

## Zeroshell as TFTP server

On June 22 a new update has been released by Fulvio BF22 (June 30, 2013)

##

Code (date)	BF22 (June 30, 2013)
Description	This update installs a TFTP server. The root directory is /tftpboot.
Release	v1.0
Download	<a href="http://www.zeroshell.net/listing/BF22-tftp.tar.bz2">http://www.zeroshell.net/listing/BF22-tftp.tar.bz2</a>
Installation	cd /Database wget http://www.zeroshell.net/listing/BF22-tftp.tar.bz2 tar xvfj BF22-tftp.tar.bz2 cd BF22 ./install.sh

So now you can install an thin client or a server on network.

I was playing with booting SmartOS [[www.smartos.org](http://www.smartos.org)] and step by step with a wonderful post of the Alan O'Dea [<http://blog.alainodea.com/en/ipxe-smartos>] and Ben Rockwood [<http://cuddletech.com/blog/?p=779>] and on wiki of the SmartOS [<http://wiki.smartos.org/display/DOC/PXE+Booting+SmartOS>]

But why don't try with Zeroshell ?

- 1) install the new update

Now you have in */Database* a new directory *opt* and if you change something inside this directory all the change will be persistent after reboot.

Now just after a little theory how the pkg is added I try to improve the update with two little patch.

- a) add `--verbose` option to `in.tftpd`, so you can see more messages in logviewer
- b) add also `MANPATH` of the pkg, so you can read the man of the `tftp` and `in.tftp`

In the first case you have to change the file *etc/rc.d/init.d/tftp* inside the tar file */Database/opt/x.x/tars/BF22-files.tar.gz*

```
root@labtest opt> pwd
/Database/opt
root@labtest opt> tar tzvf ./x.x/tars/BF22-files.tar.gz
drwxr-xr-x root/root      0 2013-06-30 21:12 etc/
drwxr-xr-x root/root      0 2013-06-30 21:12 etc/rc.d/
drwxr-xr-x root/root      0 2013-08-08 16:53 etc/rc.d/init.d/
-rwxr-xr-x root/root    592 2013-08-08 16:53 etc/rc.d/init.d/tftp
root@labtest opt>
```

```
root@labtest >cd /Database/opt/x.x/tars
root@labtest tars> tar xzvf BF22-files.tar.gz
etc/
etc/rc.d/
etc/rc.d/init.d/
etc/rc.d/init.d/tftp
```

```
root@labtest tars>
```

Now you can change the file `etc/rc.d/init.d/tftp` with a new option

```
root@labtest tars>vi etc/rc.d/init.d/tftp
loadproc /Database/opt/x.x/packages/sbin/in.tftpd --listen --user apache --secure
/Database/tftpboot --verbose >/dev/null
```

Then you have to recreate the tar file with

```
root@labtest tars> tar -czvf BF22-files.tar.gz etc/
```

Now I need to add a `MANPATH` to `/etc/man.conf` and can be done with script `/Database/opt/updater` so before the last line add a new one:

```
echo 'MANPATH /Database/opt/x.x/packages/share/man' >>/etc/man.conf
```

Now tftp is ready but if you need to boot a operating system with iPXE <http://ipxe.org> then you need to add some option to DHCP configuration. The is a wonderful post on forum of zeroshell [DHCP Global Options HowTo](http://www.zeroshell.org/forum/viewtopic.php?t=2989&highlight=dhcp+howto) (<http://www.zeroshell.org/forum/viewtopic.php?t=2989&highlight=dhcp+howto> )

For SmarOS I found that the Global Option that works for me are:

```
option space ipxe;
option ipxe-encap-opts code 175 = encapsulate ipxe;
option ipxe.priority code 1 = signed integer 8;
option ipxe.keep-san code 8 = unsigned integer 8;
option ipxe.skip-san-boot code 9 = unsigned integer 8;
option ipxe.syslogs code 85 = string;
option ipxe.cert code 91 = string;
option ipxe.privkey code 92 = string;
option ipxe.crosscert code 93 = string;
option ipxe.no-pxedhcp code 176 = unsigned integer 8;
option ipxe.bus-id code 177 = string;
option ipxe.bios-drive code 189 = unsigned integer 8;
option ipxe.username code 190 = string;
option ipxe.password code 191 = string;
option ipxe.reverse-username code 192 = string;
option ipxe.reverse-password code 193 = string;
option ipxe.version code 235 = string;
option iscsi-initiator-iqn code 203 = string;
# Feature indicators
option ipxe.pxeext code 16 = unsigned integer 8;
option ipxe.iscsi code 17 = unsigned integer 8;
option ipxe.aoe code 18 = unsigned integer 8;
option ipxe.http code 19 = unsigned integer 8;
option ipxe.https code 20 = unsigned integer 8;
option ipxe.tftp code 21 = unsigned integer 8;
option ipxe.ftp code 22 = unsigned integer 8;
option ipxe.dns code 23 = unsigned integer 8;
option ipxe.bzimage code 24 = unsigned integer 8;
option ipxe.multiboot code 25 = unsigned integer 8;
option ipxe.slam code 26 = unsigned integer 8;
```

```

option ipxe.srp code 27 = unsigned integer 8;
option ipxe.nbi code 32 = unsigned integer 8;
option ipxe.pxe code 33 = unsigned integer 8;
option ipxe.elf code 34 = unsigned integer 8;
option ipxe.comboot code 35 = unsigned integer 8;
option ipxe.efi code 36 = unsigned integer 8;
option ipxe.fcoe code 37 = unsigned integer 8;

```

and the into DHCP SERVER ---->Advance Option

put how to feed the request from thin client

The screenshot shows a DHCP server configuration interface. The main window is titled "DHCP SERVER" and has tabs for "Manage" and "Leases". The "Active on:" field is set to "ETH00". There is a "Save" button. The "Dynamic IP Configuration" section includes "Default Lease Time" and "Max Lease Time" fields, each with "Days", "Hours", and "Minutes" sub-fields. Below these are three "Range" fields (Range 1, Range 2, Range 3) with input boxes. The "Subnet Options" section includes fields for "Default Gateway", "DNS 1", "DNS 2", "DNS 3", "Domain Name", "NIS Domain", "NTP Server", and "WINS Server".

Overlaid on the main window is a "DHCP Options - Mozilla Firefox" dialog box. The URL bar shows "https://192.168.4.75/cgi-bin/kerbynet?Section=DHCP&STk=1fe838c70aafc2793972f". The dialog title is "DHCP OPTIONS" and the IP address "192.168.4.0/255.255.255.0" is displayed. The "Options" section contains the following configuration code:

```

next-server 192.168.4.75;
if exists user-class and option user-class = "iPXE" {
    filename = "smartos.ipxe";
} else {
    filename = "undionly.kpxe";
}

```

At the bottom of the dialog box are "OK" and "Cancel" buttons.

Now you have to setup the file like suggested by Aln and Ben