An *Open*BSD reference sheet

Author: Stefan Pettersson, last updated: 2010-04-04, latest version at http://www.bigpointyteeth.se/.

I. Initial configuration

1 Partitioning. TBD (remember to fix space for ROOTBACKUP).

2 Create a new administrative user account [18], switch to it and use **su**(1) instead.

```
$ /usr/bin/su -l root
```

3 Create a backup of the GENERIC kernel. [69]

```
# cp /bsd /bsd.GENERIC
# chflags schg /bsd.GENERIC
```

4 Set the local host name. [51]

5 List available keyboard maps and set the appropriate keyboard encoding.

```
# kbd -l
# wsconsctl keyboard.encoding=enc
# echo enc > /etc/kbdtype
# echo keyboard.encoding=enc >> \
> /etc/wsconsctl.conf
```

6 Configure networking.

7 Set correct time zone and adjust the system clock according to the NTP Pool Time Servers. Do this early since the logs will get messed up.

```
# ln -s /etc/localtime \
> /usr/share/zoneinfo/Europe/Stockholm
# rdate -n pool.ntp.org
```

8 Activate OpenNTPD to keep the clock synchronized.

```
# echo servers pool.ntp.org > /etc/ntpd.conf
# ntpd -n
# echo ntpd_flags=\"-s\" >> /etc/rc.conf.local
# ntpd
```

9 Find a good FTP mirror [90] and add it as an exported environment variable to the shell's startup file.

```
FTPMIRROR=ftp://ftp.../pub/OpenBSD/$(uname -r)
```

10 Download source code [64] and the ports collection [56].

11 Apply new errata patches.

12 Update the **locate**(1) database.

```
# /usr/libexec/locate.updatedb
```

13 Modify the messages in *issue*, *issue.net*, *motd*(5), *ftpwelcome*, *nologin.txt* and *adduser.message* under */etc*. Also, update the files in */etc/skel*.

14 Adjust intervals and coverage for log rotation in */etc/newsyslog.conf* [76].

15 Put some configuration files into version control. Read **rcs**(1) for more information.

```
# cd /etc
# mkdir RCS
# files="pf.conf rc.*.local *syslog.conf kshrc*"
# for f in $files; do ci -u -minit -t-$f $f; done
```

16 Activate ROOTBACKUP. Insert a USB memory stick (monitor */var/log/messages* to identify device), partition, format and mount it. Then activate the root backup system in *crontab*(5) and *fstab*(5).

```
# fdisk -i sd4
# newfs sd4c
# mount /dev/sd0c /altroot
# crontab -e   # add ROOTBACKUP=1
# echo "/dev/sd0c /altroot ffs xx 0 0" \
> >> /etc/fstab
```

17 Follow the advice given in the first "daily insecurity output" mail recieved the day after installation.

II. User management

18 View standard values for user creation. Modify *usermgmt.conf*(5) if necessary.

```
# useradd -D
```

19 Create a new standard user account.

```
# useradd -m -L default -g users -c "gecos" user
# passwd user
```

20 Create a new administrative user account.

```
# useradd -m -L staff -g staff -G wheel \
> -c "gecos" user
# passwd user
```

21 Create a new group and add a user to it.

```
# groupadd group
# usermod -G group user
```

22 Pre-generate a password hash and create user.

```
# encrypt -p
Password:
$2a$06$...
# useradd -m -p '$2a$06$...' user
```

23 Add a user interactively or manually, respectively.

```
# adduser
# vipw
```

24 Create a temporary user account.

```
# useradd -m -L default -g users \
> -c "gecos" -e "jul 31 2009" user
# passwd user
```

25 Prevent a user from logging in (unset password and set shell to **nologin**(8)).

```
# userdel -p user
```

26 Lock all user accounts. Accounts belonging to the staff class are unaffected.

```
# echo "Please come back later." > /etc/nologin
```

27 Delete a user's account details from the password and group files (keep home, crontab, spool, etc).

```
# userdel user
```

28 Remove a user account completely (including mail, crontab, home directory, etc).

```
# rmuser user
```

29 View information on a user account.

```
# userinfo user
# finger user
```

30 Change a user's shell.

```
# chsh -s /bin/csh user
```

31 To give a user a trivial password, give the same password several times although **passwd**(1) complains. It will accept any password after three tries.

