Thema: Netzwerkboot mit PXE

24.07.2014, 20:52 **#1**

Gamienator
Lt. Junior Grade

Dabei seit: Apr 2011
Ort: Oberursel
Beiträge: 274

[Anleitung] Netzwerkboot mit PXE

HowTo zum einrichten einer Bootumgebung in Debian mit Beispielen zum Einfügen verschiedener Images

Diskussion befindet sich genau hier

Inhaltsverzeichnis:

Einleitung

Mit diesem HowTo möchte ich ein Thema anschneiden, dass mir sehr viel Zeit und auch Nerven gekostet hat. Dieses System hat bestimmt jeder von euch schon einmal gesehen, sei es in der Schule oder auf der Arbeit. Neben den gängigen Methoden wie DVDs oder USB Sticks kann ein Betriebssystem auch über das Netzwerk bereitgestellt werden. In der Theorie mag dies einfach sein, jedoch habe ich mehrere Jahre gebraucht um ein funktionierendes Bootsystem zu erstellen, ob es jetzt an mangelnden deutschen Anleitungen oder an der fehlenden Lebenserfahrung lag möchte ich jetzt mal nicht thematisieren. Das Endergebnis ist für mich aber einfach großartig: Ich brauche mir keine Sorgen mehr zu machen wegen DVD Images, USB Sticks usw. Will ich ein Betriebssystem starten/installieren überlasse ich meinem Netzwerk die Arbeit. Und wie das möglich ist zeige ich euch in diesem HowTo.

Jeder User ist willkommen hier beizutragen oder Fehler zu korrigieren! Dafür gibt es ja den Diskussionsthread. Sollte ich Begriffe wie Bootloader etc. falsch in Verbindung gebracht haben sagt es ruhig, aber bitte in einem angemessenen Ton! Ich selber nutze auch nicht Linux täglich und da kann es mal passieren, dass man einen Begriff durcheinander wirft.

Aufbau

Der Aufbau ist wie folgt: Als erstes möchte ich über die Grundvoraussetzungen reden. Anschließend zeige ich euch, wie ihr in Debian alles benötigte einrichtet um euer Netzwerk die Systeme anbieten zu können inkl. kleiner Erläuterungen für die Konfigurationsdateien. Als letztes gebe ich euch dann Beispiele, wie ihr folgende Systeme einbindet:

1. Lokaler Festplattenstart
2. memtest 86+
3. Debian 7 Netzwerkinstallation
4. Linux Mint Live DVD
5. GParted
6. Windows Installationsumgebung

Kleine Linux Kenntnisse sind hier von Vorteil! (Zum Beispiel der Aufbau des Speichersystems, das es nicht wie in Windows unter verschiedenen Buchstaben abgespeichert wird). Dieses HowTo soll aber auch dazu dienen, dass jeder eine Bootumgebung einrichten kann, deshalb versuche ich jeden Befehl zu erläutern.

Wenn ich sage, bearbeite Datei /etc/blablabla.conf ist es euch überlassen, mit welchem Editor es macht. Erfahrene Linuxer nehmen vi oder nano. Eine Änderung der Textdateien unter Windows ist z. B. über das kostenlose WinSCP möglich. WinSCP eignet sich hervorragend, da auch Dateien einfach übertragen werden können. Und für Linux Neulige aus der Windows Welt gibt es auch noch den Hinweis: Im Gegensatz zu Windows ist es bei Linux nicht nötig bei jeder Datei eine Endung wie z.B. .txt zu haben. Das bedeutet z. B. im Ordner /tftpboot/pxelinux.cfg liegt die Datei default

Dieses HowTo ist aber nicht gedacht, einfach blind jeden Befehl abzutippen! Ich versuche jeden Schritt zu

Grundlegendes

Für diese Technik wird das Protokoll PXE (Preboot Execution Environment) verwendet und benötigt zwei Dinge:

1. Einen DHCP Server
2. Einen TFTP Server

Dabei ist der Ablauf recht simpel. Beim Hochfahren fragt die Netzwerkkarte, ob es einen PXE-Boot Server gibt und wo der zu finden ist. Wenn der DHCP Server die Anfrage beantwortet, lädt die Netzwerkkarte den Bootloader vom TFTP Server in den Arbeitsspeicher. Alles Weitere wird dann vom Bootloader erledigt. Dieses kann über verschiedene Wege realisiert werden, auch Windows DHCP Server sind in der Lage die Anfragen zu beantworten.

