OpenBSD 3.4 Installation Guide

A Quick Guide to install OpenBSD 3.4.
This guide is designed for peoples who are willing to move from Windows To OpenBSD.
Created by Adonis aKa NtWaK0

GNU Free Documentation License

Version 1.01, Marc3/20/2004 (LIVE DOCUMENT)

Read http://www.deadly.org/article.php3?sid=20040310124435&mode=flat (Bye Bye Windows Welcome PUFFY)

OpenBSD 3.4

Guide to easy Installation

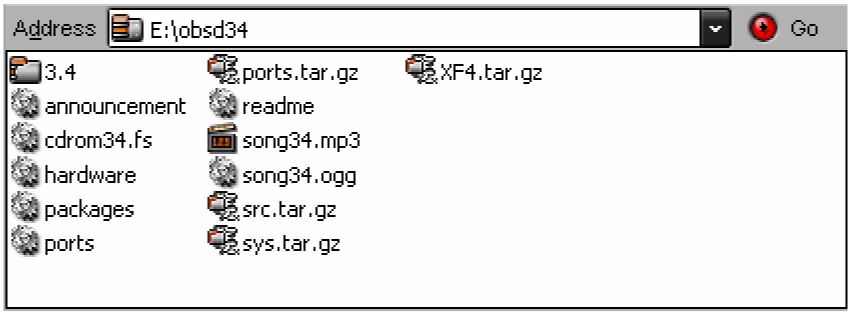
Creating a Custom ISO file

I encourage you to buy (http://www.openbsd.org/orders.html) the CD so you get the stickers and the original CD print. If you cannot afford buying it, I am going to show you how you can create your own ISO file.

Note: Change all UPPERCASE files to lowercase including the directory I386 it should be i386

I suggest you following the following structure. Grab these files from:

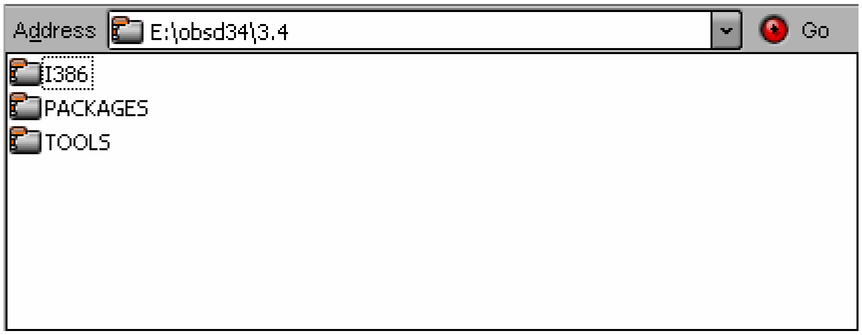
ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/



E:\obsd34

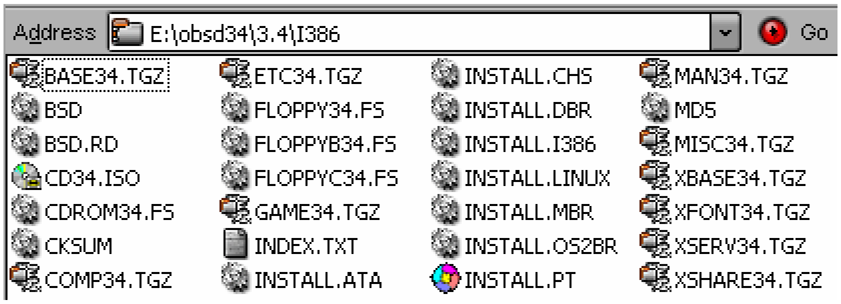
```
3.4             ports.tar.gz    XF4.tar.gz
announcement    readme
cdrom34.fs      song34.mp3
hardware        song34.ogg
packages        src.tar.gz
ports           sys.tar.gz
```

In the package directory put whatever package you like.



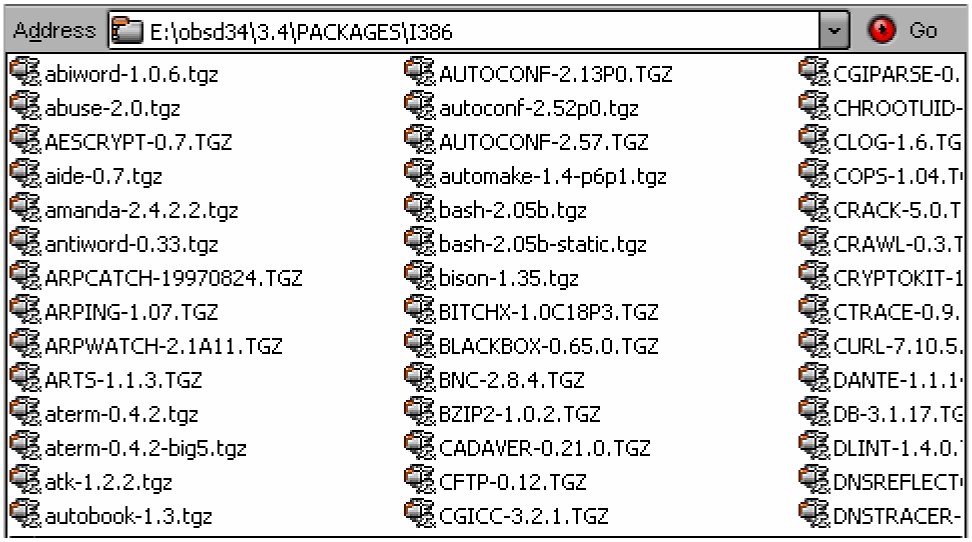
E:\obsd34\3.4

```
I386
PACKAGES
TOOLS
```



E:\obsd34\3.4\I386

BASE34.TGZ
BSD
BSD.RD
CD34.ISO
CDROM34.FS
CKSUM
COMP34.TGZ
ETC34.TGZ
FLOPPY34.FS
FLOPPYB34.FS
FLOPPYC34.FS
GAME34.TGZ
INDEX.TXT
INSTALL.ATA
INSTALL.CHS
INSTALL.DBR
INSTALL.I386
INSTALL.LINUX
INSTALL.MBR
INSTALL.OS2BR
INSTALL.PT
MAN34.TGZ
MD5
MISC34.TGZ
XBASE34.TGZ
XFONT34.TGZ
XSERV34.TGZ
XSHARE34.TGZ