32 Check users home directory storage usage.

```
# du -sh /home/*
```

III. Package management

You might want to run **script**(1) before installing or removing a package so that the output can be reviewed properly for errors. Software installed from ports will become packages so they are handled in the same way.

```
# script pkg-install.txt
```

33 Add a package mirror environment variable to the shell's startup file and export it. [7]

```
PKG_PATH=$FTPMIRROR/packages/$(machine -a)/
```

34 Download the package list as a plain text file.

```
$ ftp -o pkg_index.txt $PKG_PATH/index.txt
```

35 View the package list in HTML with descriptions.

```
$ lynx http://www.openbsd.org/\
> $(uname -r)_packages/$(machine -a).html
```

36 Search for pattern among package names.

```
# pkg_info -Q pattern
```

37 List currently installed packages (also */var/db/pkg/*).

```
$ pkg_info -a
```

38 List all packages that no other package depends on.

```
$ pkg_info -t
```

39 Find partially installed packages (remove those).

```
$ pkg_info -a | grep "^partial-"
```

40 View the comment, dependants, description and flavors of a specific package.

```
$ pkg_info package
```

41 List files included in a package.

```
$ pkg_info -L package
```

42 Install a package. Use -n to simulate.

```
# pkg_add -v package
```

43 Upgrade a package from an earlier release.

```
# pkg_add -uv package
```

44 Try to upgrade all installed packages.

```
# pkg_add -uv
```

45 Remove an installed package. Use -n to simulate.

```
# pkg_delete -v package
```

IV. Networking

46 Get dynamic network configuration via DHCP.

```
# dhclient if
```

47 Get dynamic network configuration during boot. Check *hostname.if*(5) for more information.

```
# echo "dhcp NONE NONE NONE description foo" >|\
> /etc/hostname.if
```

48 Set static network configuration.

```
# ifconfig if 10.0.0.10 netmask 255.255.255.0 up
# route add default gw 10.0.0.1
```

49 Set static network configuration during boot.

```
# echo "inet 10.0.0.1 255.255.255.0 10.0.0.255 \
> description foo" >| /etc/hostname.if
```

50 Specify a default gateway (for boot setup).

```
# route add default gw 10.0.0.1
# echo 10.0.0.1 >| /etc/mygate
```

51 Specify primary and secondary name servers.

```
# echo domain name.tld >| /etc/resolv.conf
# echo nameserver 10.0.0.2 >> /etc/resolv.conf
# echo nameserver 10.0.0.3 >> /etc/resolv.conf
```

52 Set hostname (for boot setup).

```
# hostname fqdn.name.tld.
# hostname >| /etc/myname
```

53 Check the routing table.

```
$ netstat -rn
$ route show
```

54 List network listeners.

```
$ netstat -an -f inet
$ fstat | grep internet
```

V. Ports collection

Always prefer the pre-compiled packages to ports if you have the choice. Packages will give you less headaches. Ports is, after all, a way to build a package and install it. Remember from the FAQ: "building ports on systems without X11 is not supported". Saving the output from a ports installation with **script**(1) is highly recommended.

```
# script port-install.txt
```

55 Download and unpack the ports collection. [7]

```
$ ftp -o ports.tar.gz $FTPMIRROR/ports.tar.gz
# tar zxf ports.tar.gz -C /usr/
```

56 Update the ports collection over AnonCVS. Set the CVSROOT environment variable and compare the server's public key (if available). Remember to set the proper tag (-r) if you're not on the latest *-release*. [89]

```
# export CVSROOT=anoncvs@...:/cvs
# cd /usr
# cvs -q update -P -rOPENBSD_4_4 ports
```

57 Browse through a detailed index of ports.

```
$ cd /usr/ports; make print-index | less
```

58 Search for pattern in ports .

```
$ cd /usr/ports; make search key=pattern
```

59 Create a hyperlinked set of HTML files describing the ports collection. It will take a while.

```
# cd /usr/ports; make readmes
```

60 Software installed from ports are removed in the same way packages are. [42]

61 Ports installation step-by-step. Every step implies all the preceding ones. Thus, make install will do suffice.

```
# cd /usr/ports/.../...
# make fetch
# make checksum
# make depends
# make extract
```

```
# make patch
# make configure
# make build
# make fake
# make package
# make install
```

