now continue copying important files :

Let's copy device.hints for booting the kernel loaded by loader in this pxe boot. Let's copy too some routine files for loader (and for forth shell) and a loader defaults config file.

```
# mkdir -p /expert/netboot/freebsd8/boot/defaults
# cp /usr/src/sys/amd64/conf/GENERIC.hints
/expert/netboot/freebsd8/boot/device.hints
# cp /usr/src/sys/boot/forth/loader.conf /expert/netboot/freebsd8/boot/defaults/
# cp /usr/src/sys/boot/forth/loader.4th /expert/netboot/freebsd8/boot/defaults/
# cp /usr/src/sys/boot/forth/support.4th /expert/netboot/freebsd8/boot/defaults/
```

Now I'm ensuring that /expert/netboot/freebsd8/boot/loader.rc to have the following lines :

```
include /boot/loader.4th
start
```

Which should be because loader4.th is the loader forth routines file... and the other instruction goes to the routine which basically loads /boot/kernel...

Later in loader.conf just to be this lines :

```
mfsroot_load="YES"
mfsroot_type="mfs_root"
mfsroot_name="/boot/mfsroot"
vfs.root.mountfrom="ufs:/dev/md0"
```

Yes :) without the last "c" md0c meant that the device we are talking about should be used from the first block but this is obsolete in freebsd8 as far as I have read... It works with both but as far as I know just for backwards compatibility... so without the last c.

Now let's do an automated sysinstall :

```
cd /expert/netboot/freebsd8/boot
gzip -d mfsroot.gz
```

loader won't be able to uncompress it and it's essential for booting.

Now for being able to automate sysinstall partitioning and so... sysinstall searches for install.cfg file in the root of the memory filesystem provided by mfsroot... mfsroot is a memory disk file for loader

to be able to have a / partition... so we need to attach it to a device memory disk for the kernel and later mount that attached memory disk for copying there the install.cfg file :

```
mkdir /puntomontajes
mdconfig -a -t vnode -f /expert/netboot/freebsd8/boot/mfsroot -u 200
mount /dev/md200 /puntomontajes
```

( I have used the 200 number but really it's optional you could not specify -u \_\_\_\_ and it will be attached to the first free in numerical order kernel memory disk)

```
cp /locationofourdefinedinstallcfg/install.cfg /puntomontajes
```

done !! :)

Now :

```
cd /
umount /puntomontajes
After ensuring it's unmounted... let's remove our attached kernel memory disk...
mdconfig -d -u 200
```

We ensure it's removed with :

```
mdconfig -l
```

Now finally let's configure /etc/rc.conf of our pxe server for launching all services automatically :

```
## PXE services
```

```
dhcpd_enable="YES"
dhcpd_ifaces="fxp0"
inetd_enable="YES"
rpcbind_enable="yes"
mountd_enable="yes"
nfs_server_enable="yes"
```

Now important... without doing this tftp server won't be able to find pxeboot under /expert/netboot/freebsd8/boot/ :

```
chmod -R 755 /expert/netboot/freebsd8
```

That's all :).

Any doubts you can write me to [egoitz@ramattack.net](mailto:egoitz@ramattack.net).