

25 DAYS UNTIL THE SCHEDULED RELEASE OF OPENBSD 4.6!

[INSTALLATION](#)

[PATCHING](#)

[UPDATING](#)

[SECURITY](#)

[TIPS & TRICKS](#)

[CONTACT](#) [HOME](#)

TIPS AND TRICKS FOR OBSD 4.5

What follows are some suggestions that new users to OpenBSD might find helpful.

- [Single user mode \(reset root password\)](#)
- [Disabling root logins to OpenSSH](#)
- [Encrypt the swap partition](#)
- [Installing the BASH shell](#)
- [Formatting a floppy disk \(3.5\)](#)
- [Keeping system time with OpenNTPD](#)
- [Getting ftpd \(non-anonymous\) up and running](#)
- [Getting Apache \(httpd\) up and running](#)
- [Changing the timezone](#)
- [lsof command](#)
- [Editing files with nano](#)
- [Locking a user out of their account](#)
- [Locking all users out of their accounts](#)
- [Mounting a USB flash drive](#)
- [Burning a CD-RW](#)
- [Useful commands](#)



SINGLE USER MODE (RESET ROOT PASSWORD)

If you have forgotten root's password or have mistakenly disabled the shell you used for root and other accounts, you will need to boot into single user mode to fix these problems.

- step 1 - Start single user mode
- step 2 - Check file systems with fsck (file system consistency check)
- step 3 - Mount filesystems
- step 4 - Reset root password

1. When you restart the system wait until you see something similar to the below

```
Using drive 0, partition 3.
Loading...
probing : pc0 com0 apm mem[634K 319M a20=on]
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 3.01
boot>
```

at this point you are going to want to enter into single user mode:

```
Using drive 0, partition 3.
Loading...
probing : pc0 com0 apm mem[634K 319M a20=on]
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 3.01
boot> boot -s
```

2. Now run fsck on all partitions, to make sure things are okay for changes

```
Enter pathname of shell or RETURN for sh: <press enter>
# fsck -p
```

3. Mount all filesystems

```
# mount -a
```

export the TERM environmental variable only if you need to edit files:

```
# export TERM=vt220
```

4. Reset root's password and then reboot

```
# passwd
Changing local password for root.
New password: ILikeMonkeys
Retype new password: ILikeMonkeys
# shutdown -r now
```

DISABLE ROOT LOGINS TO OPENSSSH

The OpenSSH server is defaulted to allow root logins. Disabling root access will help you from practicing the bad habit of using the root account as your primary account. A more secure way of using OpenBSD is to log in as a user who belongs to the group `wheel` and using the `su` command to become root whenever root privileges are needed.

- step 1 - Add a user to the wheel group
- step 2 - Edit the sshd configuration file to disallow root logins

1. Adding the user nathan to the wheel group

```
# usermod -G wheel nathan
```

2. Edit the entry in the `/etc/ssh/sshd_config` file from

```
#PermitRootLogin yes
```

to:

```
PermitRootLogin no
```

now restart sshd so the changes take effect without rebooting:

```
# kill -HUP `cat /var/run/sshd.pid`
```

ENCRYPT THE SWAP PARTITION

By default OpenBSD 4.5 will encrypt the swap partition. To turn this on for OpenBSD versions 3.7 and below:

- step 1 - Enable this feature without a reboot
- step 2 - Edit the sysctl config file, so that after a reboot the swap partition will be encrypted

1. Change the kernel state variable

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

2. Edit `/etc/sysctl.conf` from

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

to:

```
vm.swapencrypt.enable=1
```

INSTALLING THE BASH SHELL

If you did not buy your copy of OpenBSD then you won't have any packages available locally in order to install the bash shell. The below example will show how to obtain the BASH package remotely.

- step 1 - Adding the BASH shell remotely
- step 2 - Setting BASH as your login shell

1. Add the i386 package for the BASH shell

```
# pkg_add -v ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/i386/bash-3.2.48.tgz
```

2. Setting BASH as your login shell

```
# chsh -s bash
```

FORMATTING A FLOPPY DISK (3.5)

Setting up a floppy drive is slightly different than what you would do on Linux.

