FreeBSD Tutorial

Hervey Allen, Joel Jaeggli, Noah Sematimba

Index

Introduction

Many of the topics covered in this short session will be discussed in more detail in the Monday evening FreeBSD session.

Why Did we Choose FreeBSD Rather than Linux?

We are using FreeBSD 4.2. It has a number of features which make it more appropriate than Linux for use in an ISP environment:

- Very stable, especially under load
- Superior TCP/IP stack (so much so that Microsoft literally stole it for use in Windows 2000)
- A merged virtual memory and filesystem buffer cache. Linux has three virtual memory systems in the 2.4 kernel track, and none of them perform as well.
- Heavily used and tested in large service providers
- Scalability features as standard: e.g. pwd.db (indexed password database), which give you much better performance and scales well for very large sites. Linux uses a flat structure password file that is inefficient for large sites.
- Optional "softupdates" filesystem combines crash-safety of BSD filesystem with speed of Linux filesystem - see /usr/src/sys/contrib/softupdates/README
- Similarities to BSDI and other "industrial strength" Unixes
- see http://www.freebsd.org/features.html for more discussion.

Linux users will find some annoyances: for example, that 'bash' is not installed as standard, and new package management tools to learn. However, we feel that the robustness of FreeBSD is the most overriding factor for ISP's, and it's worth you getting to know and use FreeBSD. The FreeBSD Website is located at http://www.freebsd.org.

Accounts Information

ACCOUNT: root
PASSWORD: Given in class

PLEASE!*Do not change the root password.*This will cause numerous problems throughout the workshop. The machine in front of you is yours for the duration of the workshop.

IF YOU CHANGE THIS PASSWORD YOUR MACHINE MAY BE REINSTALLED FROM SCRATCH!

Creating a User Account for Yourself

There are several ways to do this. We'll use one method for now, but you can use the interactive `adduser` script if you wish. This may require additional setup on your part the first time you run the script.

Below create the exim user account and your own personal user account as well using the following commands:

- pw useradd exim -u 90
- pw useradd *username* -m -G exim,wheel -s /usr/local/bin/bash

The first command created the account *exim* with no home directory and a UID of 90. The second command created your personal userid that you will use during the rest of this workshop, automatically setup a home directory for the userid ("-m" option), placed the user in the wheel and exim groups, and set the default shell to be bash.

By placing your user in the wheel group this allows you to become the superuser *root* by using the `su` command. For more information on the `pw` command type `man pw` at the prompt.

Some Basic FreeBSD Commands

man	Help pages for commands
cd	Change Directory
ls	LiSt files/directories
cp	CoPy files/directories
rm	ReMove files/directories
mv	MoVe files/directories
mkdir	MaKe a DIRectory
ps	See currently running ProceSses
cat	ConCATenate a file to the screen (by default)
less	Display a file to the screen with editing functionality
more	Display a file to the screen and pause
tail	See the end (tail) of a a file
gzip	ZIP (compress) a file, or set of files
gunzip	UNZIP (decompress) a zip archive
bunzip2	ZIP/UNZIP alternate compression format
tar	Archive/unarchive files/directories to file or tape

grep	Look for pattern(s) in file(s)

Back to Top

Command Line Editing

FreeBSD by default uses sh or csh, which provide very rudimentary editing capabilities. **Bash** has much better editiing features so that's what the t1 accounts use. We do not recommend changing the default shell of root; rather, use '**su -m**' rather than '**su**' to get root access

```
#   bash
bash-2.03#
```

If you change the default shell for root it is possible to create a situation where you cannot get a a shell if your machine's environment becomes damaged.

Using 'vi' as an Editor

If you are unfamiliar with the 'vi' text editor, here is the absolute minimum set of commands you need to know. Beware: when you start 'vi' it is in command mode, so anything you type is interpreted as a command, not text to be entered! We have included additional commands and a longer vi tutorial as a handout for the workshop.

vi <filename> .... [ESC]	-- edit a file
i .... [ESC]	-- insert text before cursor position
A .... [ESC]	-- append text to end of current line
x	-- delete character under cursor
dd	-- delete whole line
:wq .... [ESC]	-- save and exit
:q!	-- exit without saving
ESC Shift-ZZ	-- save and exit
o .... [ESC]	-- insert a line after cursor position
dd	-- delete the current line
h j k l	-- left|down|up|right (preferred)

Post-installation Configuration

There is a menu-based system you can use for configuring the system - setting IP address, configuring hostname, adding further distribution sets and packages, adding users etc: `# /stand/sysinstall`

However, you may find that some of this is quicker to do from the command line, some of which is explained below.

