Global Information Assurance Certification Paper

Building an Unix Based Exploit Launcher

By Mary M. Chaddock

Introduction

This document will provide step by step instructions for installing, configuring and securing the BSD/OS 4.2 (BSDi) operating system on a machine that will be used to review, research and test exploit code.

The hardware required is minimal. BSDi is a powerful, durable and lightweight operating system. The hardware used for this launcher is a Pentium 166MHz CPU with 96MB of RAM, one 3GB hard drive, a CD-ROM drive, and an Ethernet interface card.
The amount of disk space required for the operating system installation is less than 1 GB.

The network security administrator will use the exploit launcher to provide insight to the network traffic generated by exploits. This information will be used to configure intrusion detection systems, firewalls and other network devices to resist the attacks. Additionally, the launcher will be used to test for vulnerabilities in our network.

Step 1. Risk Analysis

Step 1.1 Identify assets.

⦸ The cost to replace the physical machine is minimal.
⦸ This system is not required for day to day operations of our company or network.
⦸ Loss of this system would be not immediately impact to our enterprise.
⦸ There are no employee or financial records on this machine.
⦸ There are no enterprise services running on this machine.[1]
⦸ The machine is not a high profile machine (email, DNS or web server).[2]
✓ The machine may contain information gathered from exploit testing.[3]
✓ Network resources are valuable and worthy of protection. An intruder could use this machine to attack other machines or networks. [4]

Step 1.2 Threats and Solutions

✓ Hardware failure - the hardware may die of old age.
Probability of occurrence: High
Solution: Backup critical information.
Probability of occurrence after applying solution: High

[1] databases, Intranet or file sharing

[2] A high profile machine is a machine that is public and normally easy to located.

[3] An attacker could use this information to identify critical machines and vulnerabilities in our network.

[4] Imagine an intruder's surprise when he discovers a bucket full of tools and exploits awaiting his arrival. It is our responsibility to insure these tools are not used to attack others.

✓ Natural Dangers - Power outages can damage or destroy computer hardware.
Probability of occurrence: High
Solution: Make use of an UPS[5] to reduce the risk.
Probability of occurrence after applying solution: Low

✓ Physical access to the machine can allow root access.
Probability of occurrence: Average
Solution: Locate the machine in a restricted area with limited access.
Probability of occurrence after applying solution: Low

✓ Attacks from outside our network
Probability of occurrence: High
Solution: Access to this machine should be restricted to our network.
Probability of occurrence after applying solution: Low

✓ Attacks from inside our network
Probability of occurrence: High[6]
Solution: Active log monitoring, immutable flags on high-risk files
Probability of occurrence after applying solution: Low to Medium[7]

✓ Malicious/ Hidden exploit code
Probability of occurrence: High
Solution: Active log monitoring, immutable flags on high-risk files
Probability of occurrence after applying solution: Medium

Step 1.3 Level of risk acceptance

Hardware failure remains a high threat, however the low cost of replacing the hardware and data make this threat an acceptable risk.

The threat of natural dangers, physical access and external network attacks are low. These are acceptable risks.

The threat of attacks from inside our network is medium. Internal attackers can normally be identified and help accountable. With increased logging and monitoring this is an acceptable risk.

---

[5] The UPS will prevent a sudden crash during a power failure that could cause fatal damage to the hard drive. The UPS will also reduce the sudden jolt of electricity that occurs when power is restored. A power jolt (surge) can damage (fry) various electronic components inside the computer.

[6] The system is located in a University network.

[7] The probability changes with the season. During summer the university population is minimal. During the fall semester, incoming freshmen increase the probability until they become familiar with our policies.

The threat of malicious and hidden exploit code is high. Use of this system will provide added protection to production systems and user workstations. Conducting the exploit tests in a controlled environment and carefully monitoring file changes makes this an acceptable risk.

Step 2. Preparing for the installation

Step 2.1 Download patches

Download patches (ftp://ftp.bsdi.com) - Take time to download all the patches you will need before you begin your installation. By writing them to a CD, you will be able to apply the patches before enabling network connectivity to the machine.

Step 2.2 Additional software

BSDi includes most of the required security tools needed, including;TCP Wrappers, SSL, OpenSSH, PGP, OpenSSL tcpdump, Samba, encryption utilities, and a Linux application platform[8].

