

A FreeBSD 4.8-RELEASE Operating System Security Checklist

[Support the BSD projects by getting a subscription to one or more of the family: FreeBSD, OpenBSD, NetBSD and Darwin](#)

Updated: June 12, 2003

Location: <http://www.sddi.net/FBSDSecCheckList.html>

Location of the checklist version: <http://www.sddi.net/FBSDSecCheckListaslist.html>

This document is also available in Portuguese, translated by someone or something that didn't notify me, but is still much appreciated. Assuming it's accurate, of course.

If you are part of an institution or business, and you found this document useful, consider paying a fee to assist us in producing more documents and keeping this one up-to-date. You can use [Paypal](#) and send to the account of payment@sddi.net.

Location: http://www.traduzweb.com.br/scripts/tws.dll/ingport?p=kounen&lg=in_pt&url=http%3A%2F%2Fwww.sddi.net%2FFBSDSecCheckList.html%23Installation

Please note that I haven't spent much time recently on the document, but I do plan to expand and revise. As 5.0 is not meant for production, I will at least wait for 5.1 to be out for a while before I approach locking down FreeBSD 5.x.

This document is intended to be a working checklist of security settings implemented on FreeBSD servers version 4.8-RELEASE.

There are a number of well-written and often brilliant documents providing overviews, how-to's and faqs on FreeBSD security for the practical systems administrator, but there is no to-the-point, checklist that can be a tool for each time a server is built. While there are no elaborate explanations supplied here, you can check out the bibliography at the end of this document. Remember, nothing upsets your fellow sysadmins more than not RTFM.

It is not intended to be final document, but rather a working, regularly updated tool, complemented with the input of others and [myself](#).

Ideally, this document could be accessed over the internet or printed out and used as a reference when building a server.

This document has changed from its original focus, you may notice. Initially, the point was not to

provide details on why each change is made, or to integrate packages and ports outside the basic install. It became clear that doing this topic justice meant adding a few applications which would provide a more maintenance-free, secure server. These extra applications, however, do not move beyond the basic operating system, so it still does not address important basics such as firewalling with ipf, ipfw or pf ([which has now been ported to FreeBSD](#)) nor the various mail transfer agents, www servers and their configurations. Currently, those extra applications are chkrootkit which checks for a variety of root kits, rdate as a replacement for ntp and cvsup and portupgrade for updating system and port files.

A note about logging. Several people have written in suggesting adding options such as `log_in_vain`, etc. These issues weren't overlooked when this document was put together. However, from the many years I have spent time reviewing firewall logs, I have to say that I do not want to spend time looking at lots of useless data. And I don't consider port scans an intrusion into my networks. Your hardware that sits on the internet will be scanned. You know that. If port scans don't show up for a significant portion of your logs, then your ip configuration is probably off :-). I want to know if someone goes a step beyond that. If someone attempts to ssh into the box, my network time isn't correctly synchronized, etc., then I want to know. If you need the data and want to spend time pursuing or cataloging everyone who spends time doing port scans, great. Sounds like a "firewall administrator" position in San Francisco in 1999 paying six figures. Great work, if you can get it. I think for most people, this is an unreasonable way to spend time, particularly in economic times like today. The internet is still a pirate-ridden sea to a large extent, and the *BSD's are iron-clad warships. Everytime I see a pirate ship, there's no reason to panic. If they start knocking on the iron-sides, then I want to know. But in the spirit of democracy, I have added `log_in_vain` and some other changes to this document. If you disagree, feel free to email me.

Finally, although I have tried to be as accurate as possible, it should be clear that I'm not responsible for any errors you make using this document.

