Konfiguration eines Servers mit FreeBSD 7.0

Von Beat Gätzi

beat@chruetertee.ch

Aktuelle Version auf: www.chruetertee.ch

Version: 28.12.07 22:53:27

Inhaltsverzeichnis

1. Einführung....3
   1.1. Gmirror (RAID 1)....3
2. Partitionen....4
3. Serielle Konsole einrichten....5
4. Passwortverschlüsselungsalgorithmus ändern....5
5. GENERIC Kernel sichern....6
6. Kernelkonfiguration....6
   6.1. Grundsätzliches....6
   6.2. Kernel anpassen....6
   6.3. Kernel backen....8
7. Konfigurationsdateien....9
   7.1. /etc/src.conf....9
   7.2. /etc/make.conf....11
   7.3. /boot/loader.conf....12
   7.4. /etc/rc.conf....12
   7.5. /etc/sysctl.conf....13
   7.6. /etc/syslog.conf....15
   7.7. /etc/ssh/sshd_config....15
   7.8. /etc/ntp.conf....15
   7.9. /etc/ttys....16
   7.10. /etc/fstab....16
   7.11. /etc/login.conf....17
   7.12. /etc/vi.exrc....18
   7.13. /etc/csh.cshrc....19
8. IPFW Firewall....24
9. pf Firewall....25
10. Dateisystem....26
   10.1. Temporäres Verzeichnis....26
   10.2. /tmp in RAM-Disk auslagern....26
   10.3. Zugriffsrechte....27
   10.4. Dateiflags....28
11. Software....28
12. Jails....28
   12.1. Erstellen 28
   12.2. in '/etc/rc.conf' einfügen 29
   12.3. für weitere jails 30
   12.4. Jail starten....31
   12.5. Anzeigen der aktiven Jails....31
   12.6. Jails kopieren....31
   12.7. Verzeichnisse des Hostsystemes in Jail mounten....31
   12.8. Jail beenden....31
13. Schöne TCSH....32
14. Aktualisierung....32
   14.1. Kernel und Userland....32
   14.2. Jails....33
   14.3. Ports....34
15. Programm löschen....34

1. Einführung

Diese Konfigurationsanleitung bezieht sich auf die Konfiguration eines FreeBSD 7.0 auf einem Server ohne graphische Oberfläche. Dies ist keine Kochbuchanleitung, sondern mehr eine Ideensammlung und Gedächtnisstütze. Also mach nur das was Du verstehst und brauchst und lass die Finger von allem anderen! Der Autor übernimmt keine Verantwortung für das was Du machst!!!

1.1. Gmirror (RAID 1)

Hier wird beispielhaft ein RAID 1 mit 2 Festplatten (ad0 und ad1) eingerichtet. Es wird von folgenden Punkten ausgegangen:

- auf ad0 befindet sich eine FreeBSD Installation
- ad1 ist mindestens genauso gross wie ad0
- der letzte Sektor auf ad0 kann **überschrieben** werden, hier werden Metainformationen von gmirror gespeichert
- auf ad1 befinden sich keine Daten, wenn doch werden diese Daten **gelöscht**

Als erstes booten wir von der livefs-CD in Sysinstall und starten unter dem Punkt *Fixit* dann *CDROM/DVD* eine Shell.

In dieser chrooten wir als erstes nach /dist:

```
# chroot /dist
```

Dann wird devfs gemountet:

```
# mount -t devfs devfs /dev
```

geom_mirror Kernelmodul laden und ad0 als Teil eines Mirrors kennzeichnen

```
# gmirror load
# gmirror label -v -b round-robin gm0 /dev/ad0
Metadata value stored on /dev/ad0.
Done.
```

ad1 zum Mirror hinzufügen

```
# gmirror insert gm0 /dev/ad1
```

Jetzt wird ad1 auf ad0 syncronisiert, dies sollte man **abwarten**. Informationen gibt es mit

```
# gmirror list
```

Wenn die Syncronisation abgeschlossen ist mounten wir Partition **a** auf dem Mirror nach /mnt

```
# mount /dev/mirror/gm0s1a /mnt
```

Das Kernelmodul muss natürlich beim nächsten Neustart geladen werden, also

```
# echo 'geom_mirror_load="YES"' >>/mnt/boot/loader.conf
```

Jetzt muss in der Datei */mnt/etc/fstab* jedes Vorkommen von **ad0** mit **mirror/gm0** ersetzt werden, dies geht natürlich mit einem Editor, oder schneller mit

```
# sed -i -- "s%ad0%mirror/gm0%g" /mnt/etc/fstab
```

Jetzt können wir die Shell mit zweimal *exit* verlassen, Sysinstall beenden und von Festplatte booten. Fertig!

2. Partitionen

Mountpoint	*Minimal*	*Empfohlen*
/	512 MB	1 GB
/tmp	50 MB	512 MB
swap	0 MB	1 – 2 mal RAM Grösse
/var	256 MB	1 - 2 GB
/usr	2 GB	Rest

Dies ist der "Standardfall", bedenke das je nach Anwendung des Servers diese Werte nicht stimmen müssen und einige Partitionen massiv vergrössert werden müssen, z.B. /var bei einem Datenbankserver.

Alternativ kann auch noch eine Partitionen für /home gemacht werden. Auch können die Jails in eine eigene Partition z.B. /jails oder /usr/jail erstellt werden. Eine Jail ohne installierte Programme braucht ca. 150 MB.

3. Serielle Konsole einrichten

Von http://www.freebsd.org/doc/de_DE.ISO8859-1/books/handbook/serialconsole-setup.html

Dieser Abschnitt fasst zusammen, wie Sie eine serielle Konsole einrichten. Es wird vorausgesetzt, dass Sie die Voreinstellungen verwenden und wissen, wie serielle Schnittstellen verbunden werden.

Verbinden Sie die serielle Konsole mit COM1 sowie dem Kontrollterminal.

Um die Startmeldungen der seriellen Konsole zu sehen, geben Sie als `root` Folgendes ein:

```
# echo 'console="comconsole"' >> /boot/loader.conf
```

Ändern Sie in `/etc/ttys` den Eintrag für `ttyd0` von `off` auf `on`. Zusätzlich sollten Sie den Wert `dialup` auf `vt100` ändern. Nur so wird auf der seriellen Konsole eine Eingabeaufforderung mit einer Passwortabfrage aktiviert.

Starten Sie nun das System neu, damit die serielle Konsole aktiviert wird.

4. Passwortverschlüsselungsalgorithmus ändern

Von http://wiki.bsdforen.de/index.php?title=FreeBSD_-_Server_absichern
Ursprünglich geschrieben von Highfish unter http://www.bsdforen.de/showthread.php?t=2174
Lizenzbestimmung des BSDForen.de Wikis:
http://wiki.bsdforen.de/index.php/BSDForen.de_Wiki:Lizenzbestimmungen

Du kannst den Passwortverschlüsselungsalgorithmus wechseln. Der bereits genügend sichere MD5-Algorithmus kann durch den vermutlich noch besseren Blowfish-Verschlüsselungsalgorithmus ersetzt werden. Folgende Änderungen sind notwendig:

/etc/login.conf

```
 :passwd_format=md5:\   =>   :passwd_format=blf:\ (Erstes Zeichen ist ein Leerschlag!)
```

Damit die Änderungen in der /etc/login.conf auch wirklich übernommen werden, muss folgendes Kommando ausgeführt werden:

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

/etc/auth.conf

```
# crypt_default = md5 des   =>   crypt_default = blf
```

Jetzt müssen die Passwörter der Benutzer einzeln auf den neuen Verschlüsselungsalgorithmus umgestellt werden:

```
# passwd <Benutzername>
```

Alle mit dem Blowfish-Algorithmus verschlüsselten Passwörter beginnen in der Passwörter-Datei /etc/master.passwd mit "$2".