E:\obsd34\3.4\PACKAGES\I386

abiword-1.0.6.tgz
abuse-2.0.tgz
AESCRYPT-0.7.TGZ
aide-0.7.tgz
amanda-2.4.2.2.tgz
antiword-0.33.tgz
ARPCATCH-19970824.TGZ
ARPING-1.07.TGZ
ARPWATCH-2.1A11.TGZ
ARTS-1.1.3.TGZ
aterm-0.4.2.tgz
aterm-0.4.2-big5.tgz
atk-1.2.2.tgz
autobook-1.3.tgz
AUTOCONF-2.13P0.TGZ
autoconf-2.52p0.tgz
AUTOCONF-2.57.TGZ
automake-1.4-p6p1.tgz
bash-2.05b.tgz
bash-2.05b-static.tgz
bison-1.35.tgz
BITCHX-1.0C18P3.TGZ
BLACKBOX-0.65.0.TGZ
BNC-2.8.4.TGZ
BZIP2-1.0.2.TGZ
CADAVER-0.21.0.TGZ
CFTP-0.12.TGZ
CGICC-3.2.1.TGZ
CGIPARSE-0.
CHROOTUID-
CLOG-1.6.TG
COPS-1.04.T
CRACK-5.0.T
CRAWL-0.3.T
CRYPTOKIT-1
CTRACE-0.9.
CURL-7.10.5.
DANTE-1.1.1
DB-3.1.17.TG
DLINT-1.4.0.
DNSREFLECT
DNSTRACER-

Get ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/ports.tar.gz save to <drive>:\obsd\
Get ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/src.tar.gz save to <drive>:\obsd\
Get ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/sys.tar.gz save to <drive>:\obsd\
Get ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/XF4.tar.gz save to <drive>:\obsd\

Get the package you want from ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/packages/ and save them to <drive>:\ obsd34\3.4\packages. Note your CD can contain 700 MB.

Creating the ISO file using mkisofs.exe

Get this tool http://openbsddiary.org/download/cdrtools-1.11a12-win32-bin.zip

Unzip the cdrtools-1.11a12-win32-bin.zip file into <drive>:\somedirectory\. You only need mkisofs.exe located inside the zip file.

mkisofs -o C:\temp\obsd34.iso -R -J -V "openBSD 3.4 NtWaK0 Custom" -b 3.4/I386/CDROM34.FS -c "boot.catalog" -A "NtWaK0 at Safehack.com OpenBSD 3.4 CD" .

NOTE: The dot. at the end is needed.

To burn your ISO under UNIX use:

# cdrecord -v speed=4 dev=/dev/rcd0c:x,x,x -data OpenBSD.iso

To burn your ISO under Windows you can use Nero or Fireburn or any other CD burning software.

You are ready to BOOT from that CD and start your OpenBSD 3.4 Installation. Good Luck

Creating a Boot Floppy To Install OpenBSD From

In case you have an old computer like me you need to boot using a floppy.

To prepare a floppy in MS-DOS or Windows, first use the native formatting tools to format the disk.

To write the installation image to the prepared floppy you can use rawrite, fdimage, or ntrw. rawrite will not work on Windows NT, 2000 or XP.

```
C:\> rewrite
Enter source file name: floppy34.fs
Enter destination drive: a

Example usage of fdimage:
C:\> fdimage -q floppy34.fs a:

Example usage of ntrw: I use this method my self to create a boot
diskette.
C:\> ntrw floppy34.fs a:
I like to use ntrw.
```

Starting the Installation

[Snip: http://www.openbsd.org/faq/faq4.html#DosFlop]

You can terminate the current install attempt by hitting **CTRL-C** and can restart it without rebooting by running **# install** at the shell prompt.

dmesg text is saved as /var/run/dmesg.boot.

```
  rootdev=0x1100 rrootdev=0x2f00 rawdev=0x2f02
  erase ^?, werase ^W, kill ^U, intr ^C, status ^T
  (I)nstall, (U)pgrade or (S)hell? i
```

```
  Welcome to the OpenBSD/i386 3.4 install program.

  This program will help you install OpenBSD in a simple and rational way. At
  any prompt except password prompts you can run a shell command by typing
  '!foo', or escape to a shell by typing '!'. Default answers are shown in []'s
  and are selected by pressing RETURN. At any time you can exit this program by
  pressing Control-C and then RETURN, but quitting during an install can leave
  your system in an inconsistent state.

  Specify terminal type: [vt220]
  Do you wish to select a keyboard encoding table? [n] ENTER
```

```
  IS YOUR DATA BACKED UP? As with anything that modifies disk contents, this
  program can cause SIGNIFICANT data loss.

  It is often helpful to have the installation notes handy. For complex disk
  configurations, relevant disk hardware manuals and a calculator are useful.

  Proceed with install? [n] y
```

```
  Cool!  Let's get to it...

  You will now initialize the disk(s) that OpenBSD will use. To enable all
  available security features you should configure the disk(s) to allow the
  creation of separate filesystems for /, /tmp, /var, /usr, and /home.

  Available disks are: wd0.
  Which one is the root disk? (or done) [wd0] Enter
```

```
  Do you want to use *all* of wd0 for OpenBSD? [no] yes
This will use all the available disk space. If you Hit Enter or answered no
fdisk will start and you have to create MBR partition.
```

Do you want to use *all* of wd0 for OpenBSD? [no] yes. If you say "yes" to this question, the entire disk will be allocated to OpenBSD. For windows users who are moving to OpenBSD I suggest you answering yes. Once you get your hand wet in OpenBSD you can re-install it and use whatever disk partition you wish.

Creating a disklabel

The next step is to use disklabel(8) to slice up the OpenBSD partition.

- p - displays (prints) the current disklabel to the screen, and you can use the modifiers k, m or g for kilobytes, megabytes or gigabytes.
- D - Clears any existing disklabel, creates a new default disklabel which covers just the current OpenBSD partition. This can be useful if the disk previously had a

disklabel on it, and the OpenBSD partition was recreated to a different size -- the old disk label may not get deleted, and may cause confusion.

- m - Modifies an existing entry in a disklabel.

If we have a disk with 10 GB we will create **(/, /swap, /tmp, /var, /home, /usr)**:

wd0a: / - 200M. Should be more than enough
wd0b: /swap - 500M. Ram * 2
wd0d: /tmp - 300M. /tmp
wd0e: /var - 2G or 4G if you have more space and you are planning to run a web server or an IDS you need more space.
wd0g: /home - 1G
wd0h: /usr - 6G. This will allow plenty of user file space.

