

PXE network boot server

Howto: Turn your Funtoo machine into a Network Boot Server

This guide helps explain how to set up a PXE server using `in.tftpd` and `dnsmasq`. This may be useful for installing an operating system on a machine that has no optical drive and/or an older BIOS which doesn't support booting from USB.

This guide will cover the basics of getting your server set up to allow clients to boot from the network to a `pxelinux/syslinux` menu and choose an option of installing / running your preferred distribution or installing a MS Windows operating system - the possibilities are endless and you are free to use it as you wish!! The funtoo way!

Dependencies

The following packages are required:

- `Dnsmasq` (/Package:Dnsmasq)
- `net-ftp/tftp-hpa` (package not on wiki - please add (/Adding_an_Ebuild_to_the_Wiki))
- `sys-boot/syslinux` (package not on wiki - please add (/Adding_an_Ebuild_to_the_Wiki))

For Windows PXE Booting:

- `net-fs/cifs-utils` (package not on wiki - please add (/Adding_an_Ebuild_to_the_Wiki))
- `Samba` (/Package:Samba)
- A simple fileserving protocol configured and working properly. Both FTP and HTTP work fine. You can use either one.

Note

This guide will use System Rescue CD as an example of the PXE Boot Process.

- Download System Rescue CD[1] (<http://www.sysresccd.org>)

Note

The following packages are only required if you intend to install Microsoft Windows via Net

- NFS support - Kernel configuration : `CONFIG_NFS_FS=y` | `m`

Understanding the PXE/Network Boot process

A PXE Network boot isn't much different than a traditional boot from your hard drive, in fact you will probably find the boot loader to be very similar to what you are already familiar with! In a nut shell here is what happens: You set your BIOS / Boot options to boot from Network. The client will obtain an IP address from the PXE server (via `DNSMasq`), after the IP address is obtained it simply looks for the `tftp` daemon running on the server (`tftp-hpa`). The DHCP Server sends the PXE information to the NIC and it loads up a menu that you define in your `pxelinux` configuration (`syslinux`), or depending on your configuration it may go straight into the OS / Installation that you

configure.

Sounds easy huh? For the most part it is very simple to set up. However if you plan to set up a MS Windows install via the network, it gets a bit more tricky, mainly due to MS not using a case sensitive file system, and requiring files to be located using drive letters and back slashes "\" instead of slashes "/" What this requires is a remapping file. With a remapping file, tftp daemon will remap the characters, symbols, and/or drive letters to suit the needs. This is why I recommend tftp-hpa in this guide.

DNSMasq actually provides a tftp server if you want to use it, however I recommend the use of tftp-hpa as it allows remapping in the event you ever intend to boot a Windows environment over your network.

This guide will cover the basics of getting your PXE server up and running for a linux based client, in this guide we will be using System Rescue CD (which is a Gentoo Live CD Image).

In the event that you want to install Microsoft Windows over the network, you will have your server already configured, and you will only need to do some minor config changes, host your Windows installation files / Preinstallation Environment and set your tftp remapping configuration. (This can become a headache if you plan to host several releases of MS Windows over the network - due to conflicting remappings)

It is also very important to understand that the PXE network is only responsible for giving you network access up until the operating system is loaded. What this means is that your kernel you are loading will need to have support for the network card on your client(s). You may want to consider a generic kernel that supports several different NICs. For Windows, this means you will probably want to include all NIC drivers in your installation files and ensure that they are loaded during installation.

Installing and Configuring tftp-hpa (in.tftpd) for serving your network boot files

Install net-ftp/tftp-hpa (package not on wiki - please add (/Adding_an_Ebuild_to_the_Wiki)) using portage:

```
# emerge net-ftp/tftp-hpa
```

Create a directory for your tftp server - this is where your pxe configuration files and any files that will be accessed directly from the PXE boot process will be located (You can put it anywhere you have access to, I will be using /tftpboot):

```
# mkdir /tftpboot
```

Edit your PXE configuration: Set the path to /tftpboot or your preferred directory created above. We are going to also go ahead and add a remapping file just in case you intend to use it later it will be \${INTFTPD_PATH}tftpd.remap. Edit /etc/conf.d/in.tftpd:

```
# Path to server files from
# Depending on your application you may have to change this.
# This is commented out to force you to look at the file!
#INTFTPD_PATH="/var/tftp/"
#INTFTPD_PATH="/tftpboot/"
INTFTPD_PATH="/tftpboot/"

# For more options, see in.tftpd(8)
# -R 4096:32767 solves problems with ARC firmware, and obsoletes
# the /proc/sys/net/ipv4/ip_local_port_range hack.
# -s causes $INTFTPD_PATH to be the root of the TFTP tree.
# -l is passed by the init script in addition to these options.
INTFTPD_OPTS="-m ${INTFTPD_PATH}tftpd.remap -R 4096:32767 -s ${INTFTPD_PATH}"
```