Short Example Using Some FreeBSD Commands

We will be placing source for programs during the workshop in a directory called /usr/local/src. The /usr partition contains enough space for us to use this for this purpose. The /usr partition, also, contains your home directories. See the Installation Notes section below for more information on partition slices.

To create /usr/local/src do the following:

- cd /usr/local
- mkdir src

Take a look at what's in /usr/local/src (which is nothing):

- ls
- ls -la (long listing and all files - note "." files)

SSH and SCP instead of Telnet and FTP

Later in the week during the security discussion we will go over SSH in detail. For now, ***use SSH and SCP*** instead of Telnet and FTP. Telnet and FTP send your passwords and sessions out using clear text - no encryption. This is bad and a major security hole. We will, however, use FTP for *anonynmous* user sessions to get software. This is because with anonymous ftp your password is your email address.

Note: if you are accustomed to using telnet, ssh looks almost identical and will behave the same for much of what you wish to do, but ssh has *much* more functionality available if you decide to use it.

Basic SSH and SCP Commands

ssh username@host	-- Connect to host using an encrypted session
scp filename user@host:/path/filename	-- Copy filename to path at host using username for authentication
scp -r * user@host:/path/.	-- Copy all files and all subdirectories to host under path specified. This is much more difficult using ftp.
scp user@host:/path/filename \ /path/filename	-- Copy file from host authenticating with user and put it in path with filename.
scp userX@hostX:/path/filename \ userY@hostY:/path/filename	-- Copy filename from hostX authenticating as userX to hostY authenticating as userY. This is something that you cannot do with ftp.

Getting FreeBSD 4.5 Files and Others

The FreeBSD 4.5 CD-ROM contents have been made available to you locally. Additional files may be placed in /pub (see below) as needed.

- ftp noc.ws.afnog.org
- login as userid *anonymous*
- username@ws.afnog.org
- cd pub/FreeBSD [to get FreeBSD 4.5 CD-ROM files]

pkg_add: Adding Packages or Ports by Hand

Many third-party software is supplied in ready-to-use form, but is not in the base system. This is (a) to keep the base system smaller, and (b) because of different licensing terms. (The FreeBSD license is actually *less* restrictive than the GNU/GPL license under which a lot of open-source software is distributed)

All this third-party software installs under /usr/local.

You can use /stand/sysinstall to add packages, but it is quicker to use 'pkg_add' from the command line. The example below assumes that you have the FreeBSD mounted to the mount point /cdrom. For example, to add the editor 'joe':

```
# cd /cdrom/packages/All
# ls
# pkg_add joe-2.8_3.tgz
```

[For the workshop you can simply enter in the command below. Note that you must be *root* to do this.]

```
# pkg_add ftp://noc.ws.afnog.org/pub/FreeBSD/4.5-RELEASE/packages/All/joe-2.8_3.tgz
```

Note that the configuration files for third-party software are in /usr/local/etc, and scripts to start daemons are installed under /usr/local/etc/rc.d

You can also compile packages directly from the source code, if you have the "ports" distribution installed (not installed in this workshop). The ports system automatically fetches the source file via FTP or anonCVS, applies any FreeBSD-specific patches, and compiles and installs the code. A "package" is really just a "port" which has been compiled.

```
# cd /usr/ports/shells/bash
# make
# make install
# make clean
```

Sometimes you will find that a "port" exists, but no corresponding binary "package". This is usually because of licensing or export restrictions. The "port" is always able to be distributed because it does not include any software, only instructions on how to fetch and compile the software from somewhere else.

You can query installed packages, or package .tgz files, using *pkg_info*.

pkg_info -aI	-- list all installed packages (one line per package)
pkg_info joe-2.8_3	-- description of package
pkg_info -L joe-2.8_3	-- list all files in package
man pkg_info	-- read this for more details

Back to Top

Network Information

ifconfig

This is the command you use to interactively start and stop network interfaces and to define how they run. You need root access to use this command.

rc.conf

The main system configuration file is /etc/rc.conf. This file is edited by /stand/sysinstall, but it’s perfectly OK to edit this by hand. It is in this file that you configure the hostname, IP address for each interface, and so on. Changes you make in here won’t take effect until you reboot.

```
defaultrouter="80.248.72.126"
hostname="host1.t1.ws.afnog.org"
ifconfig_ed0="inet 80.248.72.1 netmask 255.255.255.248"

# On hosts where you don’t want sendmail to accept incoming port 25
# (but you still want daemons to be able to send outgoing mail):
sendmail_flags="-q30m"
```

The sendmail flag is not set in this workshop. The full list of options, and their default values, can be found in /etc/defaults/rc.conf - but don’t edit this file, edit /etc/rc.conf instead. This makes it easier to upgrade your system to a later version of FreeBSD.