Step 2.2.1 Download sudo

Download SUDO [9]- SUDO is not included with BSDi. SUDO allows execution of privileged commands from a non-privileged user. Accountability is essential on any system, including a single user launcher. SUDO provides for accountability. SUDO allows limited the interactive requirements for the root account.

Step 2.2.2 Download logcheck (logsentry)

LOGCHECK[10]- reviews system logs and reports unusual entries.

Step 2.4 Create a CD with patches, logcheck, and sudo.

Step 2.5 Research hardware compatibility

Verify your hardware is supported - Hardware incompatibility can delay or even prevent the operating system installation. There is a list of supported hardware and hardware configuration in the BSD/OS 4.2 Administrators Guide on page 242. A few minutes reviewing this information can save hours of frustration.

Step 2.6 Gather information

Step 2.6.1 Gather network configuration information

. IP address _____

. Hostname _____

. Domain name _____

[8] Capable of running Linux programs on BSD.
[9] http://www.sudo.ws/sudo
[10] http://www.psionic.com/products/logsentry.html

. The fully qualified hostname
(launcher.univerisity.edu) _____

. Netmask _____

. IP address of your gateway . _____

Step 2.6.2 Locate BSDi License key[11]

. License Key: _____

Step 3 General security considerations

Step 3.1 Unplug your machine from the network.

Disconnect the machine from the network during the installation to reduce the time the machine is vulnerable to an attack. The machine will be reconnected after it has been configured and patches applied.

Step 3.2 Configure firewall to block IP

Configure your enterprise firewall to block the machine from the Internet.

Step 3.3. System BIOS security considerations

Review your BIOS settings - Take a look at your BIOS settings. Don't make changes unless you know what you are doing. The BIOS on my machine offers the ability to "Boot from LAN". I disabled this.

To review your BIOS settings, place the BSD/OS CD in the drive and turn the machine on. As the machine starts, a message should display on your screen with instructions how to access your setup. My machine displayed " Press <F1> to enter SETUP". The option to display this prompt can be disabled in the BIOS settings. If you do not have documentation for your machine, the manufacturer should be able to provide this information.

Step 4 Install BSDi

Step 4.1 Boot the machine.

✓ Place the CD labeled *BSD/OS 4.2 BSDi Internet Server For x86 & Pentium Processor Systems* into the CD drive and turn the computer on.
✓ Select keyboard type.
✓ Enter your license key.

[11] The license key is not required to complete the installation. When prompted to enter your license key, you will also be given instruction to continue without a license key (type the word "none").

Step 4.2 Select express install[12]

Custom installation allows the installer to manipulate the hard drive settings, partitions and filesystems. The express installation handles this configuration automatically.

Step 4.2.1 Select No to reserve space for DOS.[13]

Step 4.2.2 Erase and reconfigure your hard drive

Type the word "erase" to Erase and Reconfigure your drive for BSD/OS. After you type the word "erase" and press the <Enter> key, the filesystems will be created, boot sectors added and software packages installed. [14]

Step 4.2.3 Enter a password for the root account.[15]

Password guidelines/restrictions will be displayed to help you select a password.

Step 4.2.4 Remove the CD and press ENTER to reboot.

Step 5. Network configuration

Step 5.1 Select manual configuration

After your machine has rebooted you will be offered three methods to use to configure your network settings; MaxIM, Manual and none.

- ✓ Enter timezone
- ✓ Enter hostname (Enter your fully qualified hostname)
- ✓ Enter IP for machine
- ✓ Enter network netmask
- ✓ Enter gateway IP
- ✓ Select network interface media type[16]

[12] If the launcher had a larger hard drive, a custom installation would be advised to create a separate filesystem for /var to isolate growing logs from critical filesystems. However manipulating the default filesystem structure on a small disk is not a wise thing to do and can easily backfire. The default configuration of the express installation creates a / (root) and /usr filesystem, a swap partition and a memory based filesystem (MFS) /tmp filesystem.
[13] The decision is not security related. There is simply not enough disk space to share.
[14] This may take a little time.
[15] If you must write the password on a piece of paper, or record it somewhere (i.e. Palm Pilot) do not add information that would identify the user (root) or the machine. Consider using a sentence, a Bible scripture, or paragraph of a poem. Consider replacing the letter o with 0 (zero) or S with 5 (the number five).

[16] If your network card provides support for more than one media type, you will be prompted to select the type. If your network card has the capability to auto-detect, select auto.

Step 5.2 Login to the machine[17]

Step 5.2.1 Configure DNS information.

