Exercises: FreeBSD: ccTLD Workshop: Amman, Jordan

November 26, 2007

1.) Edit /etc/make.conf [Top]

During the workshop you will be using the `make` command to install software via the FreeBSD ports system. We must override a default setting on your system so that FreeBSD will attempt to fetch the source for the port from a local server and not from the default remote site of ftp.freebsd.org.

We can specifiy this by editing the file /etc/make.conf and adding a single line (don't break the line) that will force `make` to look on a local server first. Use vi or ee to edit the /etc/make.conf file.

First, edit /etc/make.conf (you must be root to do this):

```
$ su -
# vi /etc/make.conf
```

Scroll down to the last line of the file, press the "o" (lower-case oh) key to insert a line below the last line, and then type in the following:

MASTER_SITE_OVERRIDE=ftp://noc/pub/FreeBSD/ports/distfiles/

To save and exit you must press:

:wq <ENTER>

For this class our local DNS (193.188.252.195 or noc.eu.cctld.org) and FreeBSD 6.2 will do domain completion for you, thus we can get away with only using "noc".

2.) Use ports to install the Joe editor [Top]

For this exercise you need to logged in as root in a terminal (`su -`).

The ports collection, which we installed during the FreeBSD install, is larger than the packages collection available on the first FreeBSD install cd-rom. As of November 2007 there are over 17,800 ports available. The FreeBSD Handbook has an excellent discussion of using Ports in section 4.5. In a nutshell here are a few points that might be useful.

Let's install the Joe editor version 3.x. This is an alternative to using the vi and ee editors, which you can use if you wish.

First do the following:

```
# cd /usr/ports
# make search name=joe | more
```

On the first page of output you should see something like this:

```
Port:   joe-3.5.1
Path:   /usr/ports/editors/joe
Info:   Joe's Own Editor
Maint:  toasty@dragondata.com
B-deps: aspell-0.60.4_4 gettext-0.14.5_2 gmake-3.81_1 libiconv-1.9.2_2
R-deps: aspell-0.60.4_4 gettext-0.14.5_2 libiconv-1.9.2_2
WWW:    http://sourceforge.net/projects/joe-editor/
```

From this you can see that the Joe editor resides in /usr/ports/editors/joe. So, to install Joe version 3.5.1 you do the following:

```
# cd /usr/ports/editors/joe
# make
# make install
```

Make will download the joe editor source (hopefully from our local server!), and then compile the source. Note, if you are missing any of the items listed on the "deps" lines, then these may be downloaded and compiled as well.

"Make install" will place the compiled binary files in the appropriate directories on your system and update the appropriate configuration files if necessary. You can issue the single command "`make install`" instead.

You can now type:

```
# rehash       (to re-read items in your directory path)
# man joe
```

For more information.

If you wish to deinstall a port once it is installed, for instance joe, you would do the following (please don't do this):

```
# cd /usr/ports/editors/joe
# make deinstall
```

And, if you decided that was a mistake, you can now do the following to reinstall a port (after it's been installed once):

```
# cd /usr/ports/editors/joe
# make reinstall
```

Also, note that the Joe port has been built and installed and will now appear as a package. You can type:

```
# pkg_info joe\*
```

You will see that the Joe port now appears as an installed package. Thus, you could manipulate this software with package commands (`pkg_info`, `pkg_delete`).

3.) Use the package system to add Apache version 2 with SSL support [Top]

For this exercise you need to be logged in as root.

Later in the week we may need this software to do our work. We will be installing:

- **Apache 2.2.3**: Apache version 2 web server software
- **mod_ssl**: SSL extension module to the Apache web server

This software is a single package. To make this software takes quite some time and the default options for this software will work fine for us. To speed things up the software is already available on your local server. Thus, to install using the package facility simply type:

```
# pkg_add
ftp://noc/pub/FreeBSD/6.2-RELEASE/i386/packages/All/apache-2.2.3.tbz
```

If you want to learn more about apache type:

```
# rehash
# man httpd
```

In our next exercise we will go over how to configure Apache version 2.2 to run at system startup. You may have noticed that as the software installs you are told how to do this as well...

4.) Edit /etc/rc.conf to enable Apache [Top]

For this exercise you need to be loggedin as root in a terminal.

Remember where third party software installs configuration and startup scripts?