- 1. [Installation](#)
- 2. [Configuration](#)
- 3. [Users](#)
- 4. [Message of the Day](#)
- 5. [OpenSSH](#)
- 6. [rc.conf](#)
- 7. [login.conf & auth.conf](#)
- 8. [sysctl.conf](#)
- 9. [fstab](#)
- 10. [CVSup & portupgrade](#)
- 11. [Cron Jobs](#)
- 12. [Kernel Changes](#)

- [13. File Permissions](#)
- [14. Network Time Protocol](#)
- [15. TCP Wrappers](#)
- [16. Console Access](#)
- [17. Bash Shell](#)
- [18. chflags](#)
- [19. Cleaning Up](#)
- [20. Possible Additions](#)
- [Bibliography](#)

Key

italicized represents screen prompts

the command line interface is represented by [user@server /dir]# and indented slightly

an editing session (in vi, for instance) is indented further

ZZ means to hold down the shift key and hit the letter z twice. That's a quick way to save and exit a file with vi.

Also in vi, if you want to see the line numbers, get out of edit mode and type `:set num`. This will list line numbers along the left side of the margin.

So let's get started.

1. Installation

When determining slices and mount points, more than the default is better. It will allow us to set some options on each slice by mount point in the fstab file later on.

If you do not have more than a couple of users, then there's no need to make /usr/home any significant size. And if your logs are sitting on a separate syslog server, then /var doesn't need to be enormous.

For example, your slices may look like this:

none (swap)

/

/tmp

/usr

/usr/home

/var

/root

2. Configuration

No inetd.conf

No port_map if not using nfs

No to ntp, since we are going to use rdate. Note, however, the ntp server (s) to be used.

We do want to install the ports collection, which goes into /usr/ports

When prompted to browse the available ports, add the following:

- /security/chkrootkit-0.40
- /sysutils/rdate-1.0
- /sysutils/portupgrade20030228
- /net/cvsup-without-gui-16.1g (gui's don't belong on servers, IMO)

3. Users

It is critical to setup an additional, non-privileged user for several reasons. For instance, we are not going to allow root to ssh into the box. This user should be in the wheel group, so that they can su into root when necessary.

Create a group for OpenSSH users, and include any non-privileged accounts that should have remote access are in this group.

```
[user@server /dir]#vi /etc/group
```

Add the following line, with the group for OpenSSH users. After the last colon, add the non-privileged user (s) who should have OpenSSH access.

```
sshusers:*:1001:nonprivileged users
```

Save and exit vi.

ZZ

4. Message of the Day

```
[user@server /dir]#cp /etc/motd /etc/motd.old
```

```
[user@server /dir]#rm /etc/motd
```

```
[user@server /dir]#vi /etc/motd
```

Be sure to explicitly state that only authorized users performing authorized tasks are permitted to access the server. You also want to state that by logging onto the server, the user is consenting to their activity being logged and monitored. This is important in the event of the prosecution of unauthorized users.

Save and exit vi

ZZ

```
[user@server /dir]#cp /etc/motd /etc/issue          # we want to use the motd for remote
                                                    logins also. see /etc/ssh/sshd_config.
```

5. OpenSSH

We want to use dsa keys for authentication, and not passwords. Of course, if someone has access to your client ssh machine with the dsa key, you're cooked. Same is true for someone finding out your password. But using dsa keys means that the remote user can't even attempt brute force password cracking. You can go with password or dsa keys, it's your choice. Just don't dare to use telnet for remote access.

Open and edit /etc/ssh/sshd_config so we can edit the ssh daemon settings for users accessing the box remotely.

```
[user@server /dir]#vi /etc/ssh/sshd_config
```

```
Port 22
```

```
Protocol 2
```

```
#Hostkey /etc/ssh/ssh_host_key
```

```
PermitRootLogin no
```

```
MaxStartups 5:50:10
```

```
#after 5 bad logins, refuse 50% of new
ones, and refuse more than 10 total
```

```
X11Forwarding no
```

```
PrintLastLog yes
```

```
SyslogFacility auth
```

```
# Send log information to /var/log/auth.
```

```
LogLevel VERBOSE
```

```
# since OpenSSH is the only legitimate
remote access to the box, we want to watch
everything.
```

```
PasswordAuthentication no
```

```
PermitEmptyPasswords no
Banner /etc/issue
AllowGroups sshusers # create some group that you can put
OpenSSH users into
```

```
Save and exit vi
```

```
ZZ
```

```
Next, we'll open and edit /etc/ssh/ssh_config
```

```
[user@server /dir]#vi /etc/ssh/ssh_config
```

```
ForwardAgent no
```

```
ForwardX11 no
```

```
PasswordAuthentication no
```

```
CheckHostIP yes
```

```
Port 22
```

```
Protocol 2
```

```
Save and exit vi
```

```
ZZ
```