- step 1 - Create a floppy directory
- step 2 - Format the floppy
- step 3 - Partition the floppy
- step 4 - Create the file system
- step 5 - Mounting the floppy drive

1. Create the /mnt/floppy directory

```
# mkdir /mnt/floppy
```

make /mnt/floppy directory accessible by root only:

```
# chmod 700 /mnt/floppy
```

2. Format a floppy, red text represents user input

```
# fdformat fd0
Format 1440K floppy `/dev/rfd0c'? (y/n): y
```

3. Next we need to partition the floppy

```
# disklabel -E fd0
Initial label editor (enter '?' for help at any prompt)
> a a
offset: [0] <press enter>
size: [2880] <press enter>
FS type: [4.2BSD] <press enter>
> w
> q
No label changes.
```

4. Create the file system

```
# newfs fd0a
```

5. Mounting the floppy drive

```
# mount -t ffs /dev/fd0a /mnt/floppy
```

or add this line to your /etc/fstab file:

```
/dev/fd0a /mnt/floppy ffs rw,noauto 0 0
```

KEEPING SYSTEM TIME WITH OPENNTPD

The program OpenNTPD (ntpd) is included in the base install of OpenBSD. Ntpd is both server and client side software that will connect to a time server and sync its time via the NTP (Network Time Protocol). A list of NTP servers can be found [here](#).

What follows is setting up ntpd as a client

- step 1 - Create the file /etc/rc.conf.local if it doesn't exist
- step 2 - Edit rc.conf.local so ntpd will run at boot-up
- step 3 - Setting the NTP server that will be used to check time

1. Create the file /etc/rc.conf.local if it doesn't exist

```
# touch /etc/rc.conf.local
```

2. Adding an entry to the /etc/rc.conf.local file so ntpd runs at startup

```
# echo ntpd_flags="-s\" >> /etc/rc.conf.local
```

3. Edit the /etc/ntp.conf file, adding the time server to be used

```
# $OpenBSD: ntpd.conf,v 1.9 2008/10/10 11:46:22 sthen Exp $
# sample ntpd configuration file, see ntpd.conf(5)

# Addresses to listen on (ntpd does not listen by default)
# listen on *

# sync to a single server
server tick.cs.unlv.edu
```

```
# use a random selection of 8 public stratum 2 servers
# see http://twiki.ntp.org/bin/view/Servers/NTPPoolServers
# servers pool.ntp.org

# use a specific local timedelta sensor (radio clock, etc)
#sensor nmea0

# use any detected timedelta sensor
#sensor *
```

Change the `server` setting to a server [near you](#) and comment out the `servers` line.

If your time is off by a few minutes or hours a large time change will take place during boot-up (when using the `-s` flag). But while your server is running, a large time change will not take place, helping to prevent abuse to the log files. Instead a time change of one second will take place over 4 minute intervals, until time has been corrected.

To check if `ntpd` is making those changes

```
# grep ntpd /var/log/daemon
```

GETTING FTPD (NON-ANONYMOUS) UP AND RUNNING

Setting up and running `ftpd` for users:

- step 1 - Edit `/etc/inetd.conf`
- step 2 - Restart `inetd`

1. Edit (uncomment) the `ftp` entry in `/etc/inetd.conf` from

```
#ftp stream tcp nowait root /usr/libexec/ftpd ftpd -US
```

to:

```
ftp stream tcp nowait root /usr/libexec/ftpd ftpd -US
```

2. Restart `inetd` so that `ftp` will be recognized as a service

```
# kill -HUP `cat /var/run/inetd.pid`
```

Users listed in the `/etc/ftpusers` file will not be allowed to log into `ftpd`. Commenting out the `'root'` entry in this file will allow `root` to login via `ftp` (which is not recommended).

GETTING APACHE (HTTPD) UP AND RUNNING

Installing all the sets (`base45.tgz`) of OpenBSD will assure that you have Apache 1.3.29 installed. Apache 2.0 will not come with the base installation of OpenBSD due to Apache's license changes for 2.0.