5. GENERIC Kernel sichern

Vor dem Backen eines eigenen Kernels wird der GENERIC Kernel gesichert

```
# mkdir /boot/kernel.GENERIC
# cp -R /boot/kernel/* /boot/kernel.GENERIC
# cp /boot/device.hints /boot/device.hints.default
```

Im Falle eines Falles kann der GENERIC Kernel im Bootmenu mit folgenden Befehl gebooted werden

```
# boot /kernel.GENERIC
```

6. Kernelkonfiguration

6.1. Grundsätzliches

Zeigt Informationen zu einigen Kerneloptionen an.

```
# more /usr/src/sys/i386/conf/NOTES
# more /usr/src/sys/conf/NOTES
```

Erstellt die LINT Datei, mit allen verfügbaren Kerneloptionen und devices.

```
# cd /usr/src/sys/i386/conf/ && make LINT
```

6.2. Kernel anpassen

Führe ein

```
# dmesg | grep CPU
```

aus und entferne die Unterstützung für andere CPU Klassen. Im Falle eines

```
# dmesg | grep CPU
CPU: Intel(R) Pentium(R) 4 CPU 3.20GHz (3198.40-MHz 686-class CPU)
```

kommentiere die anderen beiden aus:

```
#cpu             I486_CPU
#cpu             I586_CPU
```

Auf Multiprozessor oder Hyperthreading Rechner muss SMP aktiviert werden:

```
options         SMP                     # Symmetric MultiProcessor Kernel
```

Falls Du kein IPv6 haben möchtest, kommentiere diese Option aus

```
#options         INET6                  # IPv6 communications protocols
#options         SCTP                    # Depends on INET6
#device          gif                    # IPv6 and IPv4 tunneling
#device          faith                  # for IPv6 and IPv4 translation
```

Falls dieser Server nicht ein NFS-Server wird, kommentiere die Option aus

```
#options         NFSSERVER              # Network Filesystem Server
```

Für IPFW Unterstützung, die standardmässig alles akzeptiert und Weiterleitung unterstützt.

```
options         IPFIREWALL
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPFIREWALL_VERBOSE
options         IPFIREWALL_VERBOSE_LIMIT=100
options         IPFIREWALL_FORWARD
```

Die pf-Firewall und das Logging für die pf-Firewall aktivieren:

```
device          pf
device          pflog
```

Es sollte jeweils nur die Unterstützung für die pf- oder die IPFW-Firewall verwendet werden.

Server als IP Gateway nutzen:

```
options         IPDIVERT
```

TTL wird nicht hinuntergezählt, um Server vor traceroute zu verstecken:

```
options         IPSTEALTH
```

Für device polling Unterstützung folgende Optionen hinzufügen. Dies ist nur auf schnellen Einprozessormaschinen sinnvoll die viel Netzwerkverkehr zu bewältigen haben. Mehr Informationen unter polling(4) und http://info.iet.unipi.it/~luigi/polling/ .

```
options         DEVICE_POLLING
options         HZ=1000          # oder 2000
```

Danach muss Polling auf den einzenen Interfaces aktiviert werden:

```
# ifconfig <Interface> polling
```

In /etc/rc.conf muss polling zum Interface hinzugefügt werden:

```
ifconfig_<Interface>="inet <IP-Adresse> netmask <Subnetzmaske> polling"
```

Automatischer Neustart in Falle einer Panik:

```
options         KDB_UNATTENDED
```

Verhindert durch das Drücken von Alt-Esc oder Ctl-Print Screen den Eintritt in den Kerneldebugger

```
options         SC_DISABLE_KDBKEY
```

Verhindert das neu starten des Systems durch Ctl-Alt-Del

```
options         SC_DISABLE_REBOOT
```

Falls keine Maus am Server angeschlossen ist

```
options         SC_NO_FONT_LOADING
options         SC_NO_CUTPASTE
options         SC_NO_SYSMOUSE
```

Bei einem System auf dem mehrere Personen arbeiten, können Dienste wie ps, sockstat und w durch das MAC Framework ein bisschen weniger auskunftsfreudig gemacht werden. Dieses Framework ist immer noch experimentell, also entscheide selber ob Du es brauchst.

```
options         MAC
options         MAC_SEEOTHERUIDS
```

Folgende Option sichert die Kernelkonfiguration im Kernel selbst:

```
options         INCLUDE_CONFIG_FILE
```

Die Konfiguration kann zu einem späteren Zeitpunkt wie folgt wiederhergestellt werden:

```
# strings -n 3 /boot/kernel/kernel | sed -n 's/^___//p' > MYKERNEL
```

Wird der Server als Webserver verwendet, kann die CPU Benutzung durch folgende Option gesenkt werden:

```
options         ACCEPT_FILTER_HTTP
```

Nicht gebrauchte Geräteunterstützungen können auch noch aus dem Kernel entfernt werden, um ihn ein bisschen schlanker zu machen.

6.3. Kernel backen

Neu modisch:

```
# cd /usr/src && make buildkernel KERNCONF=<KERNELNAME> && make
installkernel KERNCONF=<KERNELNAME>
```

Altmodisch:

```
# cd /usr/src/sys/i386/conf && config <KERNELNAME> && cd
../compile/<KERNELNAME> && make depend && make && make install
```