VI. Patching

62 Download OpenBSD *-release* source code. [7]

```
# ftp -o src.tar.gz $FTPMIRROR/src.tar.gz
# ftp -o sys.tar.gz $FTPMIRROR/sys.tar.gz
# mkdir /usr/src
# tar zxf src.tar.gz -C /usr/src
# tar zxf sys.tar.gz -C /usr/src
```

63 Updating to *-stable* (the "patch branch"), assuming that the *-release* code is available. Remember to set the proper tag (-r) if you're not on the latest *-release*. [90]

```
# export CVSROOT=anoncvs@...:/cvs
# cd /usr
# cvs -q update -P -rOPENBSD_4_4 src
```

64 Applying errata patches. Follow instructions in the patch files found at http://www.openbsd.org/errata.html.

65 Build, install and boot a new kernel. The old kernel will be available under the name obsd.

```
# arch=$(machine -a)
# cd /usr/src/sys/arch/$arch/conf
# /usr/sbin/config GENERIC
# cd /usr/src/sys/arch/$arch/compile/GENERIC
# make clean && make depend && make
# cd /usr/src/sys/arch/$arch/compile/GENERIC
# make install
# reboot
```

66 Rebuild system binaries. Might take a while.

```
# rm -rf /usr/obj/*
# cd /usr/src
# make obj
# cd /usr/src/etc
# env DESTDIR=/ make distrib-dirs
# cd /usr/src
# make build
```

VII. References

67 Flags for **chflags**(1), use "no"-prefix to remove.

```
sappnd system append-only flag (root only)
schg   system immutable flag (root only)
uappnd user append-only flag (owner or root only)
uchg   user immutable flag (owner or root only)
```

68 Some fields in **ps**(1) for the -l, -u and -v options.

```
VSZ     virtual size in kb
RSS     real memory size in 1024 byte units
TT      controlling terminal or "co" for console
STAT    process state
  D     in disk (uninterruptible) wait
  I     is idle (sleep > 20 seconds)
  R     a runnable process
  S     sleeping for < about 20 seconds
  T     a stopped process
  Z     a dead process (a "zombie")
  s     a session leader
  +     foreground of its terminal
STARTED the time the process was started
TIME    accumulated user + system cpu time
COMMAND command and arguments
PRI     scheduling priority
NI      process scheduling increment (nice)
WCHAN   event on which the process waits
SL      sleep time in seconds
RE      core residency time
PAGEIN  total page faults
LIM     soft limit on memory use
TSIZ    text size in kb
```

69 Routing table flags in **netstat**(1).

```
B just discard packets (during updates)
C generate new routes on use
c cloned routes (generated from RTF_CLONING)
D created dynamically (by redirect)
G destination requires forwarding by intermediary
H host entry (net otherwise)
L valid protocol to link address translation
M modified dynamically (by redirect)
R host or net unreachable
S manually added
U route usable
X external daemon translates proto to link address
```

70 Network pseudo-devices supported by **ifconfig**(8). Their respective manual page is in section 4.

```
bridge Ethernet bridge interface
carp   Common Address Redundancy Protocol
gif    generic tunnel interface
gre    encapsulating network device
lo     software loopback network interface
mpe    MPLS Provider Edge
pflog  packet filter logging interface
ppp    point to point protocol network interface
pppoe  PPP Over Ethernet protocol interface
sl     slip network interface
trunk  link aggregation and failover interface
tun    network tunnel pseudo-device
vlan   IEEE 802.1Q en-/decapsulation pseudo-device
```

71 Hard drive device types.

```
cd   CD-ROM block device, IDE or SCSI
rcd  raw CD-ROM device, IDE or SCSI
fd   floppy block device
rfd  raw floppy device
sd   SCSI block device
rsd  raw SCSI device
wd   IDE block device
rwd  raw IDE device
```

72 File system types supported by *fstab*(5).

```
cd9660 ISO 9660 CD-ROM filesystem
ext2fs local Linux compatible ext2fs filesystem
ffs    local UNIX filesystem
mfs    local memory-based UNIX filesystem
msdos  MS-DOS FAT filesystem
nfs    Sun compatible Network File System
ntfs   NTFS filesystem
procfs local filesystem with process information
swap   disk partition to be used for swapping
udf    UDF filesystem
vnd    VND image file
xx     mount point for ROOTBACKUP
```