Then let's generate DSA keys for authentication with your non-privileged user account, aka <nonprivuser>.

```
[user@server /dir]#su - <nonprivuser>
```

as root, change to the non-privileged account

```
[user@server /dir]#ssh-keygen -d
```

You'll see the following:

```
Generating public/private dsa key pair.
```

```
Enter file in which to save the key
```

```
(/home/<nonprivuser>/.ssh/id_dsa):
```

```
Enter passphrase (empty for no passphrase):
```

```
Enter passphrase (empty for no passphrase):
```

```
Your identification has been saved in /home/<nonprivuser>/.ssh/id_dsa.
```

```
Your public key has been saved in /home/<nonprivuser>/.ssh/id_dsa.pub
```

```
The key fingerprint is:
```

```
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx <nonprivuser>@<host>
```

```
[user@server /dir]#cd .ssh
```

```
[user@server /dir]#cat id_dsa.pub > authorized_keys2
```

Now copy the key to a floppy so that we can transfer the dsa key to the machine from which you'll be ac
Exit and get back to being root.

First, confirm that the file is on the floppy, then go and delete the keys from the non-privileged user's .ssl floppy is safe, because if it's damaged, you'll need to regenerate the dsa key again.

6. rc.conf

```
[user@server /dir]#vi /etc/rc.conf
```

```
inetd_enable="NO"           # Better ways to run your daemons
syslogd_enable="YES"       # We want logging, of course. If it is remote, the
                           # the next line. If it is local, make sure your /var c
                           # the potential volume.

syslogd_flags="-ss"       # This will close udp port 514 if we don't want r
                           # syslogging to or from this server. Obviously, do
                           # this if you are using a separate syslog server or a
                           # syslogs from another server.

icmp_drop_redirect="YES"   # Drop pings. However, you may want the host
                           # to pings so that you can monitor. Dropping ping
                           # your host is less likely to be picked up when som
                           # scanning ranges of ip addresses. You can limit tl
                           # bandwidth use in a kernel option setting later on
                           # document.

icmp_log_redirect="YES"   # If you want to log redirected pings.
clear_tmp_enable="YES"    # Empty out /tmp on boot.
portmap_enable="NO"       # If not running nfs
icmp_bmcastecho="NO"     # Prevent springboarding & smurf attacks
fsck_y_enable="YES"      # fsck -y will run if initial preen of filesystems f
update_motd="NO"         # Don't update message of the day
tcp_drop_synfin="YES"    # Drop synfin packets. See below for the necess
                           # change.

log_in_vain="YES"        # Set this if you want to log all attempts to acces
                           # by a closed port.

sshd_enable="YES"        # As this will be our way of securely accessing t
                           # remotely
```

Save and exit vi

ZZ

7. login.conf & auth.conf

The default md5 password encryption will now be replaced with Blowfish, an algorithm yet to be broken. While strong encryption is a necessity, it is vital to remember that encryption in itself is a small part of security on a server. A metaphor stated at a [NY ISSA](#) meeting is fitting: having strong password encryption is like having a really high pole sitting in your front yard. No one can get over it, but there's lots of room to go around. Additionally, other password restrictions will be implemented which will increase security in that arena.

```
[user@server /dir]#vi /etc/login.conf
```

We'll do this under the default:\ section

```
# change the password encryption to Blowfish instead of the defau
```

```
:passwd_format=blf:\      md5
:passwordtime=52d:\      # force 52 day password changes
:mixpasswordcase=true:\  # warn users to use mixed-case passwords
:minpasswordlen=9:\      # make 9 characters the minimum password length<
:idletime=32:\           # automatically logoff users idle for more than 32 minutes
```

Save and exit vi

ZZ

Now make the database.

```
cap_mkdb /etc/login.conf
```

All passwords now need to be changed so that they are Blowfish-encrypted.