Set the system hostname. This value, along with the **DNS domain name**, will be saved in the file **/etc/myname**.

```
Enter system hostname (short form, e.g. 'foo'): pubweb
```

Now it is time to configure your network.

```
Configure the network? [y] Enter
Available interfaces are: fxp0.
Which one do you wish to initialize? (or 'done') [fxp0] Enter
Symbolic (host) name for fxp0? [pubweb] Enter
The default media for fxp0 is
        media: Ethernet autoselect (100baseTX full-duplex)
Do you want to change the default media? [n] Enter
IP address for fxp0? (or 'dhcp') 111.111.111.111
Netmask? [255.255.255.0] Enter
Done - no available interfaces found.
DNS domain name? (e.g. 'bar.com') [my.domain] AdonisaKaNtWaK0.com
DNS nameserver? (IP address or 'none') [none] 111.111.111.2
Use the nameserver now? [y] Enter
Default route? (IP address, 'dhcp' or 'none') 111.111.111.1
add net default: gateway 111.111.111.1
Edit hosts with ed? [n] Enter
Do you want to do any manual network configuration? [n] Enter
```

In this example, I have used a static IP address.

Here is a sample of the network configuration part of the install, this time done with DHCP:

```
Configure the network? [y] Enter
Available interfaces are: fxp0.
Which one do you wish to initialize? (or 'done') [fxp0] Enter
Symbolic (host) name for fxp0? [pubweb] Enter
The default media for fxp0 is
        media: Ethernet autoselect (100baseTX full-duplex)
Do you want to change the default media? [n] Enter
IP address for fxp0? (or 'dhcp') dhcp
---
DNS domain name? (e.g. 'bar.com') [AdonisaKaNtWaK0.com] Enter
DNS nameserver? (IP address or 'none') [111.111.111.2] Enter
Use the nameserver now? [y] Enter
Default route? (IP address, 'dhcp' or 'none') [dhcp] Enter
add net default: gateway 111.111.111.1
Edit hosts with ed? [n] Enter
Do you want to do any manual network configuration? [n] Enter
```

NOTE: **Only one interface can easily be configured using DHCP** during an install. Now, we set the password for the root account:

```
Password for root account? (will not echo) pAssWOrd
Password for root account? (again) pAssWOrd
```

```
  Sets can be located on a (m)ounted filesystem; a (c)drom, (d)isk or (t)ape
  device; or a (f)tp, (n)fs or (h)ttp server.
  Where are the install sets? c
  Available CD-ROMs are: cd0.
```

```
  Available CD-ROMs are: cd0.
  Which one contains the install media? (or 'done') [cd0] Enter

  Pathname to the sets?  (or 'done') [3.4/i386] Enter=
```

```
  The following sets are available. Enter a filename, 'all' to select
  all the sets, or 'done'. You may de-select a set by prepending a '-'
  to its name.

          [X] bsd
          [ ] bsd.rd
          [X] base34.tgz
          [X] etc34.tgz
          [X] misc34.tgz
          [X] comp34.tgz
          [X] man34.tgz
          [ ] game34.tgz
          [X] xbase34.tgz
          [X] xshare34.tgz
          [X] xfont34.tgz
          [X] xserv34.tgz

  File Name? (or 'done') [bsd.rd] -game34.tgz –bsd.rd
  File Name? (or 'done') [bsd.rd] done
```

You can do all kinds of nifty things here -- -x* would remove all X components. In our case we are going to run X. I do not suggest you running X if the host is going to be a server.

```
  File Name? (or 'done') [done] Enter
  Ready to install sets? [y] Enter
  Getting bsd ...
  100% |**************************************************|  4735 KB    00:03
  ---
  Sets can be located on a (m)ounted filesystem; a (c)drom, (d)isk or (t)ape
  device; or a (f)tp, (n)fs or (h)ttp server.
  Where are the install sets? (or 'done')done
```

```
  Do you expect to run the X Window System? [y] y
  Saving configuration files......done.
 ---
  What timezone are you in? ('?' for list) [US/Pacific] Canada/Eastern
 ---
  Setting local timezone to 'US/Eastern'...done.
```

```
  Making all device nodes...done.
  Installing boot block...
  ---

  CONGRATULATIONS! Your OpenBSD install has been successfully completed!
  To boot the new system, enter halt at the command prompt. Once the
  system has halted, reset the machine and boot from the disk.
  # halt
  syncing disks... done

  The operating system has halted.
  Please press any key to reboot.
```

Do this after your first Boot

Errata

By the time that you have installed your system, it is quite likely that bugs in the release have been found. All significant and easily fixed problems will be reported at http://www.openbsd.org/errata.html

NOTE: You do not have to apply errata patch if you are going to grab the src using cvs.

Deny root SSH Login

If you wish to **deny root logins over the network**, **edit the /etc/ssh/sshd_config** file and set PermitRootLogin to ``no''

Note: Be sure to create a user and add it to the wheel group as described below.

Mail Aliases

```
Edit /etc/mail/aliases and set the three standard aliases to go
to either a mailing list, or the system administrator.
# Well-known aliases -- these should be filled in!
           root:           root@localhost
           manager:        root@localhost
           dumper:         root@localhost
Run # newaliases after changes.
$ newaliases
```

Check Local Time Soft link

```
$ ln -fs /usr/share/zoneinfo/Canada/Eastern /etc/localtime
```

Check hostname

Use the **hostname** command to verify that the name of your machine is correct.
You will also need to edit the /etc/myname file to have it stick around for the next reboot.

```
%hostname
```

Check Gateway

```
% cat /etc/mygate
```

Check disk mounts

```
Check that the disks are mounted correctly by comparing the
/etc/fstab file against the output of the mount and df commands.
           # cat /etc/fstab
           # mount
           # df
           # pstat -s
```

Change user logon information

```
# chpass root
```

Creating a user with root rights

The easiest way to add a user in OpenBSD is to use the **adduser**.

You can edit **/etc/adduser.conf**.

Adduser allows for consistency checks on /etc/passwd, /etc/group, and shell databases. It will create the entries and $HOME directories for you. It can even send a message to the user welcoming them. Here is an example user, testuser being added to a system

```
# adduser
```

Settings are located in /etc/usermgmt.conf

Below are a few commands you can use to quickly check that everything was created correctly.

```
$ ls -la /home
$ id testuser
uid=1002(testuser) gid=31(guest) groups=31(guest)
```

If you want a user to be able to use the root password add the user to wheel group.