7. Konfigurationsdateien

7.1. /etc/src.conf

Entferne nicht gebrauchte Teile des Basissystems in der Datei, durch entfernen der Raute beim jeweiligen Eintrag. src.conf(5) gibt Auskunft über die Bedeutung der verschiedenen Einträge.

```
#WITHOUT_ACPI=              # Do not build acpiconf(8), acpidump(8)...
#WITHOUT_ASSERT_DEBUG=     # compile without the assert(3) checks
WITHOUT_ATM=               # Do not build stuff related to ATM networking
#WITHOUT_AUDIT=            # Do not build audit support
#WITHOUT_AUTHPF=           # Do not build authpf(8)
WITHOUT_BIND=              # prevent any part of BIND from being build
#WITH_BIND_LIBS=           # install BIND libraries and include files
WITHOUT_BLUETOOTH=         # Do not build Bluetooth related stuff
#WITHOUT_BOOT=             # Do not build the boot blocks and loader
#WITHOUT_BZIP2=            # Do not build contributed bzip2 software
#WITHOUT_BZIP2_SUPPORT=    # build programs without bzip2 support.
#WITHOUT_CALENDAR=         # Do not build calendar(1)
#WITHOUT_CDDL=             # Do not build code licensed under Sun's CDDL
#WITHOUT_CPP=              # Do not build cpp(1)
#WITHOUT_CRYPT=            # Do not build any crypto code.
#WITHOUT_CVS=              # Do not build CVS
#WITHOUT_CXX=              # Do not build g++(1) and related libraries
#WITHOUT_DICT=             # Do not build the Webster dictionary files
#WITHOUT_DYNAMICROOT=      # you do not want to link /(s)bin dynamically.
#WITHOUT_EXAMPLES=       # avoid installing examples to/usr/share/examples
#WITHOUT_FORTH=            # build bootloaders without Forth support
#WITHOUT_FORTRAN=          # Do not build g77(1) and related libraries
#WITHOUT_FP_LIBC=          # build libc without floating-point support
WITHOUT_GAMES=             # Do not build games
#WITHOUT_GCOV=             # Do not build the gcov(1) tool
#WITHOUT_GDB=              # Do not build gdb(1)
#WITHOUT_GNU=              # Do not build contributed GNU software
#WITHOUT_GNU_SUPPORT=      # build programs without optional GNU support
#WITHOUT_GPIB=             # Do not build GPIB bus support
#WITHOUT_GROFF=            # Do not build groff(1)
#WITH_HESIOD=              # build Hesiod support
WITHOUT_HTML=              # Do not build HTML docs
WITHOUT_I4B=               # Do  not build isdn4bsd package
#WITH_IDEA=                # build the IDEA encryption code
WITHOUT_INET6=             # Do not build stuff related to IPv6
```

```
WITHOUT_INET6_SUPPORT=     # build programs without IPv6 support
WITHOUT_INFO=              # Do not make or install info(5) files
WITHOUT_IPFILTER=          # Do not build IP Filter package
WITHOUT_IPX=               # Do not build stuff related to IPX networking
WITHOUT_IPX_SUPPORT=       # build some programs without IPX support
WITHOUT_KERBEROS=          # do not want to build Kerberos 5 (KTH Heimdal)
WITHOUT_KERBEROS_SUPPORT=  # build some programs without Kerberos support
#WITHOUT_KVM=              # Do not build the libkvm library
#WITHOUT_KVM_SUPPORT=      # Do build some programs without libkvm support
#WITHOUT_LIB32=            # On amd64, set to not build 32-bit library set
#WITHOUT_LIBKSE=         # Do not build the libkse (M:N threading) library
#WITHOUT_LIBPTHREAD=       # Do not build the libpthread libraries
#WITHOUT_LIBTHR=         # Do not build the libthr (1:1 threading) library
#WITHOUT_LOCALES=          # Do not build localization files
WITHOUT_LPR=               # Do not build lpr(1) and related programs
#WITHOUT_MAILWRAPPER=      # not build the mailwrapper(8) MTA selector
#WITHOUT_MAN=              # Do not build manual pages
WITHOUT_NCP=              # Do not build stuff related NetWare Core proto
#WITHOUT_NETCAT=           # Do not build nc(1) utility
WITHOUT_NIS=               # Do not build NIS(8) support
#WITHOUT_NLS=              # Do not build NLS catalogs
#WITHOUT_NLS_CATALOGS=     # Do not build NLS catalog support for csh(1)
#WITHOUT_NS_CACHING=     # disable name caching in the nsswitch subsystem
#WITHOUT_OBJC=             # Do not build Objective C support
#WITHOUT_OPENSSH=          # Do not build OpenSSH
#WITHOUT_OPENSSL=          # Do not build OpenSSL
#WITHOUT_PAM=              # Do not build PAM library and modules
#WITHOUT_PAM_SUPPORT=      # build some programs without PAM support
#WITHOUT_PF=               # Do not build PF firewall package
#WITHOUT_PROFILE=          # avoid compiling profiled libraries
WITHOUT_RCMDS=             # Disable building of the BSD r-commands
#WITHOUT_RCS=              # Do not build rcs(1) and related utilities
#WITHOUT_RESCUE=           # Do not build rescue(8)
#WITHOUT_SENDMAIL=         # Do not build sendmail(8)
#WITHOUT_SETUID_LOGIN=     # disable login(1) as a set-user-ID program
WITHOUT_SHAREDOCS=         # not build the 4.4BSD legacy docs
#WITHOUT_SSP=              # not build propolice SSP library
#WITHOUT_SYMVER=           # disable symbol versionin
#WITHOUT_SYSCONS=          # Do not build syscons(4) support files
#WITHOUT_TCSH=             # Do not build and install /bin/csh
```

```
#WITHOUT_TOOLCHAIN=      # Do not install programs used for program devel
#WITHOUT_USB=             # Do not build USB-related programs
#WITHOUT_ZFS=             # Do not build ZFS file system
#WITHOUT_ZONEINFO=        # Do not build the timezone database
```

7.2. /etc/make.conf

Mit der /etc/make.conf kann das Verhalten von make(1) beim Bau des Basissystemes, des Kernels und der Ports beeinflusst werden. make.conf(5) gibt Auskunft über die Bedeutung der verschiedenen Einträge.

Die Wartezeit des Bootloaders kann verkürzt werden. Die Zeit ist in Millisekunden.

```
BOOTWAIT=3000
```

Falls Du IPv6, CUPS und X11 Unterstützung beim Bauen vom Paketen nicht haben willst, so wende die folgenden 3 Zeilen an:

```
WITHOUT_IPV6=yes
WITHOUT_CUPS=yes
WITHOUT_PRINT=yes
WITHOUT_X11=yes
```

Es sind zwar nur 12 Ports die ein WITHOUT_DEBUG unterstützen, aber schaden tut es sicher nicht, also fügen wir noch folgendes an:

```
WITHOUT_DEBUG=yes
```

Und diesen noch um die Last auf dem Apache FTP's ein bisschen zu verteilen.

```
MASTER_SITE_APACHE_HTTPD?=
http://mirror.switch.ch/mirror/apache/dist/httpd/
http://mirror.switch.ch/mirror/apache/dist/httpd/ ftp://mirror.switch.ch/
mirror/apache/dist/httpd/ http://www.apache.org/dist/httpd/
http://www.eu.apache.org/dist/httpd/
```

Lass die Finger von Compilerflags wie -O2 -fast-math -funroll-loops! Dies bringt nur Probleme und definitiv keine Performance. Auf einem FreeBSD 4.9 lief scp nicht mehr, nachdem ich das Userland mit -fast-math -funroll-loops gebaut habe. Auf einem FreeBSD 5.4 beendeten Programme plötzlich mit einem Segmentation fault (core dumped), nachdem ich sie mit CPUTYPE?=pentium4m aus den Ports gebaut habe. Falls es wirklich Optimierung sein muss, verwende CFLAGS= -O -pipe und COPTFLAGS= -O -pipe, da dies durch den FreeBSD Kernel und Userland unterstützt wird.

7.3. /boot/loader.conf

Verkürzt der Countdown auf 3 Sekunden und macht den Daemon schön farbig:

```
autoboot_delay="3"
loader_logo="beastie"
```

Ob ein Server Hyperthreading besitzt, kann man wie folgt herausfinden:

```
# dmesg | grep Features
Features=0xbfebfbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,
MCA,CMOV,PAT,PSE36,CLFLUSH,DTS,ACPI,MMX,FXSR,SSE,SSE2,SS,HTT,TM,PBE>
```

Befindet sich HTT in der Liste, so kann Hyperthreading wie folgt in der loader.conf aktiviert werden:

```
machdep.hlt_logical_cpus=0
machdep.hyperthreading_allowed=1
```

Was man beachten sollte ist, dass Hyperthreading auch ein Sicherheitsrisiko sein kann: ftp://ftp.freebsd.org/pub/FreeBSD/CERT/advisories/FreeBSD-SA-05:09.htt.asc

Benötigt ein Prozess mehr als 512MB RAM ("Out of memory"-Fehler), so kann dieses Limit erhöht werden, z.B auf 1GB. Dazu muss aber mehr als 1GB Arbeitsspeicher im System vorhanden sein:

```
kern.maxdsiz="1073741824"  # 1GB max memory size
kern.dfldsiz="1073741824"  # 1GB default memory size
kern.maxssiz="134217728"   # 128MB maximum stack size
```