Confirm this with vipw, which is a tool to quickly edit your /etc/master.passwd file.

We are only concerned with the second and last fields, which are separated by a colon : . The second field is the user password in ciphertext. Make sure this begins with \$2. That confirms you have successfully transitioned from md5 to Blowfish encryption. The last field, which refers to the user login, should be the respective shell. This would be /usr/local/bin/bash if you were using the bash shell. For users who should not have logon rights, such as the www user for Apache, reconfirm that the shell is :/sbin/nologin.

Remove the toor user created by bash

```
dd on the line that contains the toor user name
```

Now make Blowfish the default encryption for all new users added.

```
vi auth.conf
```

```
crypt_default=blf
```

Now save and exit vi.

ZZ

8. sysctl.conf

```
vi /etc/sysctl.conf
```

```
net.inet.tcp.blackhole=2      # blackhole pings, traceroutes, etc.
net.inet.udp.blackhole=1
kern.ps_showallprocs=0       #only allow root to see all processes
```

9. fstab

We now want to assign certain attributes to certain mounts. For instance, there is probably no reason why anyone should be able to execute a program in /tmp.

There are more severe degrees to take the fstab file. You could only make some mounts read-only, such as /usr/local, which would mean that with any software installs or cvs updates, you would have a backup less restrictive fstab file. For our purposes, we'll just set some options that we won't need to change on a regular basis.

For our purposes, we'll set the following under the options section of the fstab file.

#Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/ad0s1b	none	swap	sw	0	0
/dev/ad0s1a	/	ufs	rw	1	1
/dev/ad0s1f	/tmp	ufs	rw,noexec	2	2
/dev/ad0s1g	/usr	ufs	rw	2	2
/dev/ad0s1h	/usr/home	ufs	rw,nosuid,noexec	2	2
/dev/ad0s1i	/var	ufs	rw,noexec	2	2
/dev/fd0	/floppy	MSDOS	rw,noauto,noexec,nosuid,nodev,noatime	0	0
/dev/acd0c	/cdrom	cd9660	ro,noauto	0	0
proc	/proc	procfs	rw	0	0

Here is a list of options to set:

- ro: for read-only
- rw: read-write, default
- sw: swap
- nosuid: no suid can take effect
- noexec: no execution of files
- nodev: disallow files to be viewed as a device
- noauto: not mounted during boot sequence
- noatime: prevents the filesystem from recording the file access time

10. CVSup

To keep the source code and documentation updated, we will have CVSup running regularly as setup in the [cron jobs section](#). You will notice that we are not using the ports. We are only looking at dealing with the necessary source updates.

```
[user@server /dir]# cp /usr/share/examples/cvsup/stable-supfile /root
```

```
[user@server /dir]# cd /root
```

```
[user@server /dir]# vi stable-supfile
```

Change the following lines:

```
*default host=                                #go to http://www.freebsd.org/doc/en\_US.ISO8859-1/books/handbook/cvsup.html to find your CVSup mirror, tested with pings, and enter it here. This is line 68 in the file.
*default release=cvs                          #assuming of course, that's what you're using. This is line 73 in the file.
tag=RELENG_4_7                                the file.
#src-all                                       #hash out this since we at least don't want src-games. This is line 84 in the file.
```

In the next section of the stable-supfile, hash out the source updates you don't want. This would definitely include src-games. Check this bibliography below for more information.

