

Diesen Beitrag schrieb ich vor 4 Jahren. Behalte das beim Lesen bitte im Hinterkopf.



Heute mal etwas ganz anderes: nämlich einen Gastbeitrag von iRainer, den ihr auf Twitter unter [@stixihaftigkeit](#) erreichen könnt. Er sorgt dafür, dass hier mal etwas eher Ungewöhnliches erscheint: ein Artikel zum Thema Windows :D Ich wünsche euch viel Vergnügen beim Nachmachen...

Wir wollen einen Windows-Thin-Client bauen, der über keine lokale Festplatte verfügt, sondern sich sein Dateisystem via lokalem Netzwerk von einem Fileserver lädt. Warum? Weils geht ;) Ernsthaft: ich möchte einen möglichst simplen, dummen lokalen PC verwenden, der eben nur als Interface für Tastatur/Maus/Monitor/sonstige Peripherie zuständig ist und im Falle eines Defekts möglichst simpel durch ein baugleiches neues Modell ersetzt werden kann, ohne dass man Windows komplett neu installieren muss o.ä. Daher nutzen wir das booten via Netzwerk/PXE, um uns von einem Server mit dem Betriebssystem zu versorgen. Weiterer Vorteil: der Server kann dann ein hübsches RAID besitzen und zentral Backup fahren.

Zu dem Thema gibt es einige Anleitungen im Netz, hier einmal eine Auswahl die mir nützlich war:

- <http://it-joe.com/howtos/winiscsi.php>
- <http://it-joe.com/howtos/pxe.php>
- <https://rom-o-matic.eu/>
- <http://ipxe.org/>
- <http://www.syslinux.org/>
- <https://www.debian.org/releases/wheezy/i386/ch04s05.html.de>
- <http://prefetch.net/articles/iscdhcpd.html>

Als Serversystem verwenden wir das aktuelle Debian8, für unsere Windows-Clients kommen Windows 7 SP1 und Windows 8.1 zum Einsatz. Folgende Komponenten brauchen wir:

1. einen iSCSI-Server
2. das Windows Preinstallation Environment
3. einen PXE-Server
4. einen DHCP-Server
5. einen TFTP-Server
6. einen Samba-Server
7. Windows7/8-Installationsmedien

1.) Der iSCSI Target Server

Zunächst brauchen wir ein Festplattenabbild für jede Windowsversion. Unser Arbeitsverzeichnis wird im Folgenden /tftpboot sein. Erzeugen wir also je ein 40GB grosses Image für Win7 bzw. Win8:

```
$ mkdir -p /tftpboot/iscsi
$ dd if=/dev/zero of=/tftpboot/iscsi/windows7_master.img bs=1M count=40000
$ dd if=/dev/zero of=/tftpboot/iscsi/windows8_master.img bs=1M count=40000
```

Nun wird der iSCSI-Server installiert und konfiguriert:

```
$ apt-get install iscsitarget
```

```
$ vim /etc/default/iscsitarget
ISCSITARGET_ENABLE=true
```

Jetzt geben wir die oben angelegten Image-Dateien per iSCSI frei.

```
$ vim /etc/iet/ietd.conf
Target iqn.2015-04.me.spiller:windows7_master
# Benutzername/Passwort sparen wir uns hier, um das System erstmal nicht
unnötig zu verkomplizieren
IncomingUser
OutgoingUser
Lun 0 Path=/tftpboot/iscsi/windows7_master.img,Type=fileio
Alias Windows7
```

```
Target iqn.2015-04.me.spiller:windows8_master
IncomingUser
OutgoingUser
Lun 0 Path=/tftpboot/iscsi/windows8_master.img,Type=fileio
Alias Windows8
```

```
$ sudo /etc/init.d/iscsitarget start
```

2.) Windows Preinstallation Environment

Damit wir Windows zunächst auf der iSCSI-Freigabe installieren können, benötigen wir eine iSCSI-fähige Installationsumgebung. Diese erhalten wir mit einem Windows Preinstallation Environment (WinPE) Bootimage. Je nachdem, ob wir Windows 7 oder Windows 8 installieren wollen, müssen wir das WAIK oder WADK verwenden. Hierzu wird ein Win7 (oder höher) System benötigt, auf dem wir die Pakete installieren können.