7.4. /etc/rc.conf

Mein Vorschlag, picke Dir raus, was Dir gefällt. rc.conf(5) gibt Auskunft über die Variablen.

```
keymap="swissgerman.iso"        # keymap in /usr/share/syscons/keymaps/*
moused_enable="NO"              # Run the mouse daemon.
sshd_enable="YES"               # Enable sshd
usbd_enable="NO"                # Run the usbd daemon.
sendmail_enable="NO"            # Run the sendmail inbound daemon
nfs_server_enable="NO"          # This host is an NFS server
nfs_client_enable="NO"          # This host is an NFS client
rpcbind_enable="NO"             # Run the portmapper service
syslogd_enable="YES"            # Run syslog daemon
syslogd_flags="-ss"         # Operate in secure mode.  Do not log
                            # messages from remote machines.  No network
                            # socket will be opened at all, which also
                            # disables logging to remote machines.
clear_tmp_enable="YES"          # Clear /tmp at startup.
#---- IPFW Firewall ----        # Entweder IPFW oder pf verwenden.
firewall_enable="YES"           # Enable firewall functionality
firewall_script="/etc/ipfwrules"# Which script to run to set up the ipfw
firewall_logging="YES"          # Set to YES to enable events logging
```

```
# ---- pf Firewall ----          # Entweder IPFW oder pf verwenden.
pf_enable="YES"                 # Set to YES to enable packet filter (pf)
pflog_enable="YES"              # Set to YES to enable pf logging
# ---- Ende pf -----
fsck_y_enable="YES"         # Set to YES to do fsck -y if the initial
                            # preen fails. Diese Option ist nicht immer
                            # zu empfehlen!
background_fsck="YES"           # Attempt to run fsck in the background
                                # where possible
kern_securelevel_enable="YES"   # kernel security level (see init(8))
kern_securelevel="3"        # range: -1..3 ; `-1' is the most insecure
                            # Note that setting securelevel to 0 will
                            # result in the system booting with
                            # securelevel set to 1, as init(8) will
                            # raise the level when rc(8) completes.
ipv6_enable="NO"                # Set to YES to set up for IPv6.
inetd_enable="NO"               # Run the network daemon dispatcher
tcp_drop_synfin="YES"           # drop TCP packets with SYN+FIN
icmp_drop_redirect="YES"        # ignore ICMP REDIRECT packets
icmp_log_redirect="YES"         # log ICMP REDIRECT packets
icmp_bmcastecho=NO              # respond to broadcast ping packets
ntpd_enable="YES"               # Run ntpd Network Time Protocol
```

7.5. /etc/sysctl.conf

Folgende Zeilen werden hinzugefügt, um das Ganze ein bisschen sicherer zu machen:

```
# Prevent users from seeing information about processes that
# are being run under another UID.
security.bsd.see_other_uids=0
# Verify packet arrives on correct interface
net.inet.ip.check_interface=1
# Generate random IP_ID's
net.inet.ip.random_id=1
# Do not send RST when dropping refused connections
net.inet.tcp.blackhole=2
# Do not send port unreachables for refused connects
net.inet.udp.blackhole=1
# Drop TCP packets with SYN+FIN set
net.inet.tcp.drop_synfin=1
# Disable the setting of the hostname from within a jail
security.jail.set_hostname_allowed=0
```

```
# Random PID modulus
kern.randompid=347
```

Wird seeotheruids aus dem MAC Framework verwendet, so kann diese Beschränkung mit folgender Variable für Benutzer der Gruppe wheel aufgehoben werden.

```
# Make an exception for credentials with a specific
# gid as their real primary group id or group set
security.mac.seeotheruids.specificgid_enabled=1
```

7.6. /etc/syslog.conf

Bei folgende Zeilen wird die Raute entfernt, da wir alle Konsolen ausgaben geloggt haben wollen:

```
console.info                                          /var/log/console.log
```

Danach ein “touch /var/log/console.log“ als root ausführen, um die Logdatei zu erstellen

7.7. /etc/ssh/sshd_config

Folgendes sollte in der sshd_config stehen:

```
Port 22
Protocol 2
LogLevel INFO
PermitRootLogin no
AllowTcpForwarding no
X11Forwarding no
Subsystem       sftp    /usr/libexec/sftp-server
AllowGroups <Benutzergruppe die Zugriff per SSH haben soll>
```

7.8. /etc/ntp.conf

Wird NTP verwendet, so können die NTP-Server in der ntp.conf definiert werden. In der rc.conf muss ntpd_enable=“YES“ gesetzt sein.

```
restrict default nomodify notrap noquery
restrict 127.0.0.1
server ch.pool.ntp.org
restrict ch.pool.ntp.org noquery nomodify
```

Soll der NTP Status in der täglichen Statusmail angezeigt werden, kann folgender Eintrag in der /etc/periodic.conf gemacht werden:

```
daily_status_ntpd_enable="YES"
```

7.9. /etc/ttys

Datei muss wie folgt geändert werden, dass sich root nicht mehr direkt anmelden kann und auch beim starten in den Singleusermode nach dem root-Passwort gefragt wird. Alle Einträge von secure auf insecure ändern.

```
console none                            unknown off insecure
#
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
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off  insecure
# Serial terminals
# The 'dialup' keyword identifies dialin lines to login, fingerd etc.
ttyd0   "/usr/libexec/getty std.9600"   dialup  off insecure
ttyd1   "/usr/libexec/getty std.9600"   dialup  off insecure
ttyd2   "/usr/libexec/getty std.9600"   dialup  off insecure
ttyd3   "/usr/libexec/getty std.9600"   dialup  off insecure
# Dumb console
dcons   "/usr/libexec/getty std.9600"   vt100   off insecure
```

7.10. /etc/fstab

/tmp und /var können mit den Optionen "nosuid,noexec" eingebunden werden. Eventuell kann auch / und /usr Schreibgeschützt einhängen werden (Option: "ro"). Wird /usr Schreibgeschützt gemountet, können die home-Verzeichnisse der Benutzer auch nicht beschrieben werden! Falls /tmp mit “noexec“ eingehängt wird, so muss diese Partition vor einem make installworld ohne diese Option neu gemounted werden, da dieses sonst fehlschlägt.

```
# Device      Mountpoint  FStype   Options                 Dump  Pass #
/dev/ad0s1a   /           ufs      ro                      1     1
/dev/ad0s1b   none        swap     sw                      0     0
/dev/ad0s1e   /var        ufs      rw,nosuid,noexec        1     2
/dev/ad0s1f   /tmp        ufs      rw,nosuid,noexec        0     2
/dev/ad0s1g   /usr        ufs      ro                      1     2
```