No need to worry about the contents of the tftpd.remap file for now, but to prevent the daemon from panicking on a missing file, just create an empty one like so:

```
# touch /tftpboot/tftpd.remap
```

Installing and Configuring DNSMasq for DHCP / PXE Booting

Install Dnsmasq (/Package:Dnsmasq) if you don't already have it installed (use the tftp useflag): Even though we won't be using the built-in tftp server for dnsmasq, we will still need it to be tftp-aware:

```
# echo "net-dns/dnsmasq tftp" >> /etc/portage/package.use/dnsmasq
# emerge net-dns/dnsmasq
```

DNSMasq is a powerful daemon that has the capability of functioning as a DNS cacheing server, DHCP Server, TFTP Server, and more. For now we will be focusing on one thing in the configuration, the DHCP Server.

The DNSMasq configuration file is located at: /etc/dnsmasq.conf and it is a very large file however there are only 3 options we need for this to work, you can later enable DNS and custom dhcp mappings if needed. Those 3 configuration options are:

1. dhcp-boot=pxelinux.0 #Tells the filename to grab from the tftp server for booting This is provided by the syslinux package we will be configuring in the next step
2. dhcp-range=192.168.0.100,192.168.0.250,72h #customize this range to suite your network needs.
3. interface=eth0 #The interface that will be acting as a DHCP server. If you want the DHCP server to run on a different interface be sure to change this option

Configuring PXELinux (based on syslinux)

Install sys-boot/syslinux (package not on wiki - please add (/Adding_an_Ebuild_to_the_Wiki)):

```
# emerge sys-boot/syslinux
```

PXE booting only requires one file that is installed by syslinux, however you will probably want to use more later on. For now we will use the pxelinux.0 file as we mentioned earlier while setting up DNSMasq, as well as a basic menu using the menu.c32 and a graphical menu using the vesamenu.c32 .

```
# cd /usr/share/syslinux
# cp menu.c32 vesamenu.c32 pxelinux.0 /tftpboot
# cd /tftpboot
```

PXELinux can boot a different option for each device's MAC address on your network, or it can also boot a default for all nic's on the network if a MAC address config isn't found. I will be covering the default method as it works for most simple setups. If you prefer a different boot configuration for each MAC address on your NICs then you can google for "pxelinux.cfg MAC config" and find tons of documentation for doing so. To set up the default config, first create the following directory:

```
# mkdir /tftpboot/pxelinux.cfg
```

Inside this directory is where the "default" config as well as any other custom configurations by MAC will reside. Here is an example of a graphical menu used to boot System Rescue CD, the file should be located at /tftpboot/pxelinux.cfg/default :

```
# The default menu style - using vesa menu in this example
DEFAULT vesamenu.c32
# If you have a png image in the /tftproot directory you can specify it here like so:
Menu Background netboot-1.png
# Prompt user for selection
prompt 0

#Global label identifier
label System Rescue CD
    # Set this entry as the default selection
    menu default
    # Actual viewable label text
    MENU LABEL System Rescue CD
    # The timeout for the entry is a bit unclear, but 10000 is equivalent to 10 Seconds.
    TIMEOUT 10000
    TOTALTIMEOUT 10000
    # The kernel image to load. This entry would actually reside at /tftproot/srcd/isolinux/r
    kernel srcd/isolinux/rescue64
    # The initrd relative to tftproot directory and specifying the netboot server, protocol, a
    # In this example the http protocol is used on server 192.168.0.1. The file is sysrcd.dat
    # If you have your http server set up to host files at /var/www/localhost/htdocs then this
    append initrd=srcd/isolinux/initram.igz netboot=http://192.168.0.1/sysrcd.dat
```

Mounting the ISO Image and Hosting the Compressed File System

In the above configuration example I was using a mounted System Rescue CD image at /tftproot/srcd. The kernel and initrd are located inside the isolinux directory of the ISO, the compressed filesystem is located at the top level of the ISO (i.e. /tftproot/srcd/sysrcd.dat).

In order to replicate the exact settings I used in this config you may do the following:

```
# cd /tftproot
# mkdir srcd
# mount -o loop /path/to/systemrescuecd.iso srcd/
```

Be sure to replace the "/path/to/systemrescuecd.iso" with the actual path you downloaded the System Rescue CD to and the actual filename.