Windows 7 Systeme

Wir benötigen folgende Downloads von Microsoft:

- <https://www.microsoft.com/de-de/download/details.aspx?id=5753>
- <https://www.microsoft.com/de-de/download/details.aspx?id=5188>

Um die Ergänzung zum Windows AIK für Windows PE 3.1 zu installieren sind folgende Schritte erforderlich:

- Brennen Sie das ISO-Abbild der Ergänzung zum Windows AIK auf eine DVD-ROM.
- Öffnen Sie eine Eingabeaufforderung mit erhöhten Rechten.
- Ersetzen Sie die installierten Windows PE 3.0-Dateien durch die entsprechenden Windows PE 3.1-Dateien aus dem ISO-Abbild. Führen Sie dazu den folgenden Befehl aus (dabei ist „E:“ das Stammverzeichnis des ISO-Abbilds):

```
xcopy E:\ "C:\Program Files\Windows AIK\Tools\PETools" /ERDY
```

Nun benötigen wir die Eingabeaufforderung für Bereitstellungstools: Start -> Programme -> Microsoft Windows AIK -> Eingabeaufforderung für Bereitstellungstools als Administrator ausführen. Für ein 64-bit-System müssen dann die folgenden Schritte ausgeführt werden:

```
C:\> copyype amd64 C:\WinPE_amd64
C:\> copy "C:\Program Files\Windows AIK\Tools\PETools\amd64\Winpe.wim"
C:\Winpe_amd64\ISO\Sources\Boot.wim
C:\> copy "C:\Program Files\Windows AIK\Tools\amd64\Imagex.exe"
C:\Winpe_amd64\ISO\
C:\> oscdimg -n -bC:\Winpe_amd64\etfsboot.com C:\Winpe_amd64\ISO
C:\Winpe_amd64\winpe3_amd64.iso
```

Windows 8 Systeme

- <https://www.microsoft.com/de-de/download/details.aspx?id=30652>

Hier führen wir Start -> Programme -> Windows Kits -> Windows ADK -> Umgebung für Bereitstellungs- und Imageerstellungstools als Administrator aus:

```
C:\> copyype amd64 C:\Winpe_amd64
C:\> MakeWinPEMedia /ISO C:\WinPE_amd64 C:\WinPE_amd64\winpe4_amd64.iso
```

Die so erzeugten ISO-Dateien kopieren wir auf unseren Debian-Server nach /tftpboot/distros/windows.

Das (i)PXE-Setup

PXE

Zunächst benötigen wir PXE-Funktionalität, damit die Netzwerkkarte booten kann. Damit es schick aussieht und man ggf. auch verschiedene Systeme booten kann, verwenden wir ein Menu.

```
$ apt-get install pxelinux syslinux-common
$ cd /tftpboot
$ cp /usr/lib/PXELINUX/pxelinux.0 .
$ cp /usr/lib/syslinux/modules/bios/menu.c32 .
$ cp /usr/lib/syslinux/modules/bios/libmenu.c32 .
$ cp /usr/lib/syslinux/memdisk .
$ cp /usr/lib/syslinux/modules/bios/ldlinux.c32 .
$ cp /usr/lib/syslinux/modules/bios/libutil.c32 .
$ mkdir pxelinux.cfg
```

```
## pxelinux.cfg/default
DEFAULT menu.c32
PROMPT 0
MENU TITLE PXE Boot
```

```
LABEL Windows 7 Install
MENU LABEL Windows 7 Install
KERNEL ipxe.lkrn
INITRD win7.ipxe
```

```
LABEL Windows 8 Install
MENU LABEL Windows 8 Install
KERNEL ipxe.lkrn
INITRD win8.ipxe
```

Die default-Config wird immer ausgewertet, wenn keine spezifischere Konfiguration im Verzeichnis aufzufinden ist. Um bspw. einem bestimmten Client eine spezielle Konfiguration zu verpassen, kann bspw. als Dateiname die mit `,-`, versehene und durch eine 01- vorangestellte MAC-Adresse verwendet werden, z.B. `pxelinux.cfg/01-00-aa-bb-cc-dd-ee`.

iPXE

iPXE brauchen wir, damit wir iSCSI-Funktionalität im Netzwerkbootloader haben. Wir verwenden der Einfachheit halber das entsprechende Debianpaket. Also:

```
$ apt-get install ipxe
$ cp /usr/lib/ipxe/ipxe.lkrn /tftpboot/
```

Anschliessend brauchen wir noch die iPXE-Startskripte die vom PXE-Menu aus aufgerufen werden.

Windows 7 iPXE Script

```
## /tftpboot/win7.ipxe
#!ipxe
dhcp
set keep-san 1
sanhook iscsi:192.168.0.50::::iqn.2015-04.me.spiller:windows7_master
set boot-url tftp://192.168.0.50
initrd ${boot-url}/distros/windows/winpe3_amd64.iso
kernel ${boot-url}/memdisk iso raw
boot
```

Windows 8 iPXE Script

```
## /tftpboot/win8.ipxe
#!ipxe
dhcp
set keep-san 1
sanhook iscsi:192.168.0.50::::iqn.2015-04.me.spiller:windows8_master
set boot-url tftp://192.168.0.50
initrd ${boot-url}/distros/windows/winpe4_amd64.iso
kernel ${boot-url}/memdisk iso raw
boot
```

