Creating a PXE Boot Environment with TinyCore

Presented by Andy Carlson

Objective

- Gain a general understanding of PXE Boot functionality
- Learn how to configure dnsmasq to provide necessary services
- Learn how to create custom remixes of TinyCore Linux for use in a PXE environment

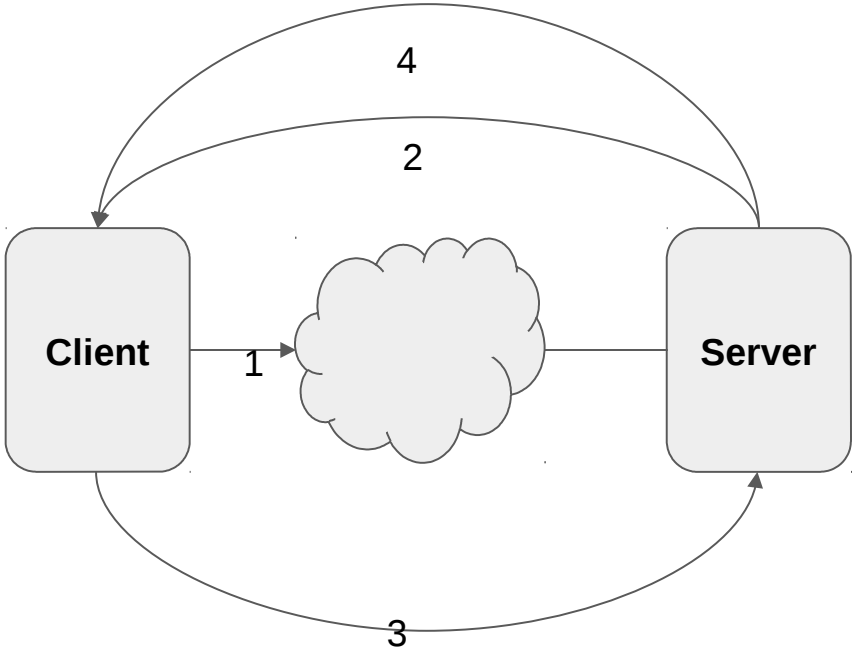
What is PXE Boot?

Preboot e**X**ecution **E**nvironment

Definition:

Provides functionality to provision an in-memory boot environment on client machines using DHCP and TFTP

How DHCP Works



1. Broadcast request for available DHCP servers
2. Response from available DHCP server
3. Request for IP address from DHCP server
4. Response with available IP Address

DHCP Options

Packet Structure

IP Headers
Host Config Data (IP address)
DHCP Options

Options include:

- Lease information
- Router(s)
- DNS
- Time
- TFTP server

Trivial File Transfer Protocol (TFTP)

- No Authentication/Authorization
- No directory browsing capabilities
- Get/Put file using whole path

} Low operational overhead
Simple to implement

PXE Booting Linux

- Request DHCP Lease
- Receive IP Lease w/ boot configuration
  - Bootloader file location (full path on server)
  - TFTP server address
  - Bootloader configuration (path relative to TFTP root)
  - Working Directory (relative to TFTP root)
- Load necessary bootfiles via TFTP
  - pxelinux.0 – PXE Firmware
  - vmlinuz – kernel
  - initrd – init ramdisk

Preparing the Boot Environment

- Step 1: Configure DNSMasq
  - DHCP server (with options)
    - dhcp-boot=<pxelinux.0-full-path>[,<TFTP-address>]
    - dhcp-option-force
      - 209 – pxelinux configuration file (isolinux.cfg)
      - 210 – pxelinux path prefix
      - 66 – TFTP server address
  - TFTP
    - enable-tftp
    - tftp-root=<full-path-tftp root>

- T

Preparing the Boot Environment

- Step 2: Obtain pxelinux.0 firmware file
  - Contained within the syslinux project
  - Shortcut: Download syslinux-4.04.tar.gz (contains pre-built binary)

- T

Preparing the Boot Environment

- Step 3: Obtain content of /boot folder from TinyCore ISO
  - Mount ISO image as loop device (mount -o loop)
  - Copy the content including:
    - core.gz – init ramdisk
    - vmlinux – kernel
    - isolinux folder – boot menu

- T

Preparing the Boot Environment

- Step 4: Configure init ramdisk to boot as desired
  - Decompress and extract core.gz (using inity.sh)
  - Download necessary TCZ packages (using getcz.sh)
  - Modify startup commands (located in /etc/profile)
    - Install TCZ packages on boot
    - Start xwindows (if necessary)
  - Add files and compress new core.gz file

- T

Preparing the Boot Environment

- Step 5: Add new configuration to isolinux.cfg
  - LABEL <environment-short-name>
  - MENU <environment-descriptive-name>
  - KERNEL <relative-TFTP-path>
  - INITRD <relative-TFTP-path>

- T

Ready to Boot

- Remember to configure BIOS to allow PXE/Network boot
- All in memory

- T

Something else…..booting raw disk images

- Copy disk image to TFTP server
- Copy memdisk from syslinux-4.04.tar.gz to boot folder
- Modify isolinux.cfg
  - KERNEL /boot/memdisk
  - INITRD <DISK-IMAGE>

Demo Time!!

Questions!!!