Stopping and Starting the Network

You may be used to something like "/etc/rc.d/init.d/network stop" under Linux. Under FreeBSD this functionality resides in /etc/rc.conf and is parsed at system startup. So, to stop your currently running network first use ifconfig to figure out what interface you wish to stop.

- ifconfig

This displays your network interface status. Notice that "ed0" (Fast Ethernet) has an IP address assigned (via DHCP). Not that lo0 as an address as well. This is your loopback device. To bring ed0 down type:

- ifconfig ed0 down

Now to bring it back up just type:

- ifconfig ed0 up

There is quite a bit more to ifconfig and we'll discuss this during the Monday evening FreeBSD session as well, or you can type "man ifconfig" for more information. Review the "rc.conf" section above as well.

Stopping and Starting Services

By default, system services are configured in /etc/rc.conf and are started at system startup. If you need to start and stop one of these services (perhaps you changed the service's configuration file) you should do the following:

- ps -awx

or to find a particular running service, like sendmail, try:

- ps -awx | grep ssh

You'll get output that includes the process ID number on the left. If you make a change to the process's configuration, or just need to restart the process, you can do the following:

- kill -HUP process#

Note, if the service is a third party package, then you can often find configuration files for the service in /usr/local/etc/. In addition you may find shell scripts that can start and stop the service with command line parameters in /usr/local/etc/rc.d. Use these scripts instead, when available, to start and stop a service.

Back to Top

Installation Notes

Slices and Partitions

What MS-DOS calls partitions are called "slices" by FreeBSD. FreeBSD almost always sits in a single slice.

The FreeBSD slice is then divided into "partitions". Example:

```
/dev/ad0     -- first ATA/ATAPI (IDE) hard drive
/dev/ad0s1   -- first slice (MSDOS "partition") on first IDE hard drive
/dev/ad0s1a  -- first partition in this FreeBSD slice
/dev/ad0s1b  -- second partition in this FreeBSD slice
/dev/ad0s1e  -- third (usable) partition
```

For historical reasons, partitions c and d are not used. We strongly recommend you configure your partitions as:

```
a:      root filesystem (/)
b:      swap space
e,f...: other filesystems
```

All "large" parts of the filesystem should be separate from the root, so that the root itself remains small (less likely to get corrupted). This means at least /usr and /var, and possibly also /home if you have user accounts. The convention we have used is to create a resonable sized /var partition (for variable data like logs), then we have put all remaining disk space in a partition called /usr, and put home directories under that (/usr/home/*username*)

Distribution sets

FreeBSD comes in several parts:

- "distribution sets" are the parts of the base system. The "bin" distribution is the minimum you can install. However there are additional distributions such as "manpages" and "XF86336".
- "ports" and "packages" are third-party software, not really part of FreeBSD itself, but supplied alongside it. "ports" build from source code, whilst "packages" are ports which have already been compiled and are ready to install.

Quick Installation Guide (Using CD-ROM)

For your reference, here is a sequence of operations you could follow to install the classroom workstations. Obviously you should adjust this as required to suit your needs (for example, most servers do not need to have X installed)

Insert boot floppy, change to root floppy when prompted
Skip kernel config
Express install
Delete any existing partitions, then select "Entire disk"
Say Yes to standard partition entry
Select BootMgr

Create partition; ctrl-U to delete number presented, enter "100m" instead
  FS   /
Create partition; ctrl-U; 100m
  Swap
Create partition; ctrl-U; 400m
  FS   /var
Create partition; ctrl-U; 400m
  FS   /usr
Create partition; hit enter to accept number given (i.e. rest of disk)
  FS   /u

X-User (must hit SPACEBAR, not Enter, to select it)
Yes install ports collection

Default answers to remaining questions (i.e. just hit Enter)
WAIT for install to complete
No extra options after install
Exit install
Reboot (remember to remove floppy and CD)
Login as root
halt

Label machine as being successfully installed.