7.11. /etc/login.conf

Um die User in den Ressourcen zu beschränken, Datei wie folgt anpassen. Die Werte können nach belieben angepasst werden und sind dem Einsatz des Servers anzupassen. Das Einschränken der Benutzer muss nicht immer eine gute Sache sein, verwende es nur, wenn es auch nötig ist. Die Variablen sind in login.conf(5) detailliert beschrieben.

```
default:\
        :passwd_format=blf:\
        :copyright=/etc/COPYRIGHT:\
        :welcome=/etc/motd:\
        :setenv=MAIL=/var/mail/$,BLOCKSIZE=K,FTP_PASSIVE_MODE=YES:\
        :path=/sbin /bin /usr/sbin /usr/bin /usr/local/sbin
/usr/local/bin ~/bin:\
        :nologin=/var/run/nologin:\
        :charset=ISO-8859-15:\
        :lang=de_CH.ISO8859-15:\
        :cputime=1h30m:\
        :datasize=8M:\
        :stacksize=2M:\
        :memorylocked=4M:\
        :memoryuse=8M:\
        :filesize=8M:\
        :coredumpsize=8M:\
        :openfiles=24:\
        :maxproc=32:\
        :sbsize=unlimited:\
        :vmemoryuse=100M:\
        :priority=0:\
        :ignoretime@:\
        :idletime=30:\
        :umask=022:
root:\
        :ignorenologin:\
        :passwd_format=blf:\
        :copyright=/etc/COPYRIGHT:\
        :welcome=/etc/motd:\
        :setenv=MAIL=/var/mail/$,BLOCKSIZE=K,FTP_PASSIVE_MODE=YES:\
        :path=/sbin /bin /usr/sbin /usr/bin /usr/games /usr/local/sbin
/usr/local/bin /usr/X11R6/bin ~/bin:\
        :nologin=/var/run/nologin:\
        :cputime=unlimited:\
        :datasize=unlimited:\
```

```
        :stacksize=unlimited:\
        :memorylocked=unlimited:\
        :memoryuse=unlimited:\
        :filesize=unlimited:\
        :coredumpsize=unlimited:\
        :openfiles=unlimited:\
        :maxproc=unlimited:\
        :sbsize=unlimited:\
        :vmemoryuse=unlimited:\
        :priority=0:\
        :ignoretime@:\
        :idletime=30:\
        :umask=022:
```

Nach dem ändern folgenden Befehl als root ausführen

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

7.12. /etc/vi.exrc

Um den vi(1) ein bisschen umgänglicher zu machen, verwende ich folgende Optionen:

```
set verbose     # Display an error message for every error
set showmode    # Display the current editor  mode
set ruler       # Display a row/column ruler on the colon command  line.
set windowname  # Change  the window name to the current file name
set autoindent  # Automatically indent new lines.
set ignorecase  # Ignore case differences in regular expressions.
set showmatch   # Note matching { and ( for } and ) characters.
set number     # Precede each line displayed with its current line number.
```

7.13. /etc/csh.cshrc

"complete"-Teil von http://www.dotfiles.com/files/21/453_.tcshrc

Um den tcsh(1) ein bisschen umgänglicher zu machen, verwende ich folgende Optionen:

```
setenv EDITOR vi
setenv LC_ALL de_CH.ISO8859-15
setenv PACKAGESITE
ftp://ftp.freebsd.ch/pub/FreeBSD/ports/i386/packages-7-stable/Latest/
setenv PKG_SITES ftp://ftp.freebsd.ch/pub/FreeBSD/ports/i386/packages-7-
stable/

alias ls ls -GF
alias ll ls -loA
alias ssh ssh -C -c blowfish-cbc,aes128-cbc,3des-cbc

if ($?prompt) then
     set prompt = "[%B%n@%m%b] %B%~%b/> "
     if (`whoami` == "root") then
          set prompt = "[%{^[[41m%}%B%n@%m%b] %B%~%b/>%{^[[39m%} "
     endif
endif

set filec                 # filename completion
set nobeep                # beeping is completely disabled
set autocorrect           # the spell-word editor command is invoked
                          # automatically before each completion attempt
set correct = all     # commands are automatically spelling-corrected
set rmstar            # the user is prompted before `rm *' is executed.
set history = 1000        # the number of history events to save
set savehist = 1000       # how  many lines are saved before exiting
set listjobs              # all jobs are listed when a job is suspended
set noclobber             # prohibit > to existing files

# completion 1) ignores case and 2) considers periods, hyphens
# and  underscores  (`.',`-'  and `_')  to be word separators and hyphens
# and underscores to be equivalent.
set complete=enhance

# Searches for documentation on the current command
alias helpcommand man
bindkey ^[OP run-help
```

```
bindkey ^[[M run-help

# DEL:
bindkey ^[[3~ delete-char

# PAGE UP : search in history backwards for line beginning as current.
bindkey ^[[I history-search-backward
bindkey ^[[5~ history-search-backward   # for x

# PAGE DOWN : search in history forwards for line beginning as current.
bindkey ^[[G history-search-forward
bindkey ^[[6~ history-search-forward    # for x

# ------------------------------------------------------------
# Complete part from Justin Randall (jrandall AT gmail DOT com)
# http://www.dotfiles.com/files/21/453_.tcshrc
# ------------------------------------------------------------
set noglob
    # Set hosts for complete of network commands from /etc/hosts
    # and from saved SSH hosts if they exist in your home directory.
        if ( -r /etc/hosts ) then
        set hosts=(`awk '/^[1-9].*/ {print $2'} /etc/hosts`)
        endif
    if ( -r $HOME/.ssh/known_hosts ) then
         set f=`cat $HOME/.ssh/known_hosts | cut -f 1 -d \ ` >& /dev/null
         set hosts=($hosts $f)
         unset f
    endif
        set hosts=($hosts mirror.switch.ch ftp.freebsd.ch)
    complete {ping,traceroute} p/1/\$hosts/
    complete {finger,[M,m]ail} 'c/*@/$hosts/' 'p/1/u/@'
    complete ssh     'n/-l/u/' 'n/*/$hosts/'
    complete ftp     c/-/"(d i g n v)"/ n/-/\$hosts/ p/1/\$hosts/ n/*/n/
    complete nslookup   p/1/x:'<host>'/ p/2/\$hosts/
    complete xhost   c/[+-]/\$hosts/ n/*/\$hosts/
    complete {,un}alias 'p/1/a/' 'p/2/c/'
    complete bindkey 'C/*/b/'
    complete env     'c/*=/f/' 'p/1/e/=/' 'p/2/c/'
    complete {fg,bg,stop}       c/%/j/ p/1/"(%)"//
    complete {,un}limit c/-/"(h)"/ n/*/l/
```

```
  complete {,un}setenv        'p/1/e/' 'c/*:/f/'
  complete {,un}set   'c/*=/f/' 'p/1/s/=' 'n/=/f/'
  complete {c,push,pop}d  'C/*/d/'
  complete {man,which,where,whereis}      'C/*/c/'
  complete kill    'c/-/S/' 'p/1/(-)//' 'c/%/j/' \
                   'n/*/`ps -U $LOGNAME | awk '"'"'{print $1}'"'"'`/'
  complete find   n/-fstype/"(nfs 4.2)"/ n/-name/f/ \
                  n/-type/"(c b d f p l s)"/ n/-user/u/ n/-group/g/ \
               n/-exec/c/ n/-ok/c/ n/-cpio/f/ n/-ncpio/f/ n/-newer/f/ \
                  c/-/"(fstype name perm prune type user nouser \
                  group nogroup size inum atime mtime ctime exec \
                  ok print ls cpio ncpio newer xdev depth \
                  daystart follow maxdepth mindepth noleaf version \
                  anewer cnewer amin cmin mmin true false uid gid \
                  ilname iname ipath iregex links lname empty path \
                  regex used xtype fprint fprint0 fprintf \
                  print0 printf not a and o or)"/ \
                  n/*/d/
complete ifconfig 'p@1@`ifconfig -l`@' 'n/*/(range phase link netmask \
                    mtu inet up metric mediaopt down delete \
                    broadcast arp debug)/'
  complete bunzip2 'p/*/f:*.bz2/'
  complete bzip2   'n/-9/f:^*.bz2/' 'n/-d/f:*.bz2/'
  complete gzip    c/--/"(stdout to-stdout decompress uncompress \
                   force help list license no-name quiet recurse \
                   suffix test verbose version fast best)"/ \
                c/-/"(c d f h l L n q r S t v V 1 2 3 4 5 6 7 8 9 -)"/\
                   n/{-S,--suffix}/x:'<file_name_suffix>'/ \
                   n/{-d,--{de,un}compress}/f:*.{gz,Z,z,zip,taz,tgz}/ \
                   N/{-d,--{de,un}compress}/f:*.{gz,Z,z,zip,taz,tgz}/ \
                   n/*/f:^*.{gz,Z,z,zip,taz,tgz}/
  complete gunzip  c/--/"(stdout to-stdout force help list license \
                no-name quiet recurse suffix test verbose version)"/ \
                   c/-/"(c f h l L n q r S t v V -)"/ \
                   n/{-S,--suffix}/x:'<file_name_suffix>'/ \
                   n/*/f:*.{gz,Z,z,zip,taz,tgz}/
  complete tar     p/1/"(-cpzvf -xzvf -tzvf -tvf)"/\
                   p/2/f:*.{tar,tar.{gz,Z},taz,tgz}/\
                   c/[ctx]vf*/"(z O p B)"/ n/*/f/
  complete umount  c/-/"(a A f c)"/ \
```

```
                    n/*/'`mount | cut -d " " -f 3`'/
  complete dd       c/--/"(help version)"/ c/[io]f=/f/ \
                    c/conv=*,/"(ascii ebcdic ibm block unblock \
                    lcase notrunc ucase swab noerror sync)"/,\
                    c/conv=/"(ascii ebcdic ibm block unblock \
        lcase notrunc ucase swab noerror sync)"/, c/*=/x:'<number>'/ \
        n/*/"(if of conv ibs obs bs cbs files skip file seek count)"/=
  complete chown    c/--/"(changes dereference no-dereference silent \
                    quiet reference recursive verbose help version)"/ \
                    c/-/"(c f h R v -)"/ C@[./\$~]@f@ c/*[.:]/g/ \
                    n/-/u/: p/1/u/: n/*/f/
 complete chgrp   c/--/"(changes no-dereference silent quiet reference \
                    recursive verbose help version)"/ \
                    c/-/"(c f h R v -)"/ n/-/g/ p/1/g/ n/*/f/
  complete chmod    c/--/"(changes silent quiet verbose reference \
                    recursive help version)"/ c/-/"(c f R v)"/
  complete df       c/--/"(all block-size human-readable si inodes \
                    kilobytes local megabytes no-sync portability sync \
                    type print-type exclude-type help version)"/ \
                    c/-/"(a H h i k l m P T t v x)"/
  complete du       c/--/"(all block-size bytes total dereference-args \
                   human-readable si kilobytes count-links dereference \
                    megabytes separate-dirs summarize one-file-system \
                    exclude-from exclude max-depth help version"/ \
                    c/-/"(a b c D H h k L l m S s X x)"/
 complete cat     c/--/"(number-nonblank number squeeze-blank show-all \
                  show-nonprinting show-ends show-tabs help version)"/ \
                    c/-/"(A b E e n s T t u v -)"/ n/*/f/
complete mv      c/--/"(backup force interactive update verbose suffix \
                    version-control help version)"/ \
                    c/-/"(b f i S u V v -)"/ \
                    n/{-S,--suffix}/x:'<suffix>'/ \
                    n/{-V,--version-control}/"(t numbered nil existing \
                    never simple)"/ n/-/f/ N/-/d/ p/1/f/ p/2/d/ n/*/f/
  complete cp       c/--/"(archive backup no-dereference force \
                    interactive link preserve parents sparse recursive \
                   symbolic-link suffix update verbose version-control \
                    one-file-system help version)"/ \
                    c/-/"(a b d f i l P p R r S s u V v x -)"/ \
                    n/-*r/d/ n/{-S,--suffix}/x:'<suffix>'/ \
```

```
                    n/{-V,--version-control}/"(t numbered nil existing \
                    never simple)"/ n/-/f/ N/-/d/ p/1/f/ p/2/d/ n/*/f/
    complete ln     c/--/"(backup directory force no-dereference \
                   interactive symbolic suffix verbose version-control \
                    help version)"/ \
                    c/-/"(b d F f i n S s V v -)"/ \
                    n/{-S,--suffix}/x:'<suffix>'/ \
                    n/{-V,--version-control}/"(t numbered nil existing \
                    never simple)"/ n/-/f/ N/-/x:'<link_name>'/ \
                    p/1/f/ p/2/x:'<link_name>'/
    complete touch  c/--/"(date reference time help version)"/ \
                    c/-/"(a c d f m r t -)"/ \
                    n/{-d,--date}/x:'<date_string>'/ \
                    c/--time/"(access atime mtime modify use)"/ \
                    n/{-r,--file}/f/ n/-t/x:'<time_stamp>'/ n/*/f/
unset noglob
```