You can add your own variables by editing /usr/sbin/adduser

This variable controls the directory where user's home directories are located

```
path=('/bin','/usr/bin','/usr/local/bin')
```

This contains the list of directories that contain legitimate shells.

```
shellpref=('csh','sh','ksh','nologin')
```

This is a list of legitimate shells. Adduser will let you choose from any of these when creating a new user.

Adding users non interactively

```
# adduser -batch OBSDWaK0 wheel 'Ado' passwordhere
```

Check nameserver (DNS client)

```
$ cat resolv.conf
domain safehack.com
nameserver 123.134.123.111
lookup file bind

In this example your domain servers are 125.2.3.4 and 125.2.3.5. You
also belong in the domain "example.com".
$ cat /etc/resolv.conf
search example.com
nameserver 125.2.3.4
nameserver 125.2.3.5
lookup file bind

You can either reboot or run the /etc/netstart script. You can do
this by simply typing (as root). To test if the DNS is working type
uname or uname -a and try to ping another host.
```

Disable RPC-based network services

`We wont be running any NFS or YP we will make sure` **`portmap=NO in /etc/rc.conf`**`.local`

A good approach is to never touch /etc/rc.conf itself. Instead, create the file /etc/rc.conf.local, copy just the lines you are about to change from /etc/rc.conf and adjust them as you like.

Installing src :: ports :: Updating System

After you're done with the installation **reboot and start installing the source code**.

Once you have obtained a source tree via anoncvs, you must rebuild the system. The stages for doing so are:

- <-> Update the src, ports, sys
- <-> Make the appropriate change to move from 3.4 to current
- <-> Rebuild the kernel
- <-> Reboot with the new kernel
- <-> Rebuild the binaries

I suggest getting the needed file and burning them on the same OpenBSD 3.4 installation CDROM.

ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/src.tar.gz
ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/sys.tar.gz
ftp://ftp.ca.openbsd.org/pub/OpenBSD/3.4/ports.tar.gz

Installing the source code

```
#!/bin/sh
Echo ”Will we set the system date”
rdate -ncv time.nrc.ca
Today="`date`"
echo Today is $Today
echo “ Will copy the src and ports of your CDROM”
echo “ Mounting the CDROM”
mount /dev/cd0c /mnt
echo “ Changing directory to /usr/src/”
cd /usr/src
echo “Uncompressing source and ports”
tar -xzvpf /mnt/src.tar.gz
tar -xzvpf /mnt/sys.tar.gz
cd /usr/
tar -xzvpf /mnt/ports.tar.gz
```

You can use the CVS to checkout the src, this is going to be slow process. I prefere to install the src and ports and sys from a CDROM

```
#cd /usr; cvs checkout -P -rOPENBSD_3_4 src

Today="`date`"
echo Today is $Today
export CVSROOT=anoncvs@anoncvs.ca.openbsd.org:/cvs

cd /usr
cvs -q get -rOPENBSD_3_4 -P src

cd /usr
cvs -q get -rOPENBSD_3_4 -P ports

cd /usr/src/sys/arch/i386/
cvs get src/sys/arch/i386

#cd /usr; cvs checkout -P -rOPENBSD_3_4 src
#cd /usr; cvs checkout -P -rOPENBSD_3_4 ports
```

X11 Source tree

There are two ways to get the XF4 sources to /usr/XF4:

copy the tree off the CD (assuming the 3rd CD is mounted on /mnt):

```
        # cd /mnt; cp -Rp XF4 /usr
```

use a union mount with the CD below a writable directory.

```
        # mount -t union -o -b /mnt/XF4 /usr/XF4
```

After this, /usr/XF4 will be ready to be used by cvs. You can for example update it to -current source (assuming you've already set the CVSROOT environment variable):

```
        # cd /usr/XF4
        # cvs -q update -Pd
```

http://www.openbsd.org/anoncvs.html

Move from 3.4 to CURRENT. This need to be done ONLY once

http://www.openbsd.org/faq/upgrade-minifaq.html

```
#!/bin/sh

rdate -ncv time.nrc.ca
Today="`date`"
echo Today is $Today
echo "Building your new kernel please wait..."
echo "Cleaning out the old object files."
cd /usr/src
find . -type l -name obj | xargs rm
make cleandir
rm -rf /usr/obj/*
make obj

echo "New user and group _pflogd (2003/10/23)"
groupadd -g 74 _pflogd
echo "run vipw and add" > dobyhand.txt
echo "_pflogd:*:74:74::0:0:pflogd
privsep:/var/empty:/sbin/nologin" >> dobyhand.txt

echo "New user and group _bgpd (2003/12/17)"
groupadd -g 75 _bgpd
echo "run vipw and add" >> dobyhand.txt
echo "_bgpd:*:75:75::0:0:BGP Daemon:/var/empty:/sbin/nologin" >>
dobyhand.txt


echo "New version of join(1) (2003/12/28)"
cp /usr/src/etc/security /etc
```

```
echo "sudo build change (2004/01/15)"
rm -f /usr/obj/usr.bin/sudo/*

echo "New user and group _tcpdump (2004/01/19)"
groupadd -g 76 _tcpdump
echo "run vipw and add" >> dobyhand.txt
echo "_tcpdump:*:76:76::0:0:tcpdump:/var/empty:/sbin/nologin" >>
dobyhand.txt

echo "Making sure all the appropriate directories are created"
cd /usr/src/etc && env DESTDIR=/ make distrib-dirs

echo "machdep.c change (i386) (2004/02/01)"
cd /usr/src/gnu/usr.bin/binutils
make -f Makefile.bsd-wrapper cleandir
make -f Makefile.bsd-wrapper obj
make -f Makefile.bsd-wrapper depend
make -f Makefile.bsd-wrapper
make -f Makefile.bsd-wrapper install

echo "Compiling a new kernel"
cd /usr/src/sys/arch/i386/conf
config GENERIC
cd ../compile/GENERIC/
make clean && make depend && make
cp /bsd /bsd.old && cp bsd /bsd

echo "Interface cloning (2003/12/07)"
cd /usr/src && make includes
cd sbin/ifconfig
make obj depend
make
make install
cp /usr/src/etc/netstart /etc
echo "reboot"
```

Rebuild the system binaries

```
#!/bin/sh

Today="`date`"
echo Today is $Today
echo "Building your system binaries"
cd /usr/src
make build
echo “reboot”
# We done already make obj so we do not have to rebuild the obj.
```

Update /etc and /dev by hand.

