

Pxe boot menu

Boot all your tools from the network

Pablo Escobar López
pescobar@cipf.es

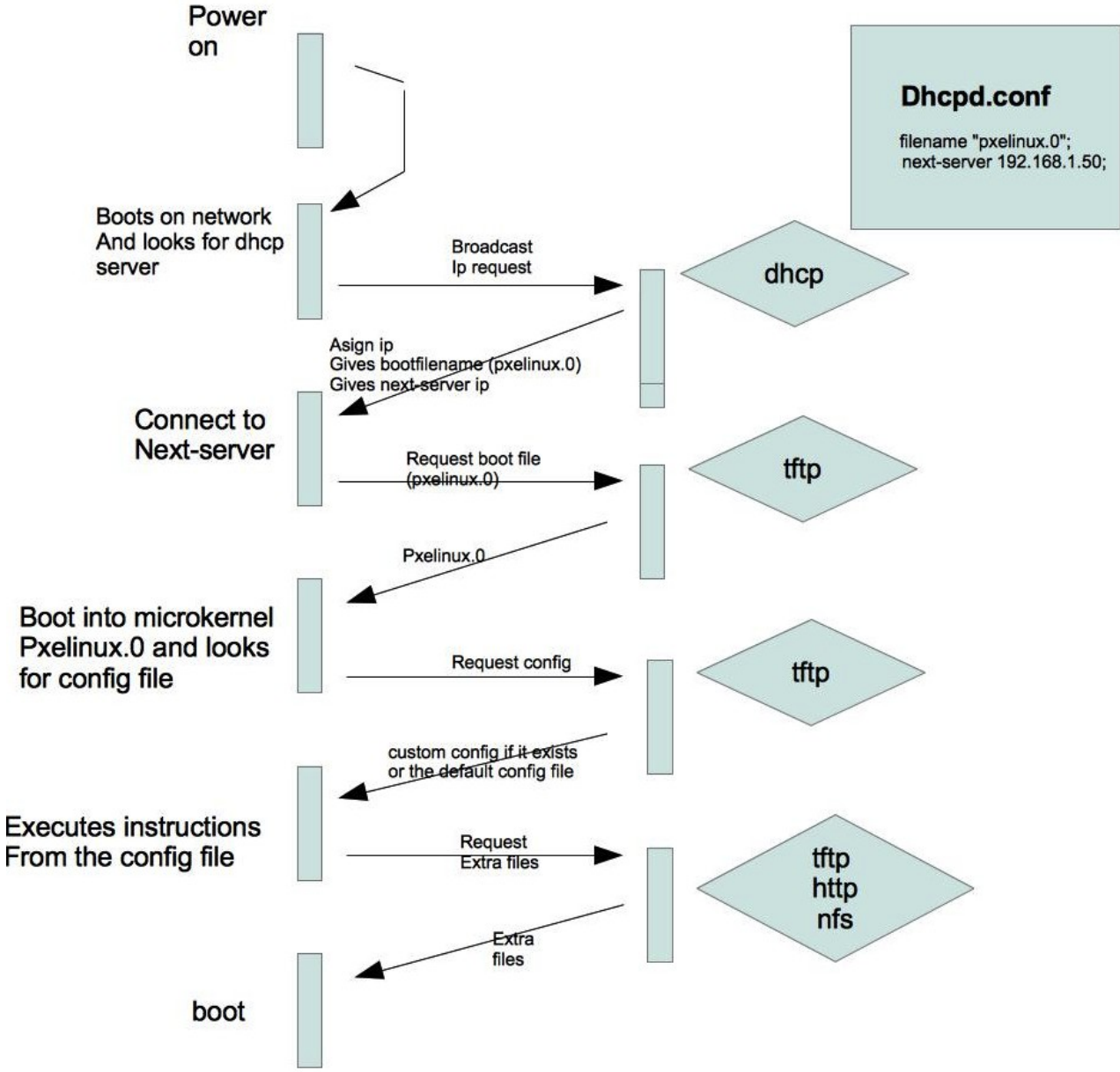
Why use pxe boot menu?

- Centralized soft repository for all your nodes
 - Boot linux installation
 - Boot recovery tools
 - Boot clone tools
- Forget about burning cd or usb
- Forget about looking for lost cd/usb
- No problem if your node has no cd or usb

What you need...