8. IPFW Firewall

Von http://www.bsdforen.de/showthread.php?p=30110

Das Firewallscript wird unter /etc/ipfwrules erstellt. Unter den Variablem open_tcpports und open_udpports können die offenen Ports per Komma getrennt eingetragen werden.

```
#!/bin/sh
#
# Erstmal alles saubermachen bevor wir anfangen

# Also die Regeln auf "Null" stellen
/sbin/ipfw -q -f flush

# IPFW-Kommando "Quiet"
fwcmd="/sbin/ipfw -q add"

# Das setzen unserer eigenen Variabeln
open_tcpports="22,80" # ${open_tcpports} Offene Ports
open_udpports=”7777” # ${open_udpports} Offene Ports

# Erlaubt Loopbackverbindungen
${fwcmd} 00100 allow ip from any to any via lo0

# Stateful Packet Inspection
${fwcmd} 00200 check-state

# Ack Packete ohne vorheriges Req werden geblockt
${fwcmd} 00250 deny log tcp from any to any established in

#Erlaubt alle Verbindungen welche von hier initiiert wurden
${fwcmd} 00300 allow tcp from any to any out setup keep-state
${fwcmd} 00310 allow udp from any to any out keep-state

# Erlaubt bereits bestehenden hergestellten Verbindungen offen zu bleiben
${fwcmd} 00320 allow tcp from any to any  established
${fwcmd} 00330 allow udp from any to any established



# Erlaubte Dienste die ausm Internet erreicht werden dürfen
${fwcmd} 00400 allow tcp from any to any ${open_tcpports} setup keep-
state
```

```
${fwcmd} 00410 allow udp from any to any ${open_udpports} keep-state

# Sendet RESET an alle Ident Pakete, welche auf Port 113 tcp eingehen
${fwcmd} 00500 reset log tcp from any to me 113 in

# Loggt ICMP Anfragen (echo und dest. unreachable)
${fwcmd} 00700 allow log icmp from any to any in icmptype 3
${fwcmd} 00710 allow log icmp from any to any in icmptype 8

# ICMP erlauben
${fwcmd} 00750 allow icmp from any to any

# Alles andere verbieten
${fwcmd} deny log ip from any to any
```