These are not updated automatically. Choose a directory with enough space to hold /, /dev, /var, and /etc. Here I'll use /home/newroot

```
  # mkdir /home/newroot
  # export DESTDIR=/home/newroot
  # cd /usr/src/etc && make distribution-etc-root-var
```

Now compare the files in /home/newroot with their installed counterparts. Replace or update the files as necessary.

```
  # rm -rf /home/newroot   (when done)
```

Reboot to make sure the new /etc files are correct

Tweaking and Applications Installation

Installing tcsh

```
$ cd /usr/ports/shells/tcsh && make install clean
edit /etc/shells and add /usr/local/bin/tcsh
```

Adding Aliases

```
Do this after installing tcsh.
Edit the file Edit ~/.cshrc or ~/.tcshrc and add

set prompt = "-- %T %n %~ -- \n$ "
alias     updb   '/usr/libexec/locate.updatedb'

updb will help you updating the file database used by locate.
```

Changing Console Display

```
This can be done automatically at boot by adding the following lines
to the end of your rc.local(8) file:

wsfontload -h 8 -e ibm /usr/share/misc/pcvtfonts/vt220l.808
wsconscfg -dF 5
wsconscfg -t 80x50 5
```

Activating Apache Web

```
Edit /etc/rc.conf file and change httpd_flags=NO to httpd_flags="-DSSL"
```

A good approach is to never touch /etc/rc.conf itself. Instead, create the file /etc/rc.conf.local, copy just the lines you are about to change from /etc/rc.conf and adjust them as you like.

Setting SSL

To start off, you need to create your server key and certificate using OpenSSL:

```
# openssl genrsa -out /etc/ssl/private/server.key 1024
```

Or, if you wish the key to be encrypted with a passphrase that you will have to type in when starting servers

```
# openssl genrsa -des3 -out /etc/ssl/private/server.key 1024
```

The next step is to generate a Certificate Signing Request which is used to get a Certifying Authority (CA) to sign your certificate. To do this use the command:

```
#openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/private/server.csr
```

This server.csr file can then be given to Certifying Authority who will sign the key. One such CA is Thawte Certification which you can reach at http://www.thawte.com/.

If you cannot afford this, or just want to sign the certificate yourself, you can use the following.

```
#openssl x509 -req -days 365 -in /etc/ssl/private/server.csr -signkey
/etc/ssl/private/server.key -out /etc/ssl/server.crt
```

With /etc/ssl/server.crt and /etc/ssl/private/server.key in place, you should be able to start httpd(8) with the -DSSL flag .

```
#!/bin/sh

Today="`date`"
echo Today is $Today

echo "Activating the Apache Web"
Edit /etc/rc.conf file and change httpd_flags=NO to httpd_flags="-DSSL" >> dobyhand.txt

echo "Setting SSL"
#openssl genrsa -out /etc/ssl/private/server.key 1024
echo "key encrypted with a passphrase that you will have to type to start the service"
openssl genrsa -des3 -out /etc/ssl/private/server.key 1024

echo "Generate a Certificate Signing Request which is used to get"
echo "a Certifying Authority (CA) to sign your certificate."
openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/private/server.csr
echo "This server.csr file can then be given to Certifying Authority"
echo "who will sign the key. One such CA is Thawte Certification"
echo "We will sign the certificate ourself"
openssl x509 -req -days 365 -in /etc/ssl/private/server.csr -signkey \
/etc/ssl/private/server.key -out /etc/ssl/server.crt
echo " With /etc/ssl/server.crt and /etc/ssl/private/server.key in place"
```

```
        echo "you should be able to start httpd with the -DSSL flag"
```

Changing /etc files

We suggest that you cd /etc and edit most of the files in that directory.

Note that the /etc/motd file is modified by /etc/rc whenever the system is booted. To keep any custom message intact, ensure that you leave two blank lines at the top, or your message will be overwritten.

Make a backup copy of all files in /etc/

Concatenated disks (ccd)

If you are using ccd(4) concatenated disks, edit /etc/ccd.conf. Use the ccdconfig -U command to unload and the ccdconfig -C command to create tables internal to the kernel for the concatenated disks. You then mount(8), umount(8), and edit /etc/fstab as needed.

Automounter daemon (AMD)

If using the amd(8) package, go into the /etc/amd directory and set it up by renaming master.sample to master and editing it and creating other maps as needed.

Alternatively, you can get your maps with YP.

System daemon configuration database

The **file /etc/rc.conf** is **used to configure the system daemons**. Administrators should make changes to the default settings by editing a separate file, **/etc/rc.conf.local** (this may need to be created from scratch). Values can then be added in as desired.

Check for any local changes needed in the files /etc/rc.conf, /etc/rc.local,

/etc/rc.securelevel, and /etc/rc.shutdown.

Turning on something like the Network Time Protocol in /etc/rc.conf.local requires making sure the package is installed.

Installing X

When you installed OpenBSD you have the choise to install X if you have done so the next step is to configure your X windows.

Two programs are recommended for creating your XF86Config file: xf86cfg and xf86config.

xf86cfg configuration is confusing if you are not used to, alternately it offers a text-mode configuration '**xf86cfg -textmode**'

I suggest you using XF86Config it help the process if you can connect from another computer 'ssh' and execute the installation sequence so you can more easily review error output, which is logged in /var/log/XFree86.0.log.

```
$ tail -f /var/log/XFree86.0.log
```

A starting point on a new machine with no idea about the video card, is to use XFree86 -configure option.

```
$ XFree86 -configure
Your XF86Config file is /root/XF86Config.new
```

To test the server, run:

```
XFree86 -xf86config /root/XF86Config.new
```

If the graphic screen worked well, use this Configuration as a basis for your X environment.

Copy the /root/XF86Config.new file to the standard location for your machine (usually at /etc/X11)

Note: make backups of any existing files before you copy anything

```
$ cp /etc/X11/XF86Config /etc/X11/XF86Config.org
# cp /root/XF86Config.new /etc/X11/XF86Config
```

Starting:

```
$ /usr/X11R6/bin/startx
```

To start X Window **using 256 color mode**:

```
# startx -- -bbp 8
```

To start X Window in **true colour mode**, use the following command

```
# startx -- -bbp 32
```

'startx' is a script to initialise services for the X Window environment which in the default OpenBSD configuration starts up a simple 'window manager' and a number of 'xterm' connections.