The FreeBSD Directory Structure

Below is a list of some of the directories used under FreeBSD. This information is taken verbatim from http://www.freebsd.org. For a complete description of the FreeBSD directory structure you can type `man hier` at the prompt.

The FreeBSD directory hierarchy is fundamental to obtaining an overall understanding of the system. The most important concept to grasp is that of the root directory, ‘‘/’’. This directory is the first one mounted at boot time and it contains the base system necessary to prepare the operating system for multi-user operation. The root directory also contains mount points for every other file system that you may want to mount.

A mount point is a directory where additional file systems can be grafted onto the root file system. Standard mount points include `/usr`, `/var`, `/mnt`, and `/cdrom`. These directories are usually referenced to entries in the file `/etc/fstab`. `/etc/fstab` is a table of various file systems and mount points for reference by the system.

Most of the file systems in /etc/fstab are mounted automatically at boot time from the script rc(8) unless they contain the noauto option. Consult the fstab(5) manual page for more information on the format of the /etc/fstab file and the options it contains.

A complete description of the filesystem hierarchy is available in hier(7). For now, a brief overview of the most common directories will suffice.

Directory	Description
/	Root directory of the filesystem.
/bin/	User utilities fundamental to both single-user and multi-user environments.
/boot/	Programs and configuration files used during operating system bootstrap.
/boot/defaults/	Default bootstrapping configuration files; see loader.conf(5).
/dev/	Device nodes; see intro(4).
/etc/	System configuration files and scripts.
/etc/defaults/	Default system configuration files; see rc(8).
/etc/mail/	Configuration files for mail transport agents such as sendmail(8).
/etc/namedb/	named configuration files; see named(8).
/etc/periodic/	Scripts that are run daily, weekly, and monthly, via cron(8); see periodic(8).
/etc/ppp/	ppp configuration files; see ppp(8).
/mnt/	Empty directory commonly used by system administrators as a temporary mount point.
/proc/	Process file system; see procfs(5), mount_procfs(8).
/root/	Home directory for the root account.
/sbin/	System programs and administration utilities fundamental to both single-user and multi-user environments.
/stand/	Programs used in a standalone environment.
/tmp/	Temporary files, usually a mfs(8) memory-based filesystem (the contents of /tmp are usually NOT preserved across a system reboot).
/usr/	The majority of user utilities and applications.
/usr/bin/	Common utilities, programming tools, and applications.
/usr/include/	Standard C include files.
/usr/lib/	Archive libraries.
/usr/libdata/	Miscellaneous utility data files.
/usr/libexec/	System daemons & system utilities (executed by other programs).
/usr/local/	Local executables, libraries, etc. Also used as the default destination for the FreeBSD ports framework. Within /usr/local, the general layout sketched out by hier(7) for /usr should be used. Exceptions are the man directory is directly under /usr/local rather than under /usr/local/share. Ports documentation is in share/doc/*port*.
/usr/obj/	Architecture-specific target tree produced by building the /usr/src tree.
/usr/ports	The FreeBSD ports collection (optional).
/usr/sbin/	System daemons & system utilities (executed by users).
/usr/share/	Architecture-independent files.
/usr/src/	BSD and/or local source files.
/usr/X11R6/	X11R6 distribution executables, libraries, etc (optional).
/var/	Multi-purpose log, temporary, transient, and spool files.
/var/log/	Miscellaneous system log files.
/var/mail/	User mailbox files.
/var/spool/	Miscellaneous printer and mail system spooling directories.
/var/tmp/	Temporary files that are kept between system reboots.
/var/yp	NIS maps.

A Few differences from Linux

Other key differences between Linux and FreeBSD center around the naming of devices and some of the configuration files:

Linux: eth0 = first ethernet device (of any type)
FreeBSD: ed0 = first NE2000 device, ep0 = first 3Com 3c509, etc.

Linux: COM1 serial port = /dev/ttyS0
FreeBSD: COM1 serial port = /dev/cuaa0 (call out) or /dev/ttyd0 (call in)

Linux: /etc/inittab configures incoming serial connections
FreeBSD: /etc/ttys configures incoming serial connections

Back to Top

Hervey Allen
Joel Jaeggli
Liberal borrowing from *Brian Candler*
May 5, 2002