DHCP-Server

Damit die Netzwerkkarte auch mit den nötigen Parametern versorgt wird, brauchen wir einen DHCP-Server der die nötigen Informationen mitteilt:

```
$ apt-get install isc-dhcp-server

option domain-name "example.com";
default-lease-time 600;
max-lease-time 7200;
# Boot-Anfragen beantworten
allow booting;
allow bootp;
```

```
# Der nächste Abschnitt muss unter Umständen an Ihre
# Situation angepasst werden.
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.200 192.168.1.253;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
    option domain-name-servers 192.168.1.3;
    # IP-Adresse des TFTP-Servers
    next-server 192.168.1.3;
}
```

TFTP-Server

Und natürlich noch einen TFTP-Server, der die Dateien ausliefert. Dies geht ganz schnell mit:

```
$ apt-get install tftpd-hpa
$ vim /etc/default/tftpd-hpa
TFTP_DIRECTORY="/tftpboot/"
```

Samba Server Configuration

Letzter Schritt ist die Bereitstellung der Windows-Installationsmedien via Netzwerk, also per Samba.

```
$ apt-get install samba
```

Zunächst kopieren wir den Inhalt des Windows-Installationsmediums in unser TFTP-Verzeichnis:

```
$ mount -o loop Windows7.iso /mnt/cdrom
$ sudo cp -R /mnt/cdrom/* /tftpboot/distros/windows/win7
```

Dann benötigen wir noch einen Benutzer der sich am Samba-Server anmelden darf

```
adduser --no-create-home --disabled-login --shell /bin/false wininst
smbpasswd -a 'wininst'
```

Und eine passende Samba-Konfiguration, welche die Installationsmedien exportiert (ein anschliessendes `service smb restart` nicht vergessen!).

```
## /etc/samba/smb.conf
[win7]
path = /tftpboot/distros/windows/win7
comment = Windows 7 Installation DVD
valid users = "wininst"
writeable = no
browseable = yes
read only = yes

[win8]
path = /tftpboot/distros/windows/win8
comment = Windows 8 Installation DVD
valid users = "wininst"
writeable = no
```

```
browseable = yes
read only = yes
```

Das System auf der iSCSI-Freigabe installieren

Windows 7

PXE-Menu erweitern

Wir erweitern unser Menu um einen weiteren Eintrag, mit dem später das eigentliche, auf der iSCSI-Freigabe installierte, System gestartet werden kann, nachdem einmalig der soeben angelegte Installationseintrag aufgerufen wurde. Zunächst wieder das iPXE-Skript:

```
## /tftpboot/iscsi/win7.iscsi
#!ipxe
dhcp
set gateway 0.0.0.0
set keep-san 1
sanboot iscsi:192.168.0.50:::iqn.2015-04.me.spiller:windows7_master
boot
```

Und dann der PXE-Menu-Eintrag:

```
## /tftpboot/pxelinux.cfg/default
LABEL Windows 7
MENU LABEL Windows 7
MENU DEFAULT
KERNEL ipxe.lkrn
INITRD iscsi/win7.iscsi
```

Zur Installation des Systems den Menueintrag „Windows 7 Install“ booten. Dies startet die WinPE-Umgebung, in deren Kommandozeile dann die Samba-Freigabe gemounted werden und das Setup gestartet werden kann:

```
$ net use z: \\192.168.1.123\win7
Enter the user name ... : 192.168.1.123\wininst
Enter the password... :

z:
z:\>setup.exe
```

Dies startet den bekannten Installationsprozess. Wichtig ist, dass für die Dauer der Installation eine lokale Festplatte angeschlossen ist. Nach erfolgreicher Installation kann diese entfernt werden.

Windows 8

Wie oben benötigen wir noch folgende Einträge:

```
## /tftpboot/iscsi/win8.iscsi
#!ipxe
dhcp
set gateway 0.0.0.0
```

```
set keep-san 1
sanboot iscsi:192.168.0.50::::iqn.2015-04.me.spiller:windows8_master
boot
```

```
## /tftpboot/prelinux.cfg/default
LABEL Windows 8
MENU LABEL Windows 8
KERNEL ipxe.lkrn
INITRD iscsi/win8.iscsi
```

Auch hier wieder zur Installation den Menueintrag „Windows 8 Install“ auswählen, die Samba-Freigabe mounten und setup.exe starten. Auch hier muss wieder eine physikalische Festplatte für die Dauer der Installation vorhanden sein. Wird nach der Installation der „Windows 8“-Eintrag gestartet, kann beim erstmaligen Start eine Fehlermeldung bzgl. fehlender Treibersignierung auftreten. Hier im ersten Menu F8 drücken und danach mit F7 die Treibersignierung deaktivieren. Danach läuft das Setup normal weiter.