11. Cron Jobs

```
chmod 0600 /etc/crontab                       # only root should be able to view or change cron jobs
touch /var/cron/deny                          # add all users to deny cron jobs to, except root, of course
chmod 0600 /var/cron/deny
```

Now let's add a couple of jobs to crontab that run regularly.

```
vi /etc/crontab
```

```
0 2 * * *      root    /usr/libexec/locate.updatedb      # update the locate database
                                     every morning at 2 am.
0 2 * * *      root    /usr/local/sbin/rdate yourNTPserver # run rdate every morning at 2
                                     am
1 3 * * *      root    /usr/local/sbin/chkrootkit        # run chkrootkit every month
```

12. Kernel Changes

Make the following changes to your kernel and rebuild it. You can also check these options in the /usr/src/sys/i386/conf/LINT file.

```
#pseudo-device      bpf                #Berkley Packet Filter. Kernel-l
                                     sniffers. Don't do this if you're u
options              SC_NO_HISTORY          # disable backscrolling in virtua
options              SC_DISABLE_REBOOT      # no ctl-alt-del
options              SC_DISABLE_DDBKEY     # disable debug key
options              TCP_DROP_SYNFIN       # see above in rc.conf.
```

options	RANDOM_IP_ID	# randomize IP ID to prevent se idlescan-style portscanning. Pre the rate of packet generation.
options	ICMP_BANDLIM	# If you do decide to allow icm limit the bandwidth in respondi against denial of service attacks

13. File Permissions

With `chmod 600`, we are only allowing root to read and write the affected file. With `chmod 700`, we are also giving root the ability to execute the file.

```
chmod 0700 /root
```

```
chmod 0600 /etc/syslog.conf
```

```
chmod 0600 /etc/rc.conf
```

```
chmod 0600 /etc/newsyslog.conf
```

```
chmod 0600 /etc/hosts.allow
```

```
chmod 0600 /etc/login.conf
```

```
chmod 0700 /usr/home/*
```

14. Network Time Protocol

As you can see from our crontab file, we will be using `rdate` instead of `ntp` for keeping our local time synchronized with the atomic clock. As an aside, maintaining accurate time is crucial for logging and keeping system integrity. Being off by a little isn't a problem, but being precise only helps in troubleshooting and forensics.

```
vi /etc/ntp.conf
```

```
restrict default ignore #don't service ntp requests as a server
```

15. TCP Wrappers

```
vi /etc/hosts.allow
```

```
sshd : localhost : allow
```

```
sshd : x.x.x.x, x.x.x.x : allow #allow ssh requests only from x.x.x.x
```

```
sshd : all : deny #drop all other ssh requests
```

In the event you are remotely accessing from a box with a dynamically assigned ip address, find another secure box with a static address that you can regularly ssh to, then use the address (es) as the one (s) this box allows ssh access from.

Be sure to edit for the rest of the services running on this box. For instance, you probably want to set the following:

```
ftpd : ALL : deny
```

16. Console Access

Remember that while we are locking down the console from unauthorized single-user mode access, this does not replace physically protecting the server. Someone with less-than-noble intentions could just remove the hard drive and mount it on another machine. If a server is physically unsecured, the server's software configuration is irrelevant. **Making the first console change does mean you will NOT be able to log on in single user mode at all if you do not have the root password.**

vi /etc/ttys

```
console none unknown off insecure      # require root password when in single-user mo
ttyv0  "/usr/libexec/getty Pc"          cons25 on insecure
# Virtual terminals
ttyv1  "/usr/libexec/getty Pc"          cons25 on insecure
ttyv2  "/usr/libexec/getty Pc"          cons25 on insecure
ttyv3  "/usr/libexec/getty Pc"          cons25 on insecure
ttyv4  "/usr/libexec/getty Pc"          cons25 on insecure
ttyv5  "/usr/libexec/getty Pc"          cons25 on insecure
ttyv6  "/usr/libexec/getty Pc"          cons25 on insecure
ttyv7  "/usr/libexec/getty Pc"          cons25 on insecure
```

17. Bash Shell

```
vi /usr/share/skel/.bash_logout          # A .bash_logout file will be created in each use
                                          directory with just the phrase 'clear' in the file to
                                          console when the user logs out. Copy to root's /

clear
```

18. chflags

The chflags command increases the level of security on specified files. This command is particularly useful for binaries or configuration files, where extra code or data could infect the original file.