Create the file new file named /etc/resolv.conf[18]
Add one line for your domain, and a line for each DNS server. For example, if my domain is university.edu and we have two names servers, 192.168.128.54 and 192.168.132.123, my /etc/resolv.conf file would have the following three lines:

(command)
```
vi /etc/resolv.conf
```

(file: /etc/resolve.conf
```
domain university.edu
nameserver 192.168.128.54
nameserver 192.168.132.123
```

Grant world read access to the /etc/resolv.conf file

(command)
```
chmod 444 /etc/resolv.conf
```

Step 5.2.2 Edit the file /etc/hosts.

The /etc/hosts file is used when there is no connection to the Internet or no access to a DNS server.

Append a line to this file to define your local machine. For example, if my IP address were 192.168.128.192, my machine name is launcher and my domain is university.edu, I would add the following line to the /etc/hosts file:

(file: /etc/hosts)
```
192.168.128.192 launcher.university.edu launcher
```

Step 6. Patches

Place your patch CD into the CD drive[19] and mount the CD drive.

(command)
```
mount /cdrom
mkdir /usr/local/patches
cp /cdrom/M420* /usr/local/patches/
```

Step 6.1 Apply the patches.

BSDi 4.2 patch files are named M420-xxx. The xxx is the number of the patch. For example, the first patch released for BSDi 4.2 is M420-001, the second is M420-002.

To apply a patch, execute the following commands:

[17] Use the root account and the password your selected.

[18] This file defines your DNS (domain name servers).

[19] This is the CD you created during your pre-installation tasks.

(commands)[20]
```
cd /usr/local/patches
./perl <patchfile> apply
```

To execute one command on all patches quickly, create the following shell script.

(command)
```
vi patchall.sh
```

(file: patchall.sh)
```
#!/bin/sh
cd /usr/local/patches
 for i in `ls M420*`
do
perl ./$i $*
done
```

Add execute permissions to patchall.sh..

(command)
```
# chmod 755 patchall.sh
```

Execute patchall.sh with the patch command you want to execute.[21]
The following command will apply all patches in /usr/local/patches and redirect the output to the file patchall.log.

(command)
```
./patchall.sh apply > patchall.log
```

Step 6.2 Rebuild and install kernel[22]

(commands)
```
cd /sys/i386/conf
cp GENERIC LOCAL
/usr/sbin/config LOCAL
cd /sys/compile/LOCAL
make clean
make depend
make
cp /bsd /bsd.save
cp bsd /bsd
```

Step 6.3 Reboot the machine.

(command)
```
reboot
```

[20] Replace <patchfile> with the name of the patch you want to apply.

[21] ./patchall.sh list - will list all files in each module, ./patchall.sh show summary - will display changed file summary.

[22] The kernel will need to be rebuilt before some patches take effect. Rebuild the kernel after all patches have been applied.

Step 7. Disable network services

The following network services are enabled by default: inetd, whod, statd, lpd, portmap, httpd-Maxim, sendmail, syslogd, and sshd.

The only network services needed for this system are syslogd and sshd.
Login to the computer

Step 7.1 Edit /etc/rc[23]

The file /etc/rc file is a bourne shell script. Lines in the file that begin with a # sign are not executed. Lines that begin with a # sign are called comments. Edit the /etc/rc file and add a # sign to the beginning of the lines that execute inetd, portmap, statd, lockd, lpd and sendmail.

The lines that execute each service are listed below.[24]

Step 7.1.1 Disable portmap

(file: /etc/rc)

```
# echo -n ' portmap';             portmap
```

Step 7.1.2 Disable statd

(file: /etc/rc)

```
# start status monitor and locking daemon if they exist
#  if [ -f /usr/libexec/statd ]; then
#      echo -n ' statd';              /usr/libexec/statd
#  fi
```

Step 7.1.3 Disable lockd

(file: /etc/rc)

```
# if [ -f /usr/libexec/lockd -a X${nfs} != X"NO" ]; then
#     echo -n ' lockd';              /usr/libexec/lockd
# fi
```

Step 7.1.4 Disable lpd

(file: /etc/rc)

```
# if [ -f /etc/printcap ]; then
#    echo -n ' printer';             lpd
# fi
```