- step 1 - Location of Apache configuration file
- step 2 - Starting `httpd`
- step 3 - Having `httpd` run at startup

1. Configuration of Apache is done using the `/var/www/conf/httpd.conf` file.

2. Starting the `http` daemon

```
# apachectl start
```

A good rule of thumb is not to edit your `/etc/rc.conf` file. Instead create then edit a file called `/etc/rc.conf.local`. Settings specified in `rc.conf.local` will take precedence over settings in the `/etc/rc.conf` file.

3. Edit your `/etc/rc.conf.local` file adding this entry

```
# echo 'httpd_flags="" ' >> /etc/rc.conf.local
```

CHANGING THE TIMEZONE

OpenBSD's timezone is set from the `/etc/localtime` binary file which will be soft linked (symbolic link) to one of the files located in the `/usr/share/zoneinfo` directory structure.

- step 1 - Location of timezone file
- step 2 - Find your timezone
- step 3 - Removing, then adding a soft link

1. Configuring the timezone is done using the `/etc/localtime` file.

2. Locate your timezone file in the `/usr/share/zoneinfo/` directories.

3. Changing your timezone to Los Angeles (pacific) time

```
# rm /etc/localtime
# ln -s /usr/share/zoneinfo/America/Los_Angeles /etc/localtime
```

LSOF (LIST OPEN FILES) COMMAND

The `lsof` command is a little used command. `lsof` will let you track down which files are open and who's using them along with open pipes and ports.

- step 1 - Remotely add the `lsof` package
- step 2 - Using `lsof`

1. Remotely add the `lsof` package

```
# pkg_add -v ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/i386/lsof-4.81p0.tgz
```

2. Using `lsof` to list open ports and established connections

```
# lsof -i
```



Note: The command `fstat` is similar to `lsof`, and comes with the base installation of OpenBSD. And since there's always more than one way to get things done in Unix, `lsof` is worth mentioning.

EDITING FILES WITH NANO

An alternative to using `vi` when editing files.

- step 1 - Remotely add the `nano` package
- step 2 - Using `nano`
- step 3 - Making `nano` your default editor in BASH

1. Remotely add the `nano` package

```
# pkg_add -v ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/i386/nano-2.0.9.tgz
```

2. Using `nano` to edit your `hosts` file

```
# nano /etc/hosts
```

3. Adding an entry to your `.bash_profile` file

```
# cd
# echo export EDITOR=nano >> .bash_profile
```

having the changes 'take' without logging out:

```
# cd
# source .bash_profile
```

LOCKING A USER OUT OF THEIR ACCOUNT

There will come a time when an administrator needs to prevent a user from using their account.

Locking the user `nathan` out of his account. As root

```
# chsh -s nologin nathan
```

Unlocking the user `nathan` from his account. As root

```
# chsh -s sh nathan
```

A better way of locking a user out of their account is by using the `userdel` command which will not only change the user shell to a nologin shell but the user's password will be changed to an "impossible" one. Also, the user's home directory will not be removed.

Locking the user `nathan` out of his account. As root

```
# userdel -p true nathan
```

LOCKING ALL USERS OUT OF THEIR ACCOUNTS

There will come a time when an administrator needs to prevent all users from using their accounts. Root does not fall under the default login class and will not be locked out.

Locking all users from their accounts. As root

```
# touch /etc/nologin
```

Allowing logins again. As root

```
# rm /etc/nologin
```

The login program is controlled by `/etc/login.conf` and can be tweaked to meet your needs, including setting default user environmental variables and fine-tuning your system security.

MOUNTING A USB FLASH DRIVE

USB drives usually show up as (emulated) SCSI drives.

- step 1 - Plug in your USB flash drive
- step 2 - Create a flash drive directory
- step 3 - Mount the flash drive

1. Physically plug in your USB flash drive.

2. Create a flash drive directory

```
# mkdir /mnt/flashdrive
```

3. Mount the flash drive

```
# mount /dev/sd0i /mnt/flashdrive
```

What follows is setting up a flash drive to work with OpenBSD and Windows

- step 1 - Plug in your USB flash drive
- step 2 - Create a flash drive directory (if not already created)
- step 3 - Create and format a MSDOS partition
- step 4 - Mount the flash drive filesystem

1. Physically plug in your USB flash drive.

2. Create a flash drive directory (if not already created)

```
# mkdir /mnt/flashdrive
```

3. Create and format a MSDOS partition

```
# newfs -t msdos sd0c
```

This will create the `i` partition on the `sd0` flash device.