9. pf Firewall

Von http://www.openbsd.org/faq/pf/de/example1.html

pf-Konfiguration wird Standardmässig in /etc/pf.conf erwartet. Das externe Interface kann bei ext_if definiert werden und die zu öffnenden TCP-Ports werden bei tcp_services mit einem Komma getrennt aufgelistet.

```
# Externes Interface
ext_if="em0"

# TCP Dienste die von aussen erreichbar sind
tcp_services="{ 22, 80, 443 }"

# Zu erlaubende ICMP Meldungen
icmp_types="echoreq"

# Optionen werden die standardmäßige Antwort für block-Filterregeln
# setzen und Statistikaufzeichnungen für das externe Interface anstellen:
set block-policy return
set loginterface $ext_if

# Jegliches Filtern auf den Loopbackinterfaces zu unterbinden.
set skip on lo
# Normalisierung von Paketen
scrub in
```

```
# Standardmäßigem Blocken
block in

# Erlaube ausgehenden Verkehr
pass out keep state

# Einsatz eines Schutzes gegen gefälschte Adressen
antispoof quick for { lo $ext_if }

# Erlaubte TCP-Ports öffnen und SYN-Proxy verwenden
# (http://www.openbsd.org/faq/pf/filter.html#synproxy)
pass in on $ext_if inet proto tcp from any to ($ext_if) port
$tcp_services flags S/SA synproxy state

# ICMP-Verkehr zulassen
pass in inet proto icmp all icmp-type $icmp_types keep state
```

10. Dateisystem

10.1. Temporäres Verzeichnis

Um /var/tmp auf der gleichen Partition wie /tmp zu verwenden, führe folgende Befehle aus:

```
# mv /var/tmp/* /tmp
# rm -rf /var/tmp
# ln -s /tmp /var/tmp
```

10.2. /tmp in RAM-Disk auslagern

Besitzt man einen Server der viel auf die /tmp Partition zugreift, die darauf abgelegten Daten jedoch nach einem Neustart nicht benötigt werden, so kann man die /tmp Partition aus Perforamcegründen in eine RAM-Disk auslagern. Dazu müssen folgende Einträge in die /etc/rc.conf eingetragen werden:

```
tmpmfs="YES"
tmpsize="128m"
```

Die Grössen der RAM-Disk kann in der tmpsize Variable angegeben werden. Ein Eintrag in der /etc/fstab wird danach für die /tmp Partition nicht mehr benötigt.

10.3. Zugriffsrechte

Um die Zugriffsrechte der Benutzer einzuschränken, kann folgendes Skript verwendet werden. Auf einem System mit nur vertrauenswürdigen Benutzern ist dies nicht notwendig.

```
#!/bin/sh
chmod 600 /var/log/*
chmod 700 /root
chmod 700 /home/*
echo "root" > /var/cron/allow
echo "root" > /var/at/at.allow
chmod o= /etc/crontab
chmod o= /usr/bin/crontab
chmod o= /usr/bin/at
chmod o= /usr/bin/atq
chmod o= /usr/bin/atrm
chmod o= /usr/bin/batch
chmod o= /etc/fstab
chmod o= /etc/ftpusers
chmod o= /etc/group
chmod o= /etc/hosts
chmod o= /etc/hosts.allow
chmod o= /etc/hosts.equiv
chmod o= /etc/hosts.lpd
chmod o= /etc/inetd.conf
chmod o= /etc/login.access
chmod o= /etc/login.conf
chmod o= /etc/newsyslog.conf
chmod o= /etc/rc.conf
chmod o= /etc/ssh/sshd_config
chmod o= /etc/sysctl.conf
chmod o= /etc/syslog.conf
chmod o= /etc/ttys
chmod o= /usr/bin/users
chmod o= /usr/bin/w
chmod o= /usr/bin/who
chmod o= /usr/bin/lastcomm
chmod o= /usr/sbin/jls
chmod o= /usr/bin/last
chmod o= /usr/sbin/lastlogin
chmod ugo= /usr/bin/rlogin
```

```
chmod ugo= /usr/bin/rsh
```

10.4. Dateiflags

Beachte Dateiflags bringen evtl. mehr Sicherheit, erschweren den Unterhalt eines Servers aber massiv. Also verwende die Dateiflags nur mit Bedacht!

```
# chflags schg /bin/*
# chflags schg /sbin/*
# chflags schg /usr/bin/*
# chflags schg /usr/sbin/*
# chflags schg /boot/kernel
# chflags schg /boot/kernel/kernel
# touch /boot.config
# chflags schg /boot.config
```

11. Software

Folgende Software lohnt sich schon standardmässig zu installieren:

/usr/ports/sysutils/cpdup	A comprehensive filesystem mirroring program
/usr/ports/ports-mgmt/pkg_cutleaves	Interactive script for deinstalling 'leaf' packages
/usr/ports/ports-mgmt/portaudit	Checks installed ports against a list of security vulnerabilities
/usr/ports/ports-mgmt/portmaster	Manage your ports without external databases or languages
/usr/ports/ports-mgmt/portsopt	Shows WITH(OUT)-knobs of a port makefile

12. Jails

12.1. Erstellen

```
# mkdir -p /usr/jail/myjail
# cd /usr/src && make buildworld && make installworld
DESTDIR=/usr/jail/myjail
# cd /usr/src/etc && make distribution DESTDIR=/usr/jail/myjail
# mount -t devfs devfs /dev
# cd /usr/jail/myjail
# ln -sf /dev/null kernel
# echo 'sshd_enable="YES"' >> /usr/jail/myjail/etc/rc.conf
# echo 'sshd_flags="-p <alternativer Port>"' >>
/usr/jail/myjail/etc/rc.conf
# echo 'sendmail_enable="NO"' >> /usr/jail/myjail/etc/rc.conf
# echo 'rpcbind_enable="NO"' >> /usr/jail/myjail/etc/rc.conf
# echo 'network_interface="“"' >> /usr/jail/myjail/etc/rc.conf
# echo ''hostname="<hostname>" >> /usr/jail/myjail/etc/rc.conf
# echo 'syslogd_enable="YES"' >> /usr/jail/myjail/etc/rc.conf
# echo 'syslogd_flags="-ss"' >> /usr/jail/myjail/etc/rc.conf
```

```
# echo 'kern_securelevel_enable="YES"' >> /usr/jail/myjail/etc/rc.conf
# echo 'kern_securelevel="3"' >> /usr/jail/myjail/etc/rc.conf
# echo "nameserver <IP des Nameservers>" >
/usr/jail/myjail/etc/resolv.conf
```

usr/jail/public/etc/ssh/sshd_config 'PermitRootLogin' auf 'no' setzen und # davor entfernen

```
# ifconfig <interface> alias <IP>/32
# jail /usr/jail/myjail  <jailname> <IP> /bin/sh
# touch /etc/fstab
# cd /etc && newaliases
```

Wurde System ohne IPv6 Unterstützung gebaut, so wird newaliases eine Fehlermeldung ausgeben. Dieses Problem wird wie folgt behoben:

```
# cd /etc/mail
# cp freebsd.mc <Vollständiger Servername mit Domain>.mc
Löschen von folgender Zeile:
DAEMON_OPTIONS(`Name=IPv6, Family=inet6, Modifiers=O')
# make
# make install
# make restart-mta
```

root Passwort, Zeitzone , Tastaturbelegung, Benutzer und Gruppen mit Hilfe von /usr/bin/sysinstall erstellen

In /etc/ssh/sshd_config ListenAddress <IP> setzen

Zeile mit “adjkerntz -a“ in der /etc/crontab mit Raute auskommentieren, da dies sonst Fehlermeldungen zur Folge hat.