Consider applying chflags to binaries that are run, and to the configuration files that are critical to the server.

The command is run as below:

```
chflags [no]appnd or [no]schg filename
```

Type `ls -ol` (lower case letters o and l) to see the file properties changes. There are two flags to highlight.

- `sappnd`
This puts the file in append-only mode, and only for root. In other words, the file can have data added to it, but the original data will be maintained.
- `schg`
This command makes the file writeable only by root.

Both of these commands can have prefixes. "u" in front of the `sappnd` or `schg` applies the same properties, but adds the file's owner to the list of those with access. "no" in front of `sappnd` or `schg` will undo the `chflag` properties on the file.

19. Cleaning Up

Confirm that all `/etc/inetd.conf` lines are hashed out.

```
sockstat -4 # final check to make sure unnecessary process
tcpdump -xX # while the box is running live on the internet, :
```

20. Possible Additions

- Further password restrictions, including `passwd` history, etc.
- `securelevel` on kernel
- Polish language version
- disk quotas
- essential versus optional marked in different color
- jails
- `portupgrade/cvs`
- `tidy`

Bibliography

Needless to say, the BEST resource is the [FreeBSD Handbook](#). Although it also happens to be the most under utilized.

FreeBSD Security How-To

<http://people.freebsd.org/~jkb/howto.html>

A Basic Guide to Securing FreeBSD 4.x-STABLE

<http://draenor.org/securebsd/>

How to Build a FreeBSD-STABLE Firewall with IPFILTER

<http://www.schlacter.net/public/FreeBSD-ST>

FreeBSD Security Guide	http://defcon1.org/html/Security/Secure-Guid
TCP Wrappers (TCPD) Under FreeBSD	http://flag.blackened.net/freebsd/tcpd.html
FreeBSD Handbook, 10.3 Securing FreeBSD	http://www.freebsd.org/doc/en_US.ISO8859-
FreeBSD Handbook, 10.10 OpenSSH	http://www.freebsd.org/doc/en_US.ISO8859-
FreeBSD Handbook, 18.10 NTP	http://www.freebsd.org/doc/en_US.ISO8859-
(permanently down, i think) Taking Advantage of TCP_Wrappers	http://www.freebsdzine.org/attic/199905/secu
BSD Security Fundamentals	http://www.subterrain.net/presentations/bsd_f
Establishing Good Password Policies	http://www.onlamp.com/lpt/a/bsd/2001/01/1
Rotating Log Files	http://www.onlamp.com/lpt/a/bsd/2001/06/1
Securing BSD Daemons	http://www.onlamp.com/lpt/a/bsd/2001/02/0
Securing FreeBSD	http://www.onlamp.com/lpt/a/2622
AusCERT UNIX Security Checklist v2.0	http://www.auscert.org.au/Information/Ausce
Changing the Default Password Encryption Algorithm	http://bsdvault.net/sections.php?op=viewartic
Hardening BSD	http://www.antioffline.com/deviation/bsd.htr
Building Linux and OpenBSD Firewalls	Wes Sonnenreich's www site: OpenlySecure. publisher: http://www.wiley.com/legacy/com
GIAC's GCUX Practical Assignment, Version 1.8 by Jason Lam on "Securing MySQL Server on FreeBSD 4.5"	http://www.giac.org/practical/Jason_Lam_GC
Simple Things to Improve Your System's Security	http://www.onlamp.com/pub/a/bsd/2002/10/3
Hardening BSD by sil	http://www.astalavista.com/library/hardening
FreeBSD Handbook, Obtaining FreeBSD, Appendix A.5, Using CVSup	http://www.freebsd.org/doc/en_US.ISO8859-

[Email comments, criticisms, or suggestions](#)

Thanks to all who contributed and especially those who continue to contribute.