X won't start

If you have X completely set up and you are using an XF86Config that you know works then the problem most likely lies in the machdep.allowaperture. You also need to make sure that:

```
option APERTURE
```

Then you need to edit /etc/sysctl.conf and set machdep.allowaperture=2. This will allow X to access the aperture driver. This would already be set if you said that you would be running X when asked during the install. OpenBSD requires for all X servers that the aperture driver be set, because it controls access to the I/O ports on video boards.

Stopping:

To exit out of X, use Ctrl+Alt+Backspace

Installing KDE

The good fast way to install KDE on OpenBSD 3.4 is to copy all needed packages from CD to usr/ports/packages/i386/All (Another good reason to buy OpenBSD CDROM.

To get the packages go to ftp://ftp.openbsd.org/pub/OpenBSD/3.4/packages/i386/

```
% cd /usr/ports/packages/i386/All
$ ls kde*
kdeartwork-2.2.1.tgz kdebase-ns-2.2.1.tgz kdegraphics-2.2.1.tgz
kdenetwork-2.2.1.tgz
kdebase-2.2.1.tgz kdegames-2.2.1.tgz kdelibs-2.2.1.tgz kdetoys-
2.2.1.tgz
$ ls kof*
koffice-1.1.tgz

$ pkg_add koffice-1.1.tgz kdenetwork-2.2.1.tgz
$ pkg_add kdegraphics* kdegames* kdetoys*
```

http://www.nomoa.com/

Configuring .xinitrc (personal desktop choice)

To allow user customisation, the startx script looks for system wide files and their corresponding version in the user home directory. If the user files are not found then the global version is used, otherwise the local user version is used. In this way, your users can have their own custom visual displays when they start up in X Window.

The **global configuration files are xserverrc and xinitrc**. The **user custom versions are ~/.xserverrc and ~/.xinitrc**

xserverrc or ~/.xserverrc contain details of which X servers to start, while **xinitrc or ~/.xinitrc contain details of which client applications are going to be run** after the X Window system has started. When the script exits, the X server is also terminated.

Create a file: ~/.xinitrc to contain the following startup commands.

```
File: ~/.xinitrc
#!/bin/sh
/usr/local/bin/startkde
```

After creating the above file, then you need to make the file executable by performing the following command as the user.

```
$ cd ~
$ chmod a+x .xinitrc
```

Now, when the user starts X (by using startx) the user will be presented with KDE as their default GUI.

Configuring xinitrc (global desktop choice)

To set the environment so **all desktops started with startx are kde**, make changes to /usr/X11R6/lib/X11/xinit/xinitrc

```
File: xinitrc
# start some nice programs
xclock -geometry 50x50-1+1 &
xconsole &
xterm -geometry 80x24 &
# fvwm || xterm
# /usr/local/bin/startkde
```

Setup a nameserver (DNS client)

To setup your nameservers, you will create or edit the file called /etc/resolv.conf

The first line tells the computer its local domain name. Label the domain name with the "domain' keyword. Nameservers can appear on subsequent lines, each labeled with an IP address.

```
# cat resolv.conf
domain safehack.com
nameserver 123.134.123.111
lookup file bind
```

In this example your domain servers are `125.2.3.4` and `125.2.3.5`. You also belong in the domain "example.com".

```
$ cat /etc/resolv.conf
search example.com
nameserver 125.2.3.4
nameserver 125.2.3.5
lookup file bind
```

You can either reboot or run the /etc/netstart script. You can do this by simply typing (as root). To test if the DNS is working type uname or uname -a and try to ping another host.

BIND Name Server (DNS)

If you are using the BIND Name Server, check the /etc/resolv.conf file.

It may look something like:

```
domain nts.umn.edu
nameserver 128.101.101.101
nameserver 134.84.84.84
search nts.umn.edu. umn.edu.
lookup file bind
```

If using a caching name server, add the line "nameserver 127.0.0.1" first. To get a local caching name server to run you will need to set "named_flags" in /etc/rc.conf.local. The same holds true if the machine is going to be a name server for your domain. In both these cases, make sure that named(8) is running (otherwise there are long waits for resolver timeouts).

DHCP Server

If you want to use OpenBSD as a DHCP server dhcpd(8), **edit /etc/rc.conf**. Set it up so that **dhcpd_flags="-q"** instead of dhcpd_flags=NO.

Put the interfaces that you want dhcpd to listen on in /etc/dhcpd.interfaces.

```
# echo xl1 xl2 xl3 >/etc/dhcpd.interfaces
Then, edit /etc/dhcpd.conf. The options are pretty self-explanatory.
option domain-name "example.com";
option domain-name-servers 192.168.1.3, 192.168.1.5;
subnet 192.168.1.0 netmask 255.255.255.0 {
option routers 192.168.1.1;
range 192.168.1.32 192.168.1.127;
}
```

This will tell your DHCP clients that the domain to append to DNS requests is example.com (so, if the user types in 'telnet joe' then it will send them to joe.example.com). It will point them to DNS servers 192.168.1.3 and 192.168.1.5. For hosts that are on the same network as an ethernet interface on the OpenBSD machine, which is in the 192.168.1.0/24 range, it will assign them an IP address between 192.168.1.32 and 192.168.1.127. It will set their default gateway as 192.168.1.1.

If you want to start dhcpd(8) from the command line, after editing /etc/dhcpd.conf, try:

```
# dhcpd -q fxp0
```

Where fxp0 is an interface that you want to start serving DHCP on. The -q flag makes dhcpd(8) quiet; otherwise it is very noisy

If you are serving DHCP to a Windows box, you may want dhcpd(8) to give the client a 'WINS' server address.

To make this happen, just **add the following line to your /etc/dhcpd.conf**:

```
option netbios-name-servers 192.168.92.55;
```

Printers

Edit /etc/printcap and /etc/hosts.lpd to get any printers set up. Consult lpd(8) and printcap(5) if needed.

Sendmail

OpenBSD ships with a **default /etc/mail/localhost.cf** file that will work for simple installations; it was generated from openbsd-localhost.mc in/usr/share/sendmail/cf.

Please see /usr/share/sendmail/README and /usr/share/doc/smm/08.sendmailop/op.me for information on generating your own sendmail configuration files. For the default installation, sendmail is configured to only accept connections from the local host and to not accept connections on any external interfaces. This makes it possible to send mail locally, but not receive mail from remote servers, which is ideal if you have one central incoming mail machine and several clients.

To cause sendmail to accept external network connections, modify the ``sendmail_flags'' variable in /etc/rc.conf.local to use the /etc/mail/sendmail.cf file in accordance with the comments therein.