Yes? Let's go read what the Apache 2.2 startup script says about getting the service to run at system start time.

```
# cd /usr/local/etc
# ls
```

You should notice that there is now an *apache22* directory. But, first let's look and see if there is an Apache startup script installed in /usr/local/etc/rc.d.

```
# cd rc.d # ls
```

You should see the file apache22. Either edit this file (vi, ee, joe filename), or display the file's contents to the screen (more, less), and read what it says.

The comment that says:

```
# Add the following lines to the /etc/rc.conf to enable apache22:
# apache22_enable (bool):          Set to "NO" by default.
#                                  Set it to "YES" to enable apache22
#
# etc...
```

So, this means you need to add a line to /etc/rc.conf that reads:

```
apache22_enable="YES"
```

You can place this line at the bottom of the file. For instance, to do this using the newly installed Joe editor you could type:

```
# joe /etc/rc.conf
```

Add in the line, save the file, and quit (ctrl-k-x)

Now, you can start the Apache web server. You should be in the directory /usr/local/etc/rc.d/. So, to start the server you simply type:

```
# ./apache22 start
```

In addition, you can type:

```
# ./apache22
```

To see all the various options available to you.

The server now started because you placed the entry, 'apache22_enable="YES"' in /etc/rc.conf. If that is not in place, then even if you run the Apache startup script correctly, the service itself will not start.

From this point on, when you reboot your machine the Apache web server will automatically start.

To verify that it is working for you you can type:

```
# links 127.0.0.1
```

And you should see the initial Apache web page for FreeBSD that says, "It works!". Type "q" to exit this text-based web browser.

5.) Mount a USB stick on your machine [Top]

For this exercise please use your *admin* user account. Note that the prompt will be shown as "$" to indicate commands to be done as a normal user. We will use the `sudo` facility to allow your user to issue privileged commands.

We've mentioned that USB sticks show up as a SCSI device and they are created dynamically when plugged in to your machine. We'll try mounting one on your machine now. If you do not have access to a USB stick, then watch as your instructor does this on the main screen in the classroom as well.

To mount a USB stick's file system under FreeBSD here is, more or less, what you do:

Open two terminal windows on your machine and you should be logged in as "admin" not root. In one terminal type:

```
# sudo tail -f /var/log/messages
```

When prompted for a password enter in your account's password.

Now plug-in your USB stick in an available USB slot on your machine.

In the open window where the `tail` command is running you should see some

descriptive message come up about your USB stick. From this you should be able to tell what device node FreeBSD has assigned for your device. Very likely the device will be *da0*. Thus, the first *slice* on your USB stick will be *da0s1*. You can check for this by doing:

```
$ ls /dev/da*
```

If your device appeared as some other device than *da0* and you need help ask your instructor or an assistant to have a look.

Now let's create a place in your file system where you can mount your USB stick's files. By convention we can use "/media". Under "/media" we will create the "/media/usb" directory and mount our USB stick. To do this do:

```
$ sudo mkdir /media/usb
$ sudo mount -t msdosfs /dev/da0s1 /media/usb
```

Now to see that the disk is mounted type:

```
$ df -h
```

Have a look at your files to verify that things are as you expected and then unmount your USB stick from the file system. Note, if you do `cd /media/usb` you must leave this directory before unmounting the device.

```
$ ls /media/usb
$ sudo umount /media/usb
```

And, prove to yourself that the disk is really unmounted:

```
$ df -h
```

If you have questions about any of this please speak up.

6.) Multi to Single to Multi -user mode [Top]

For this exercise you'll need to be root (or, you can use the `sudo` command).

We are going to drop down to single-user mode. This will stop your Gnome Desktop session and leave you in a /bin/sh shell as the user root.

To do this press CTRL-ALT-F1 to open one of your virtual consoles.

If you see a *login:* prompt, then log in as root. Not necessary if you rae already logged in as root.

Now type:

```
# init 1
```

When prompted with:

```
Enter full pathname of shell or RETURN for /bin/sh:
```

Just press ENTER.

Now, different form Linux, to get back to multi-user mode you can type:

```
# exit
```

And, that's it! Pretty simple. Note, there is no run-level 5 (as in Linux), or run-level 3 (often network, multi-user without GUI in Linux). In addition, we have not set your machines to default to a graphical login. To get back to your Gnome Desktop you will need to log back in as root, and the type:

```
# gdm
```

to start the Gnome Display Manager (gdm) once again. Once you have done this, then you can log in to your Gnome Desktop as the "admin" user once again.

Last modified: Mon Nov 26 11:45:37 EET 2007