4. Mount the flash drive filesystem

```
# mount /dev/sd0i /mnt/flashdrive
```

 **Note:** Remember to unmount the flash filesystem before unplugging the flash device from your computer or data corruption might occur.

Unmount the flash filesystem

```
# umount /mnt/flashdrive
```

Now you can safely remove your USB drive.



Note: OpenBSD will now automatically dismount USB filesystems if the device is disconnected.

BURNING A CD-RW

To create the ISO image files that are being used in the following CD burning examples you will need to have the cdrtools package installed.

So, remotely install the i386 cdrtools package

```
# pkg_add -v ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/i386/cdrtools-2.01pl.tgz
```

1ST METHOD

The following method uses the `cdio` command, which comes with the base installation of OBSD.

What follows is an example of burning the home directories to a CD-RW

- step 1 - Create an ISO image of the information targeted for the CD-RW
- step 2 - Blank the CD-RW disk
- step 3 - Burn the ISO image to the blank CD-RW

1. Next make an ISO image of the home directories

```
# mkisofs -v -l -L -r -J -R -o /root/backup.iso /home/
```

2. Now blank the CD-RW disk

```
# cdio blank
```

3. And finally burn the ISO image to the blank CD-RW

```
# cdio tao /root/backup.iso
```

2ND METHOD

The following method uses the `cdrecord` command, which comes with the cdrtools package.

What follows is an example of burning the home directories to a CD-RW

- step 1 - Create an ISO image of the information targeted for the CD-RW
- step 2 - Blank the CD-RW disk
- step 3 - Burn the ISO image to the blank CD-RW

1. Next make an ISO image of the home directories

```
# mkisofs -v -l -L -r -J -R -o /root/backup.iso /home/
```

2. Now blank the CD-RW disk

```
# cdrecord -blank fast dev=/dev/cd0c
```

3. And finally burn the ISO image to the blank CD-RW

```
# cdrecord -v dev=/dev/cd0c /root/backup.iso
```

How to figure out what device node is associated to the CD burner

I have a Memorex CD burner, so I will search `dmesg` output to see if the kernel has recognized my burner and to find out what device node the burner has been associated to.

Greping `dmesg` output

```
# dmesg | grep Memorex
```

```
cd0 at scsibus0 targ 0 lun 0: <Memorex, 48MAX 244816AJ, KWH8> SCSI0 5/cdrom removable
```

From the `dmesg` output you can see that the device node `cd0` has been associated to the Memorex burner.

Mounting the CD burner

- step 1 - Create a CD-ROM directory
- step 2 - Mount the CD filesystem

1. Create a CD-ROM directory

```
# mkdir /mnt/cdrom
```

2. Mounting the CD filesystem

```
# mount /dev/cd0c /mnt/cdrom/
```

USEFUL COMMANDS

Adding a user

```
# adduser
```

Remove a user named nathan

```
# rmuser nathan
```

Checking disk usage

```
# df -ht ffs
```

Restart OpenSSH without rebooting

```
# kill -HUP `cat /var/run/sshd.pid`
```

Using netstat

```
# netstat -a -f inet
```

Checking network usage statistics

```
# netstat -s
```

Check disk usage of the home directories

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

Finding a file, searching the entire directory structure

```
# find / -name bsd
```

Adding the user nathan to the wheel group

```
# usermod -G wheel nathan
```

Using pkg_add's interactive mode to add pidgin (gaim)

```
# export PKG_PATH=ftp://ftp.openbsd.org/pub/OpenBSD/4.5/packages/i386/  
# pkg_add -i pidgin  
Choose one package  
  0: <None>  
  1: pidgin-2.5.4p0  
  2: pidgin-2.5.4p0-gtkspell  
Your choice:
```

If you notice any errors, please [let me know](#).

OTHER OPENBSD TUTORIALS



- [Installation](#) - demonstration of a FTP installation
- [Security](#) - security suggestions
- [Patching](#) - patching and kernel building
- [Updating](#) - updating with CVSup

NO AFFILIATION BETWEEN THIS SITE AND THE OPENBSD PROJECT EXISTS OR IS IMPLIED.