Folgende Zeilen in der /etc/periodic.conf einfügen, da sonst Fehlermeldungen in den periodic Ausgaben sind. Evtl. muss die Datei zuerst angelegt werden.

daily_status_network_enable="NO"

daily_status_security_ipfwdenied_enable="NO"

daily_status_security_ipfdenied_enable="NO"

daily_status_security_pfdenied_enable="NO"

daily_status_security_ipfwlimit_enable="NO"

daily_status_security_ip6fwdenied_enable="NO"

daily_status_security_ipf6denied_enable="NO"

daily_status_security_ip6fwlimit_enable="NO"

```
# exit
```

12.2. in '/etc/rc.conf' einfügen

```
ifconfig_<interface>_alias0="inet <eine_ip_adresse> netmask 0xffffffff"
jail_enable="YES"
jail_list="public"
```

```
jail_set_hostname_allow="NO"
jail_sysvipc_allow="NO"
jail_public_rootdir="/usr/jail/public"
jail_public_hostname="<hostname>"
jail_public_ip="<die_ip_adresse_von_alias0>"
jail_public_exec="/bin/sh /etc/rc"
jail_public_devfs_enable="YES"
jail_public_devfs_ruleset="devfsrules_jail"
jail_public_mount_enable="NO"
```

12.3. für weitere jails

```
fconfig_<interface>_alias0="inet <eine_ip_adresse> netmask 0xffffffff"
ifconfig_<interface>_alias1="inet <eine_2te_ip_adresse> netmask
0xffffffff"
jail_enable="YES"
jail_list="public1 public2"
jail_set_hostname_allow="NO"
jail_sysvipc_allow="NO"
jail_public1_rootdir="/usr/jail/public1"
jail_public1_hostname="<hostname>"
jail_public1_ip="<die_ip_adresse_von_alias0>"
jail_public1_exec="/bin/sh /etc/rc"
jail_public1_devfs_enable="YES"
jail_public1_devfs_ruleset="devfsrules_jail"
jail_public1_mount_enable="NO"
jail_public2_rootdir="/usr/jail/myjail2"
jail_public2_hostname="<hostname>"
jail_public2_ip="<die_ip_adresse_von_alias1>"
jail_public2_exec="/bin/sh /etc/rc"
jail_public2_devfs_enable="YES"
jail_public2_devfs_ruleset="devfsrules_jail"
jail_public2_mount_enable="NO"
```

12.4. Jail starten

Falls Jails in der /etc/rc.conf konfiguriert sind

```
# /etc/rc.d/jail start
```

sonst

```
# jail /usr/jail/myjail <Jailname> <IP> /bin/sh /etc/rc
```

12.5. Anzeigen der aktiven Jails

```
# jls
```

12.6. Jails kopieren

```
# cpdup /usr/jail/myjail1 /usr/jail/myjail2
```

12.7. Verzeichnisse des Hostsystemes in Jail mounten

```
# mount_nullfs /usr/ports/ /usr/jails/myjail/usr/ports/
```

ACHTUNG: Dies kann eine Sicherheitslücke sein, da man Files aus dem Hostsystem in der Jail beschreibbar macht!!!

12.8. Jail beenden

Falls Jails in der /etc/rc.conf konfiguriert sind

```
# /etc/rc.d/jail stop
```

Sonst mit “jls“ die JID in Erfahrung bringen und dann mit “killall -j <JID>“ die Jail beenden.

13. Schöne TCSH

Von http://www.bsdbox.de/?page_id=6

Folgende Datei bearbeiten: /etc/csh.cshrc

Dort einfügen:

```
alias ls ls -GF
if ($?prompt) then
set prompt = "[%B%n@%m%b] %B%~%b/> "
endif
```

Danach in /root wechseln und die Datei .cshrc editieren. Dort den originalen "set prompt"-eintrag durch diesen hier ersetzen:

```
set prompt = "[%{^[[41m%}%B%n@%m%b] %B%~%b/>%{^[[39m%} "
```

14. Aktualisierung

14.1. Kernel und Userland

Zuerst die Quellcodes mittels CSup auf den aktuellen Stand bringen:

```
#  csup -L 2 /usr/sup/standard-supfile
```

Mein /usr/sup/standard-supfile:

```
*default host=cvsup.ch.FreeBSD.org
*default base=/var/db
*default prefix=/usr
*default release=cvs tag=RELENG_7_0
*default delete use-rel-suffix
*default compress
src-all
```

Danach unbedingt /usr/src/UPDATING lesen und ggf. den Anweisungen folgen. Ein komplettes Update wird wie folgt gemacht.

```
# cd /usr/src && make clean && rm -rf /usr/obj/*
# make buildworld
# make buildkernel KERNCONF=<KERNELNAME>
# make installkernel KERNCONF=<KERNELNAME>
```

Vor dem Neustart überprüfen, dass in der /etc/fstab die Partition /tmp nicht mit "noexec" eingebunden ist, da sonst ein make installworld fehlschlägt.

```
# reboot
```

In den Single-Usermode wechseln (boot -s)

```
# /sbin/adjkerntz -i
# /sbin/fsck -p
# /sbin/mount -u /
# /sbin/mount -a -t ufs
# /sbin/swapon -a
# /usr/sbin/mergemaster -p
# cd /usr/src
# make installworld
# make delete-old
# /usr/sbin/mergemaster
# /sbin/reboot
# make delete-old-libs (Fakultativ, nicht immer zu empfehlen!)
# cd /usr/src && make clean && rm -rf /usr/obj
```

14.2. Jails

Nach dem Aktualisieren des Userlands können nun die Jails direkt aus dem Hostsystem aktualisiert werden.

```
# mergemaster -p -D <Pfad zu Jail>
# make installworld DESTDIR=<Pfad zu Jail>
# mergemaster -D <Pfad zu Jail>
```

14.3. Ports

Da ich kein Freund von Aktualisierungen bin, werden nur Ports aktualisiert, die Sicherheitslücken aufweisen.

Folgender Befehl zeigt dir installierte Ports mit Sicherheitsproblemen an:

```
# portaudit -Fda
```

Portaudit befindet sich im ports-mgmt/portaudit Port.

Zuerst die den Portbaum mit portsnap auf den aktuellen Stand bringen:

```
# portsnap fetch update
```

Vor dem Update unbedingt /usr/ports/UPDATING lesen und den Anweisungen folgen!

Ist das Problem ohne Upgrade auf eine neue höhere Version des Ports zu beheben(z.B. Open-motif-2.2.3_1 auf 2.2.3_2)), so kann nur der Port mit portmaster neu gebaut und installiert werden.

```
# portmaster -b <Portname>
```

Hat die neue Version eine höhere Minor- oder Majorreleasenummer, so müssen alle vom Port abhängigen Programme neu gebaut werden.

```
# portmaster -brf <Portname>
```

Danach ggf. die Konfigurationsdateien anpassen und ggf. das Programm oder der Daemon neu starten.

Nach einem Update können nicht mehr gebrauchte Quellcode-Dateien von Ports gelöscht werden.

```
# portmaster --clean-distfiles
```

15. Programm löschen

Um ein installiertes Programm zu löschen, verwende pkg_deinstall und kein make deinstall, da ein make deinstall die Abhängigkeiten nicht überprüft und das Programm deinstalliert, obwohl andere Ports dieses noch bräuchten.

Um ein Programm mit nur von ihm genutzten Abhängigkeiten zu löschen, verwende

```
# pkg_deinstall -R -d -v <Portname>
```