Step 7.1.5 Disable sendmail

(file: /etc/rc)

```
# if [ -f /etc/mail/sendmail.cf -a -x /usr/sbin/sendmail ]; then
#   (cd /var/spool/mqueue; rm -f [lnx]f*)
## Build /etc/mail/aliases.db if there isn't one (req. in newer versions)
#   if [ ! -f /etc/mail/aliases.db ]; then
#     /usr/bin/newaliases >/dev/null 2>&1
```

[23] Disable the execution of inetd, portmap, sendmail, statd and lpd in the /etc/rc file.
[24] The lines are displayed after the comment sign was added.

```
#   fi
#   echo -n ' sendmail';            /usr/sbin/sendmail -bd -q30m
# fi
```

Step 7.1.6 Disable inetd

(file: /etc/rc)

```
# if [ -f /etc/inetd.conf ]; then
#   echo -n ' inetd';       inetd -u ${inetd_ignore:-internal}
# fi
```

Step 7.2 Edit /etc/netstart [25]

Step 7.2.1 Disable MaxIM

Change the maximflags value from "YES" to "NO"

(file: /etc/netstart)

```
maximflags=NO
```

Step 7.2.2 Disable rwhod

Change the rwhod flab from "-M16" to "NO"

(file: /etc/netstart)

```
rwhod=NO
```

Step 7.3 Modify kernel network settings

Network kernel settings can be compiled into the kernel or applied during system startup in the execution of /etc/rc.local.[26]

Step 7.3.1 Disable IP Forwarding

Disable forwarding source routed packets [27] by removing the # sign from the following line.[28]

(file: /etc/rc.local)

```
echo -n "source-route: "; sysctl -w net.inet.ip.forwsrcrt=0
```

Step 7.3.2 Disable icmp redirects.

Disable ICMP Redirects by adding the following line:

(file: /etc/rc.local)

```
sysctl -w net.inet.icmp.rediraccept=0
```

[25] Disable httpd-MaxIM and whod in the /etc/netstart file.

[26] Kernel settings can also be modified interactively with the sysctl -w command.

[27] Routers normally do source routing. Source routed packets are able to bypass firewalls and other network security devices.

[28] Many kernel parameters have two settings, on or off. 1 = on, and 0 = off (it's a binary thing)

Step 8. Secure remaining network services

Step 8.1 SSH

Secure Shell (ssh) provides encrypted remote access/login to the launcher from my workstation. The daemon that runs on the launcher is sshd, the command used to connect to the daemon is ssh.

There are several vulnerabilities in older versions of ssh.[29]. Ssh1 is supported and enabled in the default sshd configuration file.

Step 8.1.1 Disable SSH protocol 1 support

Change the "Protocol" setting by changing the line: *"Protocol 1,2"* to the following:

(file: /etc/sshd_config)
```
Protocol 2
```

Step 8.1.2 Modify the hostkey to use DSA.

Ssh1 uses the hostkey /etc/ssh_host_key, ssh2 uses /etc/ssh_host_**dsa**_key. If these keys are not present they will be created automatically during system startup[30] The default sshd configuration file uses /etc/ssh_host_key.

Change the HostKey setting from "*HostKey /etc/ssh_host_key*" to the following:

(file: /etc/sshd_config)
```
HostKey /etc/ssh_host_dsa_key
```

Step 8.1.3 Disable root ssh logins.

Change the "PermitRootLogin" setting from "yes" to "no".

(file: /etc/sshd_config)
```
PermitRootLogin no
```

Step 8.1.4 Enable X11Forwarding

X11Forwarding will allow you to login via ssh and execute an X application securely. Change the X11Forwarding setting from "no" to "yes"

(file: /etc/sshd_config)
```
X11Forwarding yes
```

[29] Specifically SSH1
[30] This is done from /etc/rc.

Step 8.2 syslogd

Step 8.2.1 Enable remote logging

Most logging on Unix systems is done via the syslog daemon (syslogd). Syslog may be configured to accept logging information from remote machines. The default configuration will deny logging from remote machines. This is the desired configuration for the launcher.

However the launcher will be configured to send copies of syslog entries to a remote syslog server on our network. Remote syslogging allows verification of log file integrity. Many rootkits include "log cleaners" that remove entries from local logs to hide the compromise.

Remote logging provides a second copy of the logs that can not be modified by the attacker without compromising the remote syslog server.

Syslog logs information based on facility and level. The facilities and levels used for logging are predefined in the file /usr/include/syslog.h. Syslog is not unique to Unix. Many network applications and hardware devices[31] provide syslog support. They do not run an independent syslog server, but are capable of sending log information to a syslog server.