Bei TFTP handelt es sich um ein vereinfachtes FTP Protokoll. Es wird nur der Transfer von Daten unterstützt, spezielle Rechtevergabe wie chmod etc. ist bei TFTP nicht vorgesehen.

Ist beides vorhanden, steht alles bereit. Ob es nun auf Windows, Linux oder sogar auf einem Router mit veränderter Firmware liegt ist dabei egal. (Ich nutze meinen Asus RT-N56U mit Custom Firmware, da die DHCP Config angepasst werden kann)

Realisierung in diesem Beispiel

In diesem HowTo erkläre ich euch, wie ihr unter Debian alles einrichtet. Ob dies dabei unter einer VM in VirtualBox / VMWare oder auf richtiger Hardware geschieht, ist dabei euch überlassen! Wichtig ist nur zu wissen, dass der Server vorhanden sein muss, wenn ihr per Netzwerk booten wollt. Also bringt es nichts die VM auf dem Rechner zu haben, auf dem ihr gleichzeitig das Betriebssystem installieren wollt. Wenn ihr euch extra Debian installiert, ist eine grafische Oberfläche nicht notwendig!

Als IP-Adresse verwenden wir ein klassisches privates Netzwerk mit den Adressen 192.168.0.0/24 (Also sind die Adressen 192.168.0.1 – 192.168.0.254 benutzbar).

Die Daten des TFTP Server werden in /tftpboot gespeichert sein

---

Installieren aller nötigen Pakete

Als erstes wollen wir alle nötigen Pakete installieren. Dieses geschieht ganz einfach mit dem Befehl

Code:

```
apt-get update
apt-get install dnsmasq syslinux
```

apt-get ist der Paketverwalter in Linux. Dieser muss immer mit root rechten ausgeführt werden! apt-get update dient dazu, eure Paketlisten zu aktualisieren, mit apt-get install installiert ihr nun die Pakete.

dnsmasq ist ein mächtiges Tool und stellt alles bereit was wir brauchen! DHCP und TFTP Server in einem. Somit ist bei der richtigen Konfiguration alles möglich.

syslinux ist ein Bootloader. Aus diesem Paket werden einige Dateien benötigt, um ein passendes Bootmenü erstellen zu können.

Einrichten von dnsmasq

Beim Einrichten von dnsmasq gibt es zwei Möglichkeiten: Es gibt bereits einen DHCP Server im Netzwerk (z. B. im Router) oder nicht. Da sich zwei DHCP Server im selben Netzwerk gar nicht leiden können, müssen wir dnsmasq anders konfigurieren, ich zeige euch nun die zwei Möglichkeiten:

dnsmasq ohne vorhandenem DHCP Server im Netzwerk

Die Konfigurationsdatei ist /etc/dnsmasq.conf und sollte ungefähr so aussehen:

Code:

```
# Disable DNS
port=0
enable-tftp
tftp-root=/tftpboot
dhcp-range=192.168.0.100,192.168.0.254,12h
dhcp-boot=/pxelinux.0,0.0.0.0
pxe-prompt="F8 druecken fuer Bootoptionen", 3
pxe-service=X86PC, "Netzwerkboot", /pxelinux
```

Code:

```
# Disable DNS
port=0
```

sorgt dafür, dass der DNS Dienst von dnsmasq abgeschaltet wird. Solltet ihr keinen DNS Server im Netzwerk haben, kann dieser natürlich aktiviert bleiben!

Code:

aktiviert den TFTP Server

Code:
```
1. tftp-root=/tftpboot
```

gibt den Speicherort des TFTP Servers an. Dieser kann beliebig gewählt werden, in unserem Beispiel ist es /tftpboot

Code:
```
1. dhcp-range=192.168.0.100,192.168.0.254,12h
```

gibt an, welcher IP-Bereich vergeben werden soll! Mit diesen Einstellungen erhalten Clients eine IP im Bereich 192.168.0.100 und 192.168.0.254. Die 12h geben dabei die Leasetime an.

Code:
```
1. dhcp-boot=/pxelinux.0,0.0.0.0
```

gibt an, welche Datei gebootet werden soll und wo der TFTP Server liegt. Bei der Angabe 0.0.0.0 nimmt dnsmasq an, dass der TFTP Server auf demselben Rechner liegt, wo dnsmasq ausgeführt wird.

Code:
```
1. pxe-prompt="F8 druecken fuer Bootoptionen", 3
2. pxe-service=X86PC, "Netzwerkboot", /pxelinux
```

An diesem Punkt kann man bereits Bootoptionen einbinden wie Boote, Windows, Linux etc. Dies habe ich hier aber bewusst ausgelassen, da ich im Bootmenu alles regeln möchte. Die ,3 in der ersten Zeile gibt die Wartezeit in Sekunden an, bis die erste Option automatisch ausgewählt wird.

dnsmasq mit bereits vorhandenem DHCP Server

Wenn ein DHCP Server bereits vorhanden ist, muss die Zeile dhcp-range wie folgt aussehen:

Code:
```
1. dhcp-range=192.168.0.0,proxy
```

Der Rest kann aus der oberen Konfiguration entnommen werden! Mit dem Zusatz ,proxy wird dnsmasq zu einem ProxyDHCP. Während der erste DHCP Server Informationen wie IP-Adresse, Gateway etc. vergibt, reagiert dnsmasq nur auf die Anfrage über einen vorhandenen TFTP Server und dessen Bootdatei.

Debian eine fest IP-Adresse zuweisen

Auch wenn ein DHCP Server vorhanden ist, ist es sinnvoll der Debianmaschine eine feste IP-Adresse zuzuweisen! Am einfachsten ist es, wenn ihr direkt eure Konfigurationsdatei ändert. Das ist die /etc/network/interfaces und sieht normalerweise so aus:

Code:
```
1. # The loopback interface
2. auto lo
3. iface lo inet loopback
4.
5. # The first network card - this entry was created during the Debian installation
6. auto eth0
7. iface eth0 inet dhcp
```

Die Zeilen mit dem interface lo interessieren uns nicht. Wichtig sind die Zeilen mit eth0. (Wer mehrere Netzwerkkarten hat, wird dementsprechend auch noch eth1, eth2 etc. finden. Passt dies natürlich eurer Umgebung an!) Diese werden dann wie folgt abgeändert:

Code:
```
1. iface eth0 inet static
2. address 192.168.0.10
3. network 192.168.0.0
4. broadcast 192.168.0.255
5. netmask 255.255.255.0
6. gateway 192.168.0.1
7. dns-nameservers 192.168.0.1
```

Code:
```
1. address 192.168.0.10
```

ist dann die von euch festgelegte IP-Adresse des Debian. Die Einträge

Code:
```
1. gateway
```

&

Code:
```
1. dns-nameservers
```

Erstellen und kopieren der Bootdateien für TFTP

Jetzt gilt es die Ordnerstruktur und die passenden Dateien zu kopieren. Auch das folgende sollte über root erfolgen:

Code:
```
1. mkdir -p /tftpboot/boot tftpboot/pxelinux.cfg
2. chmod 777 /tftpboot
3. cp -p /usr/lib/syslinux/pxelinux.0 /tftpboot
4. cp -p /usr/lib/syslinux/menu.c32   /tftpboot
5. cp -p /usr/lib/syslinux/memdisk    /tftpboot/boot
```

Die Befehle machen nun folgendes:

Code:
```
1. mkdir -p /tftpboot/boot tftpboot/pxelinux.cfg
```

erstellt die benötigten Ordner wie in den Angaben der dnsmasq Konfiguration

Code:
```
1. chmod -R 777 /tftpboot
```

sort dafür, dass jeder auf den Ordner zugreifen kann

Code:
```
1. cp -p /usr/lib/syslinux/pxelinux.0 /tftpboot
2. cp -p /usr/lib/syslinux/menu.c32   /tftpboot
3. cp -p /usr/lib/syslinux/memdisk    /tftpboot/boot
```

cp ist ein einfacher Kopierbefehl. Mit diesen drei Zeilen habt ihr nun alle nötigen Dateien kopiert.

Erstellen der Konfigurationsdatei zum Booten

Im Ordner /tftpboot/pxelinux.cfg werden die Konfigurationsdateien für den Bootloader abgelegt, dabei kann für jede MAC-Adresse oder IP-Adresse eine eigene Konfiguration angelegt werden! Sollte keine spezielle Konfiguration gefunden werden, wird die datei default verwendet. Ich möchte euch den Bootvorgang mal in einem Bild zeigen:

Code:
```
1. /tftpboot/pxelinux.cfg/01-88-99-aa-bb-cc-dd
2. /tftpboot/pxelinux.cfg/C0A80066
3. /tftpboot/pxelinux.cfg/C0A8006
4. /tftpboot/pxelinux.cfg/C0A800
5. /tftpboot/pxelinux.cfg/C0A80
6. /tftpboot/pxelinux.cfg/C0A8
7. /tftpboot/pxelinux.cfg/C0A
8. /tftpboot/pxelinux.cfg/C0
9. /tftpboot/pxelinux.cfg/C
10. /tftpboot/pxelinux.cfg/default
```

Der Rechner hatte die MAC-Adresse 88:99:aa:bb:cc:dd und bekam die IP-Adresse 192.168.0.102 Als erstes wurde nach einer speziellen MAC-Adressen Einstellung gesucht (dabei wird immer 01- davor geschrieben) Bei Misserfolg sucht er dann nach einer IP-Adressen Einstellung, dabei wird die IP-Adresse in Hexdezimal umgerechnet! Ist es wieder in Misserfolg, zieht er die letzte stelle ab usw. Am Ende lädt er dann die Datei default.

Also erstellen wir uns nun die Datei. Der komplette Pfad ist also /tftpboot/pxelinux.cfg/default und das Grundgerüst der Datei sieht dann so aus:

Code:
```
1. DEFAULT menu.c32
2. PROMPT 0
3. MENU TITLE CSG PXE Boot Menu
4. TIMEOUT 100
```

Code:
```
1. DEFAULT menu.c32
```

sagt aus, dass die Datei menu.c32 geladen werden soll. Diese hatten wir aus dem SYSLINUX Paket kopiert.

Code:
```
1. MENU TITLE stellt
```

die Überschrift ein.

Code:
```
1. TIMEOUT
```

sagt aus, nach wie viel Zeit die erste Zeile ausgewählt werden soll (es kann auch eine andere Zeiel als default markiert werden, aber da sehe ich kaum Sinn dahinter). Bei einem Timeout von 100 von nach 10 Sekunden die

alles in der Datei /tftpboot/pxelinux.cfg/default geregelt. Wie ihr einzelne Systeme nun anbindet zeige ich euch.

Am ende möchte ich euch nochmal zeigen, wie eure Ordner dann aussehen sollten:

Lokale Festplatte booten

Was im ersten Moment unsinnig erscheint kann am Ende doch hilfreich sein. Um nämlich nicht jedesmal die Bootreihenfolge ändern zu müssen, kann man doch sagen, das bei Inaktivität ganz normal die erste Festplatte starten soll. Dies ist auch das einfachste Image Das einzige was ihr dazu machen müsst, ist eure /tftpd /pxelinux.cfg/default um folgende Zeilen zu erweitern:

Code:
```
1. LABEL localhdd1
2. MENU LABEL Lokaler Boot
3. LOCALBOOT  0
```

Code:
```
1. LABEL localhdd1
```

ist der interne Label für menu.c32 und sollte einmalig sowie einfach sein! Umlaute, Sonderzeichen etc. wird gar nicht gern gesehen!

Code:
```
1. MENU LABEL Lokaler Boot
```

stellt ein, was auf dem Bildschirm zu sehen ist.

Code:
```
1. LOCALBOOT  0
```

ist der code, der bei Auswahl ausgeführt wird.

Das Einbinden weiterer Systeme wird in den nächsten Tagen hoffentlich online sein.

Geändert von Gamienator (24.07.2014 um 21:15 Uhr)

Besucht mal meinen Blog unter www.n3rds.xyz Ich schreib dort ab und zu über Ärgernisse in der IT Welt

Zitieren

+ Antworten

Facebook Twitter Google+

« Vorheriges Thema | Forum: Anleitungen und FAQs unserer Leser | Nächstes Thema »

Berechtigungen

Neue Themen erstellen: **Nein**
Themen beantworten: **Nein**
Anhänge hochladen: **Nein**
Beiträge bearbeiten: **Nein**

BB-Code ist **an**.
Smileys sind **an**.
[IMG] Code ist **an**.
[VIDEO] Code ist **aus**.
HTML-Code ist **aus**.

Foren-Regeln

Gehe zu: **Anleitungen und FAQs unserer Leser** **Nach oben**

Forum-Layout: **Feste Breite** / Flexible Breite

Alle Zeitangaben in WEZ +1. Es ist jetzt 02:05 Uhr.