Now you need to be sure that 2 files reside on your HTTP or FTP server, whichever you prefer to use for the netboot process is fine, but the System Rescue CD Netboot process will do 3 things:

1. Load Kernel
2. Load Initrd
3. Request the compressed filesystem from the network

The files needed for the 3rd step are located in the srcd/ directory if you mounted it with the above command. System Rescue CD uses a .dat file for the compressed filesystem, and it is verified during boot with a md5sum using the .md5 file in the srcd/ directory. The filenames are sysrcd.dat and sysrcd.md5. They need to be hosted on your fileserver/http server that you specify for the netboot argument in the pxelinux.cfg/default file. If you have a basic Apache/Lighttpd server set up you can do the following:

```
# ln -s /tftproot/srcd/sysrcd.dat /var/www/localhost/htdocs/
# ln -s /tftproot/srcd/sysrcd.md5 /var/www/localhost/htdocs/
```

Starting the services and preparing for use

First we want to start the PXE server:

```
# /etc/init.d/in.tftpd start
```

And now DNSMasq:

```
# /etc/init.d/dnsmasq start
```

If you are using Apache ensure it is running (If you use Lighttpd or Nginx replace this step with the appropriate service)

```
# /etc/init.d/apache2 status
```

If the service is not running, you should start it:

```
# /etc/init.d/apache2 start
```

If all your configuration options are correct and you have your HTTP/FTP server running and hosting the files properly, your configuration should be done on the server side for hosting *System Rescue CD*!! Don't get carried away just yet, we still have to test things are working :D

Testing your first network boot

The first thing you want to do now is set up your client to boot from the network. This may vary on different machines / bios, common methods are:

- Pressing F12 at boot to select boot method
- Pressing F1, F10, or DEL at boot to enter BIOS Setup
- Consult your motherboard documentation for the appropriate method of selecting boot device if the above don't work

You will want to choose a method to boot from Network as the first boot device. It may also be called "Boot From Lan" "Network Boot" "PXE Boot" Once you have selected the appropriate method you may need to save the settings, proceed on to booting. If you chose the right method you should be seeing some text on your screen, such as: PXE Boot.. Obtaining DHCP.... If all is well you will be presented with your PXELinux Boot menu. If your client system is still booting from the hard drive, or you see a failure related to obtaining DHCP IP address, please verify your settings in the above section "Installing and Configuring DNSMasq for DHCP / PXE Booting"[2] (http://www.funtoo.org/index.php?title=Installing_and_Configuring_DNSMasq_for_DHCP/_PXE_Booting&action=submit#Installing_and_Configuring_DNSMasq_for_DHCP_.2F_PXE_Booting) -make sure that your interface is set correctly, and that you are offering a DHCP range on the same internal network range as the IP address your server has. If you have any error relating to unable to find PXE boot, please verify that you have the pxelinux.0 file in your /tftpboot and that your /etc/dnsmasq.conf has the ""dhcp-boot=pxelinux.0"" configuration option.. **note that the 0 is a zero and not an o.

Upon a successful PXE configuration you will be presented with the network boot menu, with the option to boot System Rescue CD. If you have the appropriate files in the correct locations and your http/ftp server is working properly, you should be able to select the System Rescue CD menu entry and successfully boot via network. Congratulations!!

Adding more operating systems / installations to your working PXE setup

I know that by example, a lot of people probably want to use something other than the System Rescue CD. The main things have been outlined above for most linux distributions. MS Windows, see PXE Network Windows Installation (http://www.funtoo.org/wiki/PXE_Network_Windows_Installation) is quiet a bit more difficult than any linux install. I will try to cover the most important steps to serving a Windows Installation from the network soon.

If you are wondering how you go about hosting a different Linux install other than the System Rescue CD, the main things to look at are the pxelinux.cfg/default file to edit the kernel and initrd lines. You also need to be sure that those files are accessible by the PXE loader, and if your initrd requires a compressed filesystem, be sure that you have a working ftp/http server hosting the compressed filesystem (Remember once boot process has been handed

over to your kernel that you are no longer accessing the network via tftp but instead by the core services provided by the initrd + drivers provided by your kernel) I will add that you may use the `fetch=tftp://` protocol in the kernel cmdline, however it doesn't seem to work as stable as using http/ftp method. Each distro is different you may need to consult the documentation for the specific distro's needed boot cmdline. For the most part you will find it to be very similar(i.e. kernel+initrd+compressed-filesystem) Ubuntu doesn't even use a compressed filesystem on their ISO's it basically just uses a kernel and an initrd. I am currently working on getting a Funtoo netboot image developed, tested, and providing information on how to host a Funtoo Base system over the network via your Funtoo PXE Netboot Server.

Retrieved from "http://www.funtoo.org/index.php?title=PXE_network_boot_server&oldid=2031
(http://www.funtoo.org/index.php?title=PXE_network_boot_server&oldid=2031)"
Category (/Special:Categories): HOWTO (/Category:HOWTO)

This page was last modified on February 19, 2014, at 23:01.