

Netbooting Microsoft Windows 7 and XP

Chris Holman*

Centre for Advanced Internet Architectures, Technical Report 130226A

Swinburne University of Technology

Melbourne, Australia

6963420@student.swin.edu.au

Abstract—This tutorial shows how to set up Microsoft Windows 7 and XP to be booted using network attached storage. iSCSI support is built in to a Windows 7 installer using the Windows 7 Automated Installation Kit. DHCP, an iSCSI server and a TFTP server are configured on the server to provide a PXE client and an iSCSI target. Once set up, a client PC will be able to operate entirely without a local disk.

Index Terms—Windows 7, Windows XP, PXE, netbooting

I. INTRODUCTION

PXE (“Preboot eXecution Environment”) [1] booting, or net booting, can be used to load an operating system over the network without using the local hard drive. While FreeBSD and Linux Operating Systems (OSes) support a number of different PXE boot methods (including NFS, TFTP, HTTP, iSCSI and more), Windows only provides support for iSCSI - a protocol for abstracting remote storage as virtual block devices, similar to a hard drive.

The benefits of PXE booting an OS include reduced energy costs (fewer hard drives), easier maintenance (reimage the PC without it being on) and flexible boot options (have multiple OS options without needing to repartition your hard drive).

This report will present a guide to setting up an iSCSI backed PXE boot environment and installing Windows 7 to it. A Windows PreInstallation (PE) environment will be used to load the iSCSI drive before beginning setup. The changes to the process required for Windows XP are detailed in Appendix A

The report is structured as follows: some background is detailed in Section II; requirements and details of our test bed are in Section III. Instructions on setting up the required servers are in Section IV. Instructions on configuring the PXE booting are in Section V. Creating

and testing the Windows PE are detailed in Section VI. Installing to the iSCSI drive is covered in Section VII. The report concludes with Section IX and an appendix.

II. BACKGROUND

A. DHCP

Dynamic Host Configuration Protocol’s (DHCP) [2] primary purpose is to assign IP addresses and other network configuration details to hosts. It is used in this tutorial for the purposes of providing network and PXE booting configuration details, including TFTP server settings and configuration filenames.

B. TFTP

Trivial File Transfer Protocol (TFTP) [3] is a simple protocol to transfer files. It is commonly used in PXE scenarios because of its simplicity. The iPXE ROM and configuration are stored on the TFTP server, and are loaded by the PXE boot ROM on the Network Interface Card (NIC).

C. iPXE

iPXE [4] provides advanced PXE functionality by extending the range of boot options to include iSCSI and other protocols. It can be flashed on to a NIC’s boot ROM and used instead of the existing one or PXE booted via TFTP.

Once iPXE is loaded, our DHCP will provide the name of the configuration file, `conf.ipxe`. This file is interpreted by iPXE. iPXE can also chainload another executable file using this same method - including from a HTTP or FTP URI, unlike a standard PXE environment.

D. iSCSI

SCSI (Small Computer System Interface) is a protocol which allows a computer to communicate with directly attached storage devices, such as hard drives and optical drives. iSCSI [5] encapsulates the SCSI protocol within

*Author was undertaking an IBL placement at CAIA when this report was written

IP packets. iSCSI storage devices appear as a directly attached disk.

The iSCSI client is called the iSCSI initiator, while the iSCSI target refers to the storage device on the iSCSI server. An iSCSI target does not need to be a physical disk on the iSCSI server - it can be a file, a partition or any block device. In this tutorial, a file is used as the target, but this can be exchanged for any of the supported storage mechanisms easily. This tutorial uses the `istgt` iSCSI target software [6].

E. The boot process

At boot time, the PC operates in a mode that allows iPXE to hook in to the BIOS and make the iSCSI drive appear as a physical drive. The initial stages of the OS are loaded while the PC remains in this mode. Once the OS switches to “protected” mode, iPXE is no longer able to operate in that manner, and is unloaded. When iPXE is unloaded, the iSCSI drive is disconnected, and the OS must initiate its own connection. Using information from the iBFT (iSCSI Boot Firmware Table), which iPXE populates during its execution, the OS re-connects the iSCSI target and continues booting.

If the OS does not have iSCSI boot support, then when it switches to protected mode it will completely lose access to its boot disk and be unable to continue booting.

Windows supports iSCSI booting, but only if iSCSI is where Windows was initially installed to. To include support for this within the Windows installation media, Windows Automated Installation Kit (AIK) [7] is used.