This file was generated from openbsd-proto.mc. Note that sendmail now also listens on port 587 by default. This is to implement the RFC 2476 message submission protocol. You may disable this via the ``no_default_msa'' option in your sendmail .mc file. See /usr/share/sendmail/README for more information.

The /etc/mail/localhost.cf file already has this disabled.

BOOTP server

If this is a BOOTP server, edit /etc/bootptab as needed. You will have to turn it on in /etc/inetd.conf or run bootpd(8) in its stand-alone mode.

NFS server

If this is an NFS server make sure /etc/rc.conf has:

```
nfs_server=YES
```

Edit /etc/exports and get it correct. It is probably easier to reboot than to get the daemons running manually, but you can get the order correct by looking at /etc/netstart.

Daily, weekly, monthly scripts

Look at and possibly edit the /etc/daily, /etc/weekly, and /etc/monthly scripts. Your site specific things should go into /etc/daily.local, /etc/weekly.local, and /etc/monthly.local.

These scripts have been limited so as to keep the system running without filling up disk space from normal running processes and database updates. (You probably do not need to understand them.)

The /altroot filesystem can optionally be used to provide a backup of the root filesystem on a daily basis. To take advantage of this, you must have an entry in /etc/fstab with ``xx'' for the mount option:

```
/dev/wd0j /altroot ffs xx 0 0
```

and you must add a line to root's crontab:

```
ROOTBACKUP=1
```

so that the /etc/daily script will make a daily backup of the root filesystem.

Crontab (background running processes)

Check what is running by typing crontab -l as root and see if anything unexpected is present. Do you need anything else? Do you wish to change things? e.g., if you do not like root getting standard output of the daily scripts, and want only the security scripts that are mailed internally, you can type crontab -e and change some of the lines to read:

```
30  1  *  *  *   /bin/sh /etc/daily 2>&1 > /var/log/daily.out
30  3  *  *  6   /bin/sh /etc/weekly 2>&1 > /var/log/weekly.out
30  5  1  *  *   /bin/sh /etc/monthly 2>&1 > /var/log/monthly.out
```

Keep your system up-to-date.

```
#!/bin/sh

# This script will update your src,ports, and sys
rdate -ncv time.nrc.ca
Today="`date`"
echo Today is $Today
echo "You must set this in your alias first"
echo "setenv CVSROOT anoncvs@anoncvs.ca.openbsd.org:/cvs"
export CVSROOT=anoncvs@anoncvs.ca.openbsd.org:/cvs

cd /usr/src
cvs -q up -rOPENBSD_3_4 -Pd

cd /usr
cvs -q up -rOPENBSD_3_4 ports

cd /usr/src/sys/arch/i386/
cvs -q update -Pd

echo "Building your new kernel please wait..."
echo "Cleaning out the old object files."
cd /usr/src
find . -type l -name obj | xargs rm
make cleandir
rm -rf /usr/obj/*
make obj
echo "Compiling a new kernel"
cd /usr/src/sys/arch/i386/conf
config GENERIC
cd ../compile/GENERIC/
make clean && make depend && make
cp /bsd /bsd.old && cp bsd /bsd
echo "Reboot than run 4binbld.sh"


# This script will build your system binaries. Run it after you
update the system source files and build the kernel and reboot.

#!/bin/sh

Today="`date`"
echo Today is $Today
echo "Building your system binaries"
cd /usr/src
make build
# make obj && make build (use this only if you did not make obj
before)
echo "Reboot"
```

Annex :: Changing services Banner

sshd

Open /version.h and cut the "_3.5p1" from the end. Re-compile and it's done.

To recompile ssh.

```
# cd /usr/src/lib/libssl
@ make build
```

Apache

Open /src/include/httpd.h and search for:

```
#define SERVER_BASEVENDOR “Apache Group”
#define SERVER_BASEPRODUCT “Apache”
#define SERVER_BASEREVISION “”
```

Change this to the desired values (BASEVENDOR: Microsoft, BASEPRODUCT: Microsoft-IIS, BASEREVISION: 5.0). Now re-compile apache.

Next: open your httpd.conf and search for the ServerTokens directive. If it’s not there, add it. Set ServerTokens to Min (“ServerTokens Min”). More information about the ServerTokens directive is at:

http://carnagepro.com/pub/Docs/Apache2/mod/core.html#servertokens .

Annex :: Rebuilding

Rebuild Your Kernel

```
cd /usr/src
find . -type l -name obj | xargs rm
make cleandir
rm -rf /usr/obj/*
make obj
echo "Compiling a new kernel"
cd /usr/src/sys/arch/i386/conf
config GENERIC
cd ../compile/GENERIC/
make clean && make depend && make
cp /bsd /bsd.old && cp bsd /bsd
```

Rebuild Your System Binaries

```
echo "Building your system binaries"
cd /usr/src
make build
echo "Reboot"
```

Rebuild and install OpenSSL

```
cd lib/libssl
make obj
make depend
make
make install
```

Recompiling Apache.

```
apachectl stop
echo "Rebuild and install httpd and its modules"
cd usr.sbin/httpd
make -f Makefile.bsd-wrapper obj
make -f Makefile.bsd-wrapper cleandir
make -f Makefile.bsd-wrapper depend
make -f Makefile.bsd-wrapper
make -f Makefile.bsd-wrapper install
apachectl start
```

Rebuild and install X

```
# cd /usr/obj
# lndir /usr/XF4
# make build
```

Rebuild and install isakmpd

```
cd sbin/isakmpd
make clean
make depend
make
make install
```

Rebuild and install sysctl

```
cd sbin/sysctl
make depend
make
make install
```

Annex :: Patching OpenBSD 3.4 (3/20/2004)

You need to check http://www.openbsd.org/errata.html often for any security patches. You can get all the patch in on tar file from:

```
wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4.tar.gz
```

Below is a list of needed patches (3/20/2004).

```
Wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/016_openssl.patch
Apply by doing:
        cd /usr/src
        patch -p0 < 016_openssl.patch

And then rebuild and install OpenSSL:
        cd lib/libssl
        make obj
        make depend
        make
        make install


wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/015_isakmpd2.patch
Apply by doing:
        cd /usr/src
        patch -p0 < 015_isakmpd2.patch

Then rebuild and install isakmpd:
        cd sbin/isakmpd
        make clean
        make depend
        make
        make install

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/014_httpd2.patch
Apply by doing:
        cd /usr/src
        patch -p0 < 014_httpd2.patch

And then rebuild and install httpd and its modules:
        cd usr.sbin/httpd
        make -f Makefile.bsd-wrapper obj
        make -f Makefile.bsd-wrapper cleandir
        make -f Makefile.bsd-wrapper depend
        make -f Makefile.bsd-wrapper
        make -f Makefile.bsd-wrapper install

If httpd had been started, you might want to run
        apachectl stop
before running "make install", and
```

```
        apachectl start
```

afterwards.