The minimal lines required in /etc/syslog.conf file are listed below. This will log everything to /var/log/messages and to the remote syslog server.[32]

(file: /etc/syslog.conf)

```
*.emerg;authpriv.none                *
*.debug                              @192.168.28.80
*.warning;authpriv.none              root
*.warning;auth.notice;authpriv.none  /dev/console
kern.*                               /dev/console
authpriv.*                           /var/log/secure
*.debug                              /var/log/messages
```

The syslog daemon must be restarted before changes to the configuration file take effect.

(command)

```
kill -1 `cat /var/run/syslog.pid`
```

Step 9. PGP (gpg)

PGP[33] is included with BSDi. Files can be encrypted using the gpg key. Encryption will help to protect information discovered during exploit testing.

Step 8.3.1 Create a key ring[34]

(commands)

```
cd
```

[31] For example, HP networked printers and CacheOS web caching server are both capable of logging to a syslog server.
[32] Replace the IP address with the IP address of a syslog server on your network
[33] Pretty Good Privacy
[34] See appendix A for complete dialog of the gpg --key-gen command.

```
mkdir .gnupg
gpg --key-gen
```

Step 10. Crontab

Cron is scheduler application. The default configuration allows any user to schedule commands for cron to execute. Normally users do not require this service. Cron is most often used by root to perform for system maintenance. When a machine is compromised, entries may be added to cron files to activate a backdoor, sniffer or other "bad things".

According to the bsdi cron man page, if the /var/cron/allow file exists, then only users listed in that file are allowed to use cron. If there is not a /var/cron/allow file, but there is a /var/cron/deny file, then users listed in this file are not allowed to use cron.

BSDi's cron also uses the file /etc/crontab.

The /var/cron/allow and /var/cron/deny files have no effect on the execution of the commands in /etc/crontab![35]

Step 10.1 Restrict access to cron

Restrict cron access by creating an empty file named /var/cron/allow. Lock the file using the chflags command.

After the machine is fully configured we put the schg flag on the /etc/crontab file.

(commands)

```
touch /var/cron/allow
chflags schg /var/cron/allow
```

Step 10.2 Add cron entry to purge sendmail queue.

Add a cron job to flush the sendmail queue. The sendmail daemon is only required if a machine will receive incoming (non-local) mail. Email may be sent from the local machine without requiring the sendmail daemon. However, if an email message is unable to be delivered immediately,[36] the outgoing email will be placed in the local sendmail queue. If the sendmail daemon is not running on the machine, the message will not be sent again.

A cron entry should be added execute sendmail automatically and resend the queued messages.

(file: /etc/crontab)

```
30 * * * * root /usr/sbin/sendmail -q
```

Step 11. User accounts

BSDi default setting restricts the length of user passwords to 8 characters.

[35] Only root should have write access to this file
[36] This can happen if the destination host is unreachable.

Step 11.1 Activate the use of wide passwords (up to 128 characters)

BSDi defines user classes. Classes may be configured to restrict various resources[37] or define the type of authentication to use, or to allow users to login when the machine is normally in single user mode.

A user's class is listed in the fifth field of the /etc/master.passwd file. A user's class may be changed using the command "chsh".

Classes are defined in the /etc/login.conf file[38]. Locate the "default" class in this file and add the following line below the "path" setting for the default class.

(file: /etc/login.conf)
```
:widepasswords:\
```

Step 11.2 Create a user account

Create a user account that you will use to access the machine. This account is a non-privileged account.

(command)
```
adduser -m 0700 joe
```

Enter a password for the new account.
Select the default group USERS
Full name, Office, Office Phone, and Home Phone are optional
Accept default home directory
Select the shell you are most comfortable working with.

Step 11.2.1 Add the new user "joe" to the wheel group. Some commands require wheel group membership. Adding joe to the wheel group will allow him to access to read the syslog files without using the sudo.[39] Edit the /etc/group file and add joe to the first line.

(file: /etc/group)
```
wheel:*:0:root,joe
```

Step 11.3 Forward email for all accounts

Various system maintenance and security reports execute via cron. The reports are normally emailed to the root account. This information can be the first indication of a system compromise. It is important to forward this mail to an email account that will be accessed regularly.

There are two common methods used to forward email. A user can create a file in his home directory named *.forward* , or the forward can be added the file /etc/mail/aliases.

The aliases file overrides .forward files.

---