- Syslinux package (all needed files and docs)
- pxe boot support in your network card
 - Enable it in bios
- dhcp server with small config changes
 - dhcpd
 - dnsmasq
- tftp server
 - atftpd
 - dnsmasq
- Optional but useful:
 - http server
 - nfs server
 - dhcp relay

How pxe boot works



pxe config file search order

- You can have a custom pxe config file for each node
- The client looks for the config file in the folder “pxelinux.cfg” in the tftpd documentroot
- Order:

```
My IP address seems to be C0A80146 192.168.1.70
TFTP prefix:
Trying to load: pxelinux.cfg/01-00-14-22-a1-53-85
Trying to load: pxelinux.cfg/C0A80146
Trying to load: pxelinux.cfg/C0A8014
Trying to load: pxelinux.cfg/C0A801
Trying to load: pxelinux.cfg/C0A80
Trying to load: pxelinux.cfg/C0A8
Trying to load: pxelinux.cfg/C0A
Trying to load: pxelinux.cfg/C0
Trying to load: pxelinux.cfg/C
Trying to load: pxelinux.cfg/default
```

- We will use “default” for the boot menu

Network config

- If you have multiple networks/vlans and just one dhcp server you need to forward the nodes dhcp broadcasts.
 - Most network devices support it
 - Ex. cisco devices: “dhcprelay server 192.168.1.11”
 - You have available daemons in linux
 - Ex. “isc-dhcp-relay” package in debian

Sample configs: dhcp

- **dhcpcd.conf**

```
option domain-name-servers 62.31.64.39, 62.31.112.39;  
default-lease-time 86400;  
max-lease-time 604800;  
authoritative;
```

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.70 192.168.1.100;  
    filename "pxelinux.0";      <<<< boot filename  
    next-server 192.168.1.50;   <<<< tftp server ip  
    option subnet-mask 255.255.255.0;  
    option broadcast-address 192.168.1.255;  
    option routers 192.168.1.1;  
}
```

Sample configs: tftpd

- tftpd: just point the documentroot to the folder where you store all your files

```
[root][chuck-new][var/tftpboot/pxelinux.cfg]# ls -la /var/tftpboot
total 292
drwxr-xr-x  9 root root  4096 Oct  5 12:54 .
drwxr-xr-x 19 root root  4096 Sep  9 12:12 ..
drwxr-xr-x  2 root root  4096 Aug 16 12:39 clonezilla
drwxr-xr-x  2 root root  4096 Aug 16 12:54 gparted
drwxr-xr-x  3 root root  4096 Aug 16 12:33 lucid-installer
-rw-r--r--  1 root root 164504 Aug 16 12:43 memtest420
-rw-r--r--  1 root root  56164 Aug 16 13:32 menu.c32
drwxr-xr-x  3 root root  4096 Aug 16 15:30 natty-installer
-rw-r--r--  1 root root  26442 Aug 16 13:32 pxelinux.0
drwxr-xr-x  2 root root  4096 Oct 11 18:38 pxelinux.cfg
drwxr-xr-x  3 root root  4096 Aug  8 17:51 squeeze-installer
drwxr-xr-x  2 root root  4096 Sep  9 12:08 systemrescuecd
```

- Create folder “pxelinux.cfg” in the tftpd documentroot to store all config files

```
[root][chuck-new][var/tftpboot/pxelinux.cfg]# ls -la /var/tftpboot/pxelinux.cfg/
total 24
drwxr-xr-x  2 root root  4096 Oct 11 18:38 .
drwxr-xr-x  9 root root  4096 Oct  5 12:54 ..
-rw-r--r--  1 root root  1954 Oct  5 12:59 default
-rw-r--r--  1 root root  1741 Aug 16 15:33 lucidamd64.menu
-rw-r--r--  1 root root  1741 Aug 16 15:33 nattyamd64.menu
-rw-r--r--  1 root root  1345 Aug  8 17:51 squeezeamd64.menu
```


Sample configs: pxe config file

```
Label    squeeze-installer-amd64
kernel   squeeze-installer/amd64/linux
append   initrd=squeeze-installer/amd64/initrd.gz
```

- Label: menu entry name
- Kernel: kernel file to server over tftp
- Append: extra args to the kernel **or different submenu**
- “man syslinux” or /usr/share/doc/syslinux for more detailed documentation

Sample configs: pxe main menu pxelinux.cfg/default

```
[root][chuck-new][~]# head -30 /var/tftpboot/pxelinux.cfg/default
```

```
default menu.c32          << select "menu" interface (this file is in syslinux)  
prompt 0
```

```
menu title PXE Special Boot Menu  
MENU AUTOBOOT Starting Local System in 10 seconds
```

```
label bootlocal  
  menu label ^Boot Local HD  
  menu default          << this is default option if no user interaction  
  localboot 0          << boot local disk  
  Timeout 100          << timeout if no user interaction (1/10s)  
  TOTALTIMEOUT 3000_ << global timeout
```