III. REQUIREMENTS

All configuration files referenced throughout this document have full examples available at [8]. Some lines have been wrapped over multiple lines in this document. If a line ends with a slash (`\`), it continues on the next line.

A. Hardware

- A Linux or FreeBSD server - the instructions cover FreeBSD, but are easily adaptable to other BSDs or Linuxes.
- 10GB or more of free disk space on the server
- A PC or virtual machine running Windows 7

B. Software

- iPXE v1.0.0 [9]
- `istgt` v0.5, extra version 20121028 [6]
- Windows 7 AIK [7] (and a Windows PC or VM to use it on)

- Windows 7 installation media
- `inetd` TFTP [10]
- ISC DHCPD v4.2.4-P2 [11]

C. What we used

Client 1: HP Compaq DC7800. The internal hard drive was blank. The PC has an onboard Intel gigabit NIC.

Server: HP Compaq DC7900, running PC-BSD 9.0-RELEASE x64. All software except iPXE was installed through the FreeBSD ports tree, which was up to date as of 2012-12-20. iPXE was compiled on a Linux PC, as it would not compile under FreeBSD. A pre-compiled version is available at [8].

Figure 1 shows how the PCs are networked.

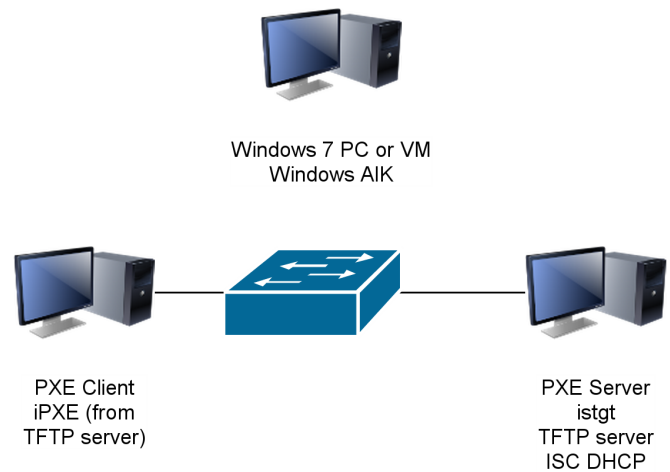


Figure 1. Software used

IV. SERVER SETUP

If you do not control your network, you should ensure that you isolate the testbed network from your usual network. The tutorial involves setting up DHCP, which can interfere with your existing network operation. In a home environment, the worst case scenario is that some PCs will not be able to access each other or the internet until after you have removed your DHCP server from the network. In many corporate environments, your network port will be shut down if a DHCP server is detected.

However, an internet connection is temporarily needed during the installation of software, as FreeBSD ports require internet access to fetch content.

Our example test bed will use `10.0.0.0/24` as a network and our server will be configured with an IP address of `10.0.0.1`.

A. DHCP

On your server, start by installing the ISC DHCP server. On FreeBSD, it is available as a port under `net/isc-dhcp42-server`. As root, install using `cd /usr/ports/net/isc-dhcp42-server && make install clean`. The default options are acceptable.

Once installed, open the configuration file - `/usr/local/etc/dhcpd.conf`.

The example simple configuration file is split in to two categories - subnet declarations, which start with a “subnet” line and contain everything within the following braces; and the global section, which is everything that is not part of a subnet declaration.

Add the following subnet declaration anywhere within the global section.

```
subnet 10.0.0.0 netmask 255.255.255.0 {
  authoritative;
  range 10.0.0.100 10.0.0.254;
  if exists user-class and
    option user-class = iPXE
  {
    filename conf.ipxe;
  }
  else
  {
    filename undionly.kpxe;
    option routers 10.0.0.1;
  }
  next-server 10.0.0.1;
}
```

This configuration declares a subnet (10.0.0.0/255.255.255.0), then states that the DHCP server is authoritative for the network interface this subnet is on. An authoritative server will ensure that all DHCP discover requests from clients result in the client being assigned an address. This is important if a client has an old DHCP lease it attempts to renew.

The `range` directive specifies the inclusive range of IP addresses that may be assigned to DHCP clients.

The `if/else` block tests for if the DHCP client is iPXE or not and conditionally sets the `filename` directive. This is covered in [V](#). It also conditionally provides the gateway (option `routers`) to clients. iPXE should not be given a gateway unless the iSCSI target is not on the same subnet. See [Appendix B](#) for more information.