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/013_tcp.patch

Apply by doing:

```
        cd /usr/src
        patch -p0 < 013_tcp.patch
```

Rebuild your kernel.

Update headers.

```
        make includes
```

Then rebuild and install sysctl:

```
        cd sbin/sysctl
        make depend
        make
        make install
```

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/012_font.patch

Apply by doing:

```
        cd /usr/XF4
        patch -p0 < 012_font.patch
```

rebuild and install X:

```
        cd /usr/obj
        lndir /usr/XF4
        make build
```

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/011_ip6.patch

Apply by doing:

```
        cd /usr/src
        patch -p0 < 011_ip6.patch
```

And then rebuild your kernel.

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/010_sysvshm.patch

Apply by doing:

```
    cd /usr/src
    patch -p0 < 010_sysvshm.patch
```

And then rebuild your kernel.

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/009_isakmpd.patch

Apply by doing:

```
        cd /usr/src
        patch -p0 < 009_isakmpd.patch
```

Then rebuild and install isakmpd:

```
        cd sbin/isakmpd
        make clean
        make depend
        make
        make install
```

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/008_sem.patch

```
Apply by doing:
        cd /usr/src
        patch -p0 < 008_sem.patch
And then rebuild your kernel.
```

```
wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/007_uvm.patch
Apply by doing:
        cd /usr/src
        patch -p0 < 007_uvm.patch
And then rebuild your kernel.
```

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/i386/006_ibcs2.patch

```
Apply by doing:
        cd /usr/src
        patch -p0 < 006_ibcs2.patch
And then rebuild your kernel.
```

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/005_exec.patch

```
Apply by doing:
        cd /usr/src
        patch -p0 < 005_exec.patch
And then rebuild your kernel.
```

wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/004_httpd.patch

```
Apply by doing:
        cd /usr/src
        patch -p0 < 004_httpd.patch

And then rebuild and install httpd and its modules:
        cd usr.sbin/httpd
        make -f Makefile.bsd-wrapper obj
        make -f Makefile.bsd-wrapper cleandir
        make -f Makefile.bsd-wrapper depend
        make -f Makefile.bsd-wrapper
        make -f Makefile.bsd-wrapper install

If httpd had been started, you might want to run
        apachectl stop
before running "make install", and
        apachectl start
afterwards.
```

```
wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/003_arp.patch
Apply by doing:
    cd /usr/src
    patch -p0 < 003_arp.patch
And then rebuild your kernel.
```

```
wget ftp://ftp.openbsd.org/pub/OpenBSD/patches/3.4/common/002_asn1.patch
Apply by doing:
        cd /usr/src
        patch -p0 < 002_asn1.patch
```

```
And then rebuild and install OpenSSL:
        cd lib/libssl
        make obj
        make depend
        make
        make install
```

Rebuild Your Kernel

```
echo "Compiling a new kernel"
cd /usr/src/sys/arch/i386/conf
config GENERIC
cd ../compile/GENERIC/
make clean && make depend && make
cp /bsd /bsd.old && cp bsd /bsd
```

echo “please reboot the puffy” by doing issuing reboot command

Annex :: Links / References

http://www.openbsd.org
ftp://ftp.openbsd.org/pub/OpenBSD/3.4/packages/i386/
http://www.openbsd.org/faq/faq10.html#Patches
http://www.openbsd.org/anoncvs.html
http://www.openbsd.org/faq/index.html
http://www.deadly.org/
http://nomoa.com/bsd/
http://www.my-snort.org/downloads/
http://geodsoft.com/howto/harden/bsdinst.htm
http://geodsoft.com/howto/harden/bsdhardn.htm
http://BSD.GeodSoft.com/howto/harden/
http://hades.uint8t.org/systrace.html
http://www.openbsd-edu.net/liens.htm#pf
http://unixscout.bay13.net/uscout/faq_snort.html
http://www.openbsd.org/cgi-bin/cvsweb/www/xf86configs/
http://store.yahoo.com/bsdmall/openbsd.html
http://www.openbsdforums.org/
http://www.tcfs.it/docs/manpages/BSD/utils.html
http://www.tongatapu.net.to/nix/OpenBSD/
http://www.ncl.ac.uk/ucs/unix/unixhelp/shell_alias.html
http://www.allcommands.com/bsd%20commands%20list.html
http://www.tcsh.org/tcsh.html/Argument_list_processing.html
http://www.google.com/bsd/
http://www.monkey.org/openbsd/
http://www.daemonnews.org
http://www.bsdforums.org/
http://www.onlamp.com/bsd/
http://pmk.sourceforge.net/
http://keihanna.dl.sourceforge.net/sourceforge/pmk/pmk-0.7.tar.gz
http://www.signull.com/documents/SNE-whitepaper.pdf
http://www.fluffy.co.uk/boxbackup/
http://mircryption.sourceforge.net/Extras/mircryption_xchatsrc.zip
ftp://ftp.openbsd.org/pub/OpenBSD/3.4/packages/i386/
http://linux.cudeso.be/obsd-info.php

Annex :: MY HOME NEC LCD

```
Size : 19"
Native Resolution  : 1280 x 1024
Recommended Resolution: 1280 x 1024 @ 60 Hz
Model Numbers LCD1912
Synchronization Range (Automatic):
Horizontal: 31.5 kHz to 81.0 kHz
Vertical: 56.0 Hz to 75.0 Hz
Resolutions Supported: 720 x 400* at 70 Hz
640 x 480* at 60 Hz to 75 Hz
800 x 600* at 56 Hz to 75 Hz
832 x 624* at 75 Hz
1024 x 768* at 60 Hz to 75 Hz
1280 x 1024 at 60 Hz to 70 Hz
```

LCD Module: 19-inch (19" viewable image size), active matrix, thin film transistor (TFT), liquid crystal display (LCD), 0.294 mm dot pitch, 250 cd/m² white luminance - typical, 500:1 contrast ratio - typical, 25ms response time – typical

GNU Free Documentation License

Version 1.2, November 2002