73 The schedule time format for *crontab*(5).

```
minute        * or 0-59
hour          * or 0-23
day-of-month  * or 1-31
month         * or 1-12
day-of-week   * or 0-7 (0 or 7 is Sunday)

@reboot    once, at cron(8) startup
@yearly    every January 1, "0 0 1 1 *"
@annually  (same as @yearly)
@monthly   first day of every month, "0 0 1 * *"
@weekly    every Sunday, "0 0 * * 0"
@daily     every midnight, "0 0 * * *"
@midnight  (same as @daily)
@hourly    every hour, on the hour, "0 * * * *"
```

74 Examples of **calendar**(1) entries.

75 Avaliable **syslogd**(8) priorities in order.

```
emerg   panic, normally broadcast to all users
alert   condition to be corrected immediately
crit    critical e.g. hard device errors
err     errors
warning warning messages
notice  not error but should be handled specially
info    informational messages
debug   messages only of use when debugging
```

76 Available **syslogd**(8) facilities.

```
none     do not log anything
auth     authorization login(1), su(1), getty(3)
authpriv same as auth but only readable by root
cron     cron daemon, cron(3)
daemon   daemons lacking their own facility
ftp      file transfer protocol daemon
kern     messages generated by the kernel
lpr      printer spooling: lpr(1), lpc(3), lpd(3)
mail     mail system
news     network news system
syslog   messages generated by syslogd(3)
user     user processes, default if none specified
uucp     UUCP system.
local0-7 reserved for local use
```

77 Flags used by **newsyslog**(8).

```
Z  compress file
B  file is binary, no log rotate message
M  this is a monitored file `
F  symbolic links should be followed
```

VIII. Miscellaneous actions

78 Boot in single-user mode with the backup kernel, check and mount disks and activate networking.

```
boot> boot /bsd.GENERIC -s
# export TERM=vt220
# fsck -p
# mount -a
# /bin/sh /etc/netstart
```

79 List disk devices recognized by OpenBSD and check their partition scheme and disklabels. See [71] and [72].

```
# sysctl hw.disknames
hw.disknames=wd0,wd1,cd0,sd0,sd1,sd2,sd3
# fdisk wd1
# disklabel wd1
```

80 Mount an *n* Mb memory partition from swap.

```
# mount_mfs -s <2048n> /dev/wd0b /mnt
# echo "/dev/wd0b /mnt mfs rw,async <2048n> 0 0"\
> >> /etc/fstab
```

81 Create a 64 Mb encrypted file system image.

```
# dd if=/dev/zero of=enc.img bs=1024 count=65536
# vnconfig -ck /dev/svnd0c enc.img
# disklabel -E /dev/svnd0c
# newfs /dev/rsvnd0c
# mount /dev/svnd0c /mnt
# umount /mnt
# vnconfig -u /svnd0c
```

82 Reset a forgotten root password.

```
boot> boot -s
# fsck -p / && mount -uw /
# fsck -p /usr && mount /usr
# passwd
# reboot
```

83 Get very verbose boot messages.

```
boot> boot -c
UKC> verbose
UKC> quit
```

84 Configure kernel settings until next reboot. Permanent changes must be saved to *sysctl.conf*(5).

```
# man 8 sysctl
# sysctl net.inet.ip.ttl
net.inet.ip.ttl=64
# sysctl net.inet.ip.ttl=255
net.inet.ip.ttl: 64 -> 255
```

IX. Security

85 Comment unneeded services from */etc/inetd.conf*. If none are needed, consider disabling **inetd**(8) altogether.

```
# echo "inetd=NO" >> /etc/rc.conf.local
```

86 Increase securelevel(7).

```
# sysctl kern.securelevel=2
```

87 Encrypt swap space (default in OpenBSD > 4.5).

```
# sysctl vm.swapencrypt.enable=1
```

88 Implement W^X for filesystems in *fstab*(5).

```
/dev/wd0a /     ffs ro 1 1
/dev/wd0h /home ffs rw,nodev,nosuid,noexec 1 2
/dev/wd0d /tmp  ffs rw,nodev,nosuid,noexec 1 2
/dev/wd0g /usr  ffs ro,nodev 1 2
/dev/wd0e /var  ffs rw,nodev,nosuid,noexec 1 2
```

89 Increase the password restrictions in *login.conf*(5).

```
minpasswordlen=12
passwordtime=7776000
passwordtries=0
```

X. Online references

90 http://www.openbsd.org/anoncvs.html#CVSROOT

91 http://www.openbsd.org/ftp.html#ftp

92 http://www.openbsd.org/errata.html

XI. License information