The `next-server` directive dictates the TFTP server to be used by a PXE client.

Restart the DHCP daemon using:

```
server isc-dhcpd onerestart.
```

1) *Advanced DHCP configuration:* The above configuration assumes that every PC on the network will be netbooting from the same image. This presents two problems - the first is that multiple clients using the same NTFS partition simultaneously will result in significant data corruption; the second is that you may not want every device on your network netbooting - eg, some Cisco devices treat the `filename` option as their configuration file, and complain about receiving an invalid configuration file.

A PC will only try to netboot if it receives the `filename` option from the DHCP server. Using a host definition, the `filename` option can be conditionally provided. Place the host definition at the end of the DHCP configuration file, after the existing subnet declaration. Multiple host definitions may be specified in separate host definitions.

```
host arbitrary-name {
  hardware ethernet 11:22:33:44:55:66;
  if exists user-class and
    option user-class = iPXE
  {
    filename conf.ipxe;
  }
  else
  {
    filename undionly.kpxe;
  }
}
```

Then remove the `filename` lines from within the subnet declaration.

This host definition defines the host `arbitrary-name`. It is good practice to have the given name match the hostname of the PC. The `hardware-ethernet` line gives the MAC address of the PC that this host definition is for.

The same `if/else` block in the subnet declaration that was controlling the `filename` option is now placed in the host definition, performing the same task, but limiting the scope of the `filename` option to only a single host on the network.

B. TFTP

Now that our hosts can automatically configure themselves, they need somewhere to get images from. TFTP is a simple file transfer protocol, commonly used to “bootstrap” an OS by loading the kernel or another

small, specialised, boot loader. We will be using it to provide iPXE and its configuration file. iPXE gets the address of the TFTP server from the DHCP server's `next-server` option.

`inetd` provides a TFTP server on both FreeBSD and Linux. This is typically installed by default.

Open `/etc/inetd.conf` and find the line that begins `"#tftp dgram udp"`. Remove the `#` and ensure the last parameter is `"/tftpboot"`.

Save the file, create the `/tftpboot` folder, `chmod 444 /tftpboot` and restart the `inetd` daemon using service `inetd onerestart`.

C. *istgt*

`istgt` is an iSCSI server. It will act as an iSCSI target for our net booted Windows.

On FreeBSD, `istgt` is available as a port under `net/istgt`. As root, install using `cd /usr/ports/net/istgt && sudo make install clean`. The default options are acceptable.

A file will be used to store our data. Create an empty file of your desired size - we assume there is a `/netboot` directory to store images in. We used 10GB for our Windows 7 32 bit installation. Microsoft recommends 16GB for 32 bit and 20GB for 64 bit installations.

```
dd if=/dev/zero \
of=/netboot/windows7.img \
bs=10m count=1000
```

All of `istgt`'s configuration files are located in `/usr/local/etc/istgt/` on FreeBSD. Copy `istgtcontrol.conf.sample` to `istgtcontrol.conf`, then open it.

Every iSCSI drive has a unique name, called the IQN (iSCSI Qualified Name). The `TargetName` in `istgtcontrol.conf` sets the IQN of the disk.

Alter the `TargetName` line to be:
`TargetName iqn.2012-12.com.example:windows7`

'iqn' indicates that it is an IQN formatted name. '2012-12' is the date the server was created. 'com.example' is the Fully Qualified Domain Name (FQDN) of the server, in reverse. 'windows7' is an arbitrary name for the disk. The first three fields form a server identifier, and the fourth identifies the disk within the server. The date and FQDN fields are a naming convention only, and do not have any effect aside from giving the disk a unique name. You may change the IQN according to your environment, ensuring to change the configuration accordingly wherever it is used.

Replace `"localhost"` on the line `"Host localhost"` with your IP address.

Save and exit.

Copy `istgt.conf.sample` to `istgt.conf`, then open it.

Under the `[Global]` section, change the `NodeBase` to `iqn.2012-12.com.example`

Under the `[LogicalUnit1]` section, change the `TargetName` to `windows7`, then add the following line, also in the `[LogicalUnit1]` section:

```
LUN0 Storage /netboot/windows7.img \
10737418240
```

The number at the end of the line is the size of the image in bytes. You can confirm this by checking the filesize through `ls -l /netboot`

Under `[PortalGroup1]`, change the line `Portal DA1 192.168.2.1:3260` to reflect your IP address.