```
LABEL Debian 6 - squeeze amd64  
  MENU LABEL ^Debian 6 squeeze installer - amd64  
  KERNEL menu.c32  
  APPEND pxelinux.cfg/squeezeamd64.menu << go to submenu
```

```
LABEL sysrescuecd - 32 bits  
  MENU LABEL ^Sysrescuecd 2.3.1 - 32 bits  
  kernel systemrescuecd/rescuecd  
  append initrd=systemrescuecd/initram.igz dhdhcp netboot=http://1.1.1.1/systemrescuecd/sysrcd.dat
```

submenu

pxelinux.cfg/squeezeamd64.menu

menu title squeeze x86_64

DEFAULT install

LABEL Main Menu

MENU LABEL ^Return to Main Menu

KERNEL menu.c32

APPEND pxelinux.cfg/default << go back to main menu

LABEL install

KERNEL squeeze-installer/amd64/linux

APPEND vga=normal initrd=squeeze-installer/amd64/initrd.gz --

LABEL linux

KERNEL squeeze-installer/amd64/linux

APPEND vga=normal initrd=squeeze-installer/amd64/initrd.gz --

What we get

Main menu

```

                                PXE Special Boot Menu
-----
Boot Local HD
Debian 6 squeeze installer - amd64
Ubuntu 10.04 Lucid installer - amd64
Ubuntu 11.04 Natty installer - amd64
Sysrescuecd 2.3.1 - 32 bits
Sysrescuecd 2.3.1 - 64 bits
clonezilla-ubuntu-21-07-2011
GParted Live - 0.9.0-7 - 28/07/2011
memtest86-3.2
memtest86-3.2-password

                                Press [Tab] to edit options

```

sub-menu

```

                                squeeze x86_64
-----
Return to Main Menu
install
linux
expert
rescue
auto

```

Boot ubuntu live cd

- Mount ubuntu live iso on nfs server
 - \$> mount -o loop ubuntu-10.04-desktop-i386.iso /mnt/iso
- Create nfs share & copy all the content of live cd to it
 - \$> mkdir /nfs-share/ubuntu
 - \$> cp -av /mnt/iso/* /nfs-share/ubuntu/
- Entries in nfs share export file
 - \$> cat /etc/exports
 - /nfs-share/ubuntu/ *(**no_root_squash**,rw,async)
- Copy initrd and vmlinuz from Ubuntu live cd to tftp documentroot
 - \$> cd /nfs-share/ubuntu/casper
 - \$> cp vmlinuz initrd.lz /tftpboot/images/ubuntu/

- pxe config file

LABEL Ubuntu

kernel /tftpboot/images/ubuntu/vmlinuz

append boot=casper netboot=nfs nfsroot=\$NFSSERVER:/nfs-share/ubuntu \
initrd=/tftpboot/images/ubuntu/initrd.lz

Redhat kickstart install

- You should already have a working kickstart server. You should have the following directory structure:

```
/var/www/pub
|-- CentOS
|-- images
    |-- pxeboot
|-- isolinux
    |-- isolinux.cfg
|-- kickstart
|-- repodata
```

- Then use this pxe config file:

```
label RHEL5
kernel RHEL5/vmlinuz
append initrd=RHEL5/initrd.img ks=nfs:192.168.0.50:/export/kickstart/ks/default.ks
```

Source: <http://secnut.blogspot.com/2010/04/pxe-kickstart-automating-centosredhat.html>

Some extra details....

- Debug with “tail -f tftpd.log” when booting clients
 - “tail -f /var/log/daemon.log” when using dnsmasq as tftpserver in debian
- tftp is slow for big files. Better http
- Password protect your menu entries

```
LABEL Boot Linux system! (32-bit kernel)
MENU PASSWD pxe123
KERNEL rip/kernel
APPEND vga=normal initrd=rip/rootfs.cgz root=/dev/ram0 rw
```
- gpxelinux.0
- memdisk

Some links....

- My working pxe menu configs (4kb)
 - <http://91.121.137.186/pxebootmenu-configs.tar.gz>
 - **Remember to edit “default” config file with your server ips!!**
- All files in my tftpd documentRoot (550M)
 - <http://91.121.137.186/pxebootmenu-full.tar.gz>
- Booting iso file
 - <https://projects.centos.org/trac/livecd/wiki/PxeBoot>
- Build your custom live-cd
 - <http://www.ubuntu-mini-remix.org/>
 - <http://sourceforge.net/projects/u-customizer/>

Questions ?