Likewise, under `[InitiatorGroup1]`, change `Netmask 192.168.2.0/24` to reflect your network's subnet - `10.0.0.0/24` in our example.

Save the file and start `istgt`.

If you have an OS that supports connecting to iSCSI, you can try connecting to it now to test it. On Windows, open a command prompt and try:

```
net start msiscsi
iscsicpl.exe
```

Then on the window that opens, enter the IP address of the server in the box at the top and click the find button next to it. It should automatically find and mount the drive, and it should be visible in disk manager.

V. PXE SETUP

When a PC attempts to PXE boot, the boot ROM obtains a network configuration from DHCP. Part of the configuration can be boot information. The `filename` option set in our DHCPd configuration file tells the boot ROM which file to obtain from TFTP. The DHCP server is configured to provide a different `filename` setting depending on which `user-class` option is sent with the DHCP request.

The default `filename` is `undionly.kpxe` - this is the UNDI (Universal Network Driver Interface) version of the iPXE image. There are card-specific versions of iPXE, however UNDI is suitable for almost any network card manufactured in the last decade.

A. *Build iPXE*

A pre-compiled version of iPXE is provided alongside the example files [8]. If compiling iPXE, be aware that at the time of writing it did not compile under FreeBSD 9.1-RELEASE or 10.0-CURRENT (Feb 2nd release) due to a compiler segmentation fault. It was successfully compiled under Linux (Ubuntu 12.04.1 x64)

Create the iPXE image by checking out the latest version from Git and compiling it:

```
git clone git://git.ipxe.org/ipxe.git
cd ipxe/src
make bin/undionly.kpxe
```

Transfer ipxe/bin/undionly.kpxe to /tftboot/ on the server.

You can optionally embed your configuration in to iPXE by creating the configuration file (see next step) and appending `EMBED=conf.ipxe` to the make line. Any embedded configuration can still be overridden by a valid filename `DHCP` parameter.

You can also install iPXE to your NICs Boot ROM, negating the need for DHCP and TFTP. The procedure is device specific and outside the scope of the tutorial.

B. iPXE configuration

The filename we provide to iPXE through DHCP dictates its configuration file.

The example configuration file hooks the iSCSI drive in to the iBFT, and returns control to the BIOS. Doing this during iPXE execution enables the BIOS to be aware of it, which is important for when the Windows installation happens.

```
#!ipxe
set keep-san 1
sanhook --drive 0x80 \
iscsi:10.0.0.1:::iqn.2012-12.com.example:windows7
exit
```

`keep-san` makes it such that the iSCSI drive is kept in the iBFT, even if no OS is found on it.

`sanhook` initiates the iSCSI connection to the specified drive. In the drive address, 'iscsi' identifies it as an iSCSI disk, '10.0.0.1' specifies the server IP address and `iqn.2012-12.com.example:windows7` is the IQN set in `istgt`. The `--drive 0x80` argument sets the BIOS drive number.

`exit` gracefully exits and passes on to the next boot device - either our installation media or the iSCSI disk.

iPXE supports menu creation, so it is possible to make mounting the iSCSI disk optional. See Appendix C for more information.

If you have issues booting the installation media in Section VII, try changing the drive number of the iSCSI disk to 0x81 or 0x82 using the `--drive` option of `sanhook`.

1) *Advanced iPXE configuration*: If you followed the steps in Section IV-A1 to make host-specific DHCP configurations, you may also make host-specific iPXE configurations. [12] shows how to use the MAC address to dynamically configure your

iPXE environment using the `${net0/mac}` or `${net0/mac:hexhyp}` variables. You can use these to load a host-specific configuration file with `chain http://10.0.0.1/boot.php?mac=${net0/mac}` and making the appropriate script on the web server.

VI. WINDOWS AIK

The Windows Automated Install Kit [7] allows system administrators to easily deploy customised images of Windows to PCs and create "Preinstallation Environments". Preinstallation environments (PEs) allow a user to boot a lightweight Windows environment without installing Windows. We will create our own PE with the iSCSI drivers built in, such that Windows setup can see the disk in order to install Windows to it. Note that our Windows PE image is not a setup image itself, but is merely the boot media we use. Once Windows PE is loaded, we will launch setup from the normal Windows installation media.

These steps are based on a Microsoft Technet article [13].

Our example will use "i386" as the architecture. If you are making a 64-bit OS image, you need to replace i386 with "amd64" where ever it appears.

After download and installing the AIK from [7], open a command prompt as an administrator.

```
cd \
mkdir winpe
cd c:\Program Files\Windows
AIK\Tools\PETools\
copyype.cmd i386 c:\winpe\
```

This will create a `winpe.wim` image file in `c:\winpe\`, as well as a number of directories. Copy and rename `winpe.wim` to `c:\winpe\ISO\sources\boot.wim`.

```
Copy c:\Program Files\Windows
AIK\Tools\i386\bcdboot.exe to
c:\winpe\ISO\
```

```
Copy c:\Program Files\Windows
AIK\Tools\i386\oscdimg.exe to
c:\winpe\Boot\
```

```
Copy c:\Program Files\Windows
AIK\Tools\PETools\i386\etfsboot.com
to c:\winpe\Boot
```

Now we need to mount the `boot.wim` image and make some modifications.

```
mkdir c:\mount
imagex /mountrw
c:\winpe\ISO\sources\boot.wim 1
c:\mount
```

Copy the following files from a computer running Windows 7 or Windows Server 2008 R2 to the same location under `c:\mount\`. Make sure that the source PC uses the same architecture (i386 or amd64) that your image is:

```
\windows\System32
iscsicli.exe
iscsicpl.dll
iscsicpl.exe
iscsidsc.dll
iscsied.dll
iscsiexe.dll
iscsium.dll
iscsiwmi.dll
\windows\System32\Drivers
msiscsi.sys
\windows\System32\en-US
iscsicpl.dll.mui
iscsicpl.exe.mui
iscsidsc.dll.mui
iscsiexe.dll.mui
```

Obtain `PE_Sys.reg` and `PE_Soft.reg` from [8] - these will make some modifications to the registry inside the image.

Open `regedit.exe`, click on “`HKEY_LOCAL_MACHINE`”, then open the file menu and “`Load Hive`”. Open the file `c:\mount\windows\system32\config\software`. When asked for a name, enter `PE_Soft`. (the name is important - do not use a different name). Repeat this step, opening the System file in the same directory, and calling it `PE_Sys`.

Merge the two `.reg` files obtained earlier into the registry by double clicking them, then select `PE_Soft`, then the file menu, and “`Unload Hive`”. Select `PE_Sys` and “`Unload Hive`” again.

Close any Windows explorer windows, then unmount the image using `imagex /commit /unmount c:\mount`

Create the ISO file using:

```
cd \winpe\boot
oscdimg.exe -n -betfsboot.com \
c:\winpe\ISO c:\winpe\winpe.iso
```

Burn the ISO image to a CD. Make sure that you are able to boot from it.

VII. BOOTING AND INSTALLATION

In your computer’s BIOS, set the boot order to boot from the network, then from optical drive or USB, depending on your installation media. Insert the Windows

PE disc. Reboot, and wait for Windows PE to finish loading. Since the first boot option is Network booting, iPXE will load and mount the iSCSI drive, then hand control back to the BIOS, which will then boot Windows PE from the optical or USB media.

If the BIOS gives an error message about invalid system disk or INT13, see Appendix E.

Once the WinPE environment has loaded, you will be presented with a command prompt.

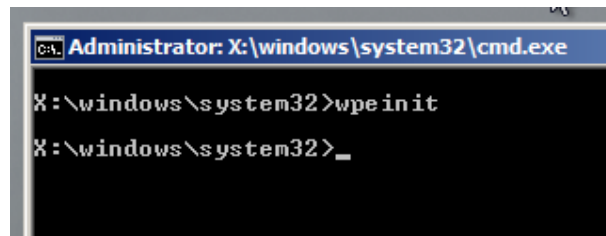


Figure 2. The initial state of the Windows PE

In the command prompt, run “`net start msiscsi`”.

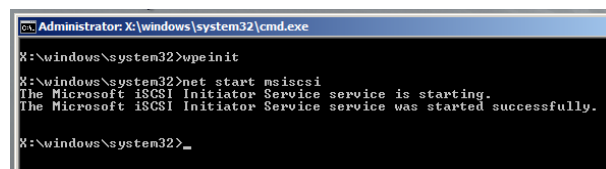


Figure 3. Starting the Microsoft iSCSI service

Then type ‘`iscsicpl.exe`’. If everything worked, a window will open and there will already be an entry in the list of targets (as in Figure 4). If there isn’t, then reboot and watch for any errors during iPXE execution. If there are no errors, go back to the iPXE setup in Section V-B and change the drive number of the iSCSI device.

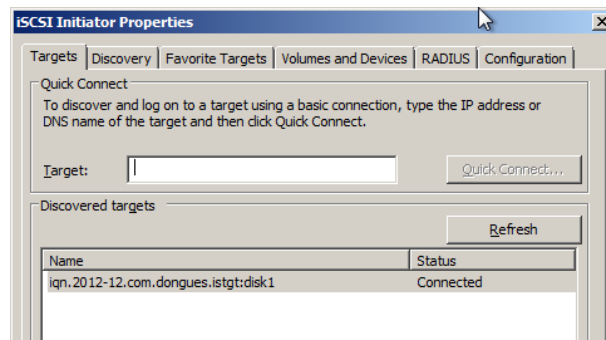


Figure 4. The iSCSI Control Panel

Remove the Windows PE disc and insert the Windows 7 installation disk. In the command prompt, change to

the optical drive by entering “d:” and pressing enter. Check that this is the right drive by doing “dir” and ensuring setup.exe, boot\ and sources\ are in the list, as in Figure 5. If not, try e:, f: and so on.

```

Administrator: X:\windows\system32\cmd.exe
X:\Windows\System32>d:
D:\>dir
Volume in drive D is GRMCENUOL_EN_DUD
Volume Serial Number is E7B5-DE04

Directory of D:\

11/20/2010  05:32 PM                43 autorun.inf
11/20/2010  05:32 PM                <DIR>          boot
11/20/2010  05:32 PM           383,786 bootmgr
11/20/2010  05:32 PM                <DIR>          efi
11/20/2010  05:32 PM           112,400 setup.exe
11/20/2010  05:32 PM                <DIR>          sources
11/20/2010  05:32 PM                <DIR>          support
11/20/2010  05:32 PM                <DIR>          upgrade
               3 File(s)           508,069 bytes
               5 Dir(s)                0 bytes free

D:\>

```

Figure 5. The contents of the Windows installation disk

Enter “setup.exe” in to the command prompt, and wait for the setup UI to load. Once it loads, continue installation like you would for any normal Windows installation. The iSCSI target will appear as a normal hard drive in the drive selection step.

After setup finishes, it should reboot to your diskless Windows installation.

VIII. TROUBLESHOOTING

A. The server configuration

In the BIOS of your client, make sure network booting is enabled and that it is the first in the boot order. Save and exit.

If PXE is working, after the BIOS has completed Power-On Self Test (POST) , you should see a screen with text at the top that mentions a PXE Boot Agent, or a Boot ROM, or similar (See Fig. 6). It should indicate that it is then attempting to establish a link, then acquire a DHCP address. If the DHCP request times out, make sure that ISC-DHCPD is started, and that there are no errors in /var/log/messages on the server.

If a DHCP address is successfully acquired, it should then load iPXE. If iPXE does not load, check that the “filename” parameter in the ISC-DHCPD configuration file matches the filename in /tftpboot/. Try downloading the file yourself by doing: cd /tmp; tftp 10.0.0.1, then entering “get undionly.kpxe”. If it times out, inetd is not running. If it gives you an access error, check the permissions on both /tftpboot/ and the files inside. It is fine for these files to be world readable.

If iPXE gives an error about the iSCSI target, make sure istgt is running and that there are no typos in the iPXE configuration file.

After iPXE finishes executing its configuration script, it may give an error about “no more network devices” - this is safe to ignore.

Before Windows is install, if no bootable optical media is inserted you may get an invalid system disk error after iPXE exits. If the invalid system disk or an INT13 error message still appears when bootable optical media is inserted later, see Appendix E.

```

PXE Software Copyright (C) 1997-2000 Intel Corporation
Copyright (C) 2010 Oracle Corporation

CLIENT Mac ADDR: 08 00 27 2b ba 03 GUID: 12893264-5672-4809-93fd-8c0b5275d447
CLIENT IP: 10.1.2.184 MASK: 255.255.255.0 DHCP IP: 10.1.2.1
PXE->EB: iPXE at 9E1C:0070, entry point at 9E1C:0104
          UNDI code segment 9E1C:192, data segment 9C99:1030 (626-639KB)
          UNDI device is PCI 00:03:0, type DIX-002.3
          626KB free base memory after PXE unload
iPXE Initialising devices...ok

iPXE 1.0.0 (71727) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: HTTP iSCSI DNS TFTP AOE bImage COMBOOT ELF MBOOT PXE PXEXT Menu

net0: 08:00:27:2b:ba:03 using undionly on UNDI-PCI00:03:0 (open)
Linkup, TX:0 RX:0 BX:0 BXE:0
DHCP (net0 08:00:27:2b:ba:03)..... ok
net0: 10.1.2.133/255.255.255.0
Next server: 10.1.2.1
Filename: conf+ipxe
tftp://10.1.2.1/conf.ipxe... ok
Registered SAN device 0x80

```

Figure 6. An example successful PXE boot with iPXE

If Windows setup loads and cannot detect the iSCSI drive, reboot and watch the boot process - make sure that iPXE successfully registers the SAN device. To make iPXE pause, edit the conf.ipxe file and insert ‘prompt’ on a new line before exit.

IX. CONCLUSIONS

This report has provided a tutorial on the steps required to boot Windows 7 and XP without a local disk. ISC DHCPD, iPXE and TFTP are used to provision a PXE boot environment, istgt is used as an iSCSI target and the Windows Automated Installation Kit is used to implement iSCSI support in to the Windows 7 installer. Once completed, Windows 7 and XP are able to use exclusively network attached storage to boot and operate.

REFERENCES

- [1] “Preboot Execution Environment (PXE) Specification.” <http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf>. Accessed Feb 5th, 2013.
- [2] R. Droms, “RFC 1531 - Dynamic Host Configuration Protocol.” <https://tools.ietf.org/html/rfc1531>. Accessed Feb 5th, 2013.
- [3] K. R. Sollins, “RFC 783 - TFTP Protocol (revision 2).” <https://tools.ietf.org/html/rfc783>. Accessed Feb 5th, 2013.
- [4] “iPXE.” <http://ipxe.org>. Accessed Feb 5th, 2013.
- [5] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, and E. Zeidner, “RFC 3720 .” <http://www.ietf.org/rfc/rfc3720.txt>. Accessed 3rd Jan, 2013.
- [6] “FreshPorts – net/istgt.” <http://www.freshports.org/net/istgt/>. Accessed Feb 5th 2013.

- [7] "Download Windows (R) AIK for Windows (R) 7 from Official Microsoft Download Center." <http://www.microsoft.com/en-us/download/details.aspx?id=5753>. Accessed 16th Dec, 2012.
- [8] C. Holman, "Example configuration files and other resources." <http://caia.swin.edu.au/tools/130226A/>. Accessed 25th Feb, 2013.
- [9] <https://git.ipxe.org/ipxe.git/commit/717279a294e414218deb1c89d33422b209574053>. Accessed Dec 13th, 2012.
- [10] "The inetd Super-Server." <http://www.freebsd.org/doc/handbook/network-inetd.html>. Accessed Feb 5th 2013.
- [11] "FreshPorts - net/isc-dhcp42-server." <http://www.freshports.org/net/isc-dhcp42-server/>. Accessed Feb 5th 2013.
- [12] "[cfg:mac]." <http://ipxe.org/cfg/mac>. Accessed Jan 4th, 2013.
- [13] "Creating, Validating, and Testing Boot Images." [http://technet.microsoft.com/en-us/library/ee619769\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/ee619769(WS.10).aspx). Accessed 16th Dec, 2012.
- [14] "SAN Boot Conf Installer, 0.9.7." <http://git.etherboot.org/sanbootconf.git/blob/60ed443b667412b0ce6f9b8173c6a1220c0728bd:sanbootconf.msi>. Accessed 16th Dec, 2012.
- [15] "Microsoft iSCSI Initiator Version 2.09." <http://www.microsoft.com/en-au/download/details.aspx?id=18986>. Accessed 16th Dec, 2012.
- [16] "If you start a system from iSCSI, the gateway specified in the iSCSI Boot solution will always be used by Windows to communicate with the iSCSI Target." <http://support.microsoft.com/kb/960104>. Accessed 3rd Jan, 2013.
- [17] G. Armitage, "Adding Intel PXE Boot under FreeBSD/VirtualBox." <http://grenville.wordpress.com/2012/03/18/adding-intel-pxe-boot-under-freebsdvirtualbox/>. Accessed 16th Dec, 2012.
- [18] "VirtualBox LAN boot ROM." <http://www.etherboot.org/wiki/romburning/vbox>. Accessed 16th Dec, 2012.

APPENDIX

A. Altering the procedure for Windows XP

Start by installing Windows XP in to VirtualBox. Be warned that while VirtualBox will create an expandable virtual disk file that grows as more disk is used, the final image we use will be as big as the capacity of the drive. Once XP is installed, install both [14] and [15] - for the second link, download the x64 or x86 version under "Checked Build links for Boot Initiator Builds".

Shutdown the XP Virtual Machine.

From the command line on the VirtualBox host, run:

```
qemu-img convert -f vmdk -O raw \
/path/to/existing-disk.vmdk \
new-disk.img
```

qemu-img is available in the FreeBSD port emulators/qemu or on Ubuntu as the package qemu.

The resulting new-disk.img file is the image that istgt should be configured to use as its target in Section IV-C. Follow the rest of the tutorial, skipping Sections VI and VII.

B. Reasons for the static route for iSCSI in Windows 7

The configuration information for the iSCSI drive, including the network configuration, is stored in the iBFT (iSCSI Boot Firmware Table). Windows uses this information to initiate its connection to the iSCSI target.

One of the assumptions that Windows makes is that you will only ever have your iSCSI connection on a NIC that is dedicated to iSCSI, and that if it is configured it with a default router, it's because the router is required to access the iSCSI target [16]. Because of this assumption, Windows will always add a static route to the iSCSI target, via the default gateway that is in use at the time the iSCSI connection is initiated. While the intention of this static route is to ensure the iSCSI target is always accessible, the consequence is that your iSCSI traffic may end up being forwarded through your router and back out on to the same LAN. This is far from ideal, for example, if the router is a consumer ADSL modem or if a wireless link is in the path to the router.

The workaround is to make the provision of a router address conditional on the DHCP client not being iPXE, as in the example DHCP configuration provided in Section IV-A.

By making this change, iPXE does not have a gateway configured when it inserts the configuration information into the iBFT, therefore Windows will not add a static route and will direct-connect to the iSCSI target. When Windows loads its own networking stack, it will not send iPXE as its user-class, so will be given the option routers configuration and the windows networking stack will operate as normal.

If a router is required to access the iSCSI target, then the option routers line should be moved to outside of the if/else block - after the next-server line, for example.

C. Creating an iPXE menu

iPXE supports creating a menu. The menus are simple, but adequate for providing choice in booting. They are declared in the following format:

```
menu Title of the menu
item value Label of this item
item value2 Label of the second item
choose answer
```

One can then use the answer to direct the script to a certain location. Here is an example menu script that gives the option of mounting iSCSI or bypassing it:


```

menu Please choose your OS
item win7 Windows 7 (iSCSI)
item default Defer to next in boot order
choose --default default os && goto ${os}
:win7
set keep-san 1
sanhook iscsi:10.0.0.1::::iqn.2012-12.com.example:disk1
exit
:default
exit

```

D. iSCSI booting Windows in VirtualBox

VirtualBox supplies two different families of NICs that it can emulate, Intel and AMD - Windows only has drives for the Intel family. However, the Intel virtual NICs do not come with a boot ROM, so this must be installed manually. This blog post [17] details how to configure VirtualBox with a boot ROM. Alternatively, you can compile iPXE to be the boot ROM, by following this article [18]. Note that, while the instructions are for gPXE, they also valid for iPXE.

Also note that the tested version of VirtualBox (4.1.22OSE) will not try secondary boot devices (ie, hard disk or optical media) once a PXE ROM has been loaded (eg, iPXE). This is an important caveat, as the given instructions rely on the BIOS to try subsequent boot

devices once the iSCSI drive is mounted. If Windows is installed elsewhere (eg, in VMWare) instead, you can boot Windows by changing sanhook to sanboot in the iPXE configuration. sanboot immediately boots from the iSCSI drive. sanhook will mount the drive, and wait for the BIOS to boot from it once the BIOS reaches it in its boot order. The latter allows you to boot from alternate media (eg, a live CD) and still have the iSCSI drive visible - this is a requirement to install Windows 7.

Windows XP installation is not affected by this, as XP is installed using traditional methods, then the installation is modified to support iSCSI booting.

E. PCs that fail to continue booting after loading a PXE

Some PCs will not try further boot devices if a PXE ROM executes then exits - such as how iPXE is operating in this tutorial. This problem does not appear to affect the majority of PCs. If this happens, try installing on another computer. Windows must be installed straight to iSCSI, but once installed, may be booted on a different PC.

Installing to PCs that exhibit this behaviour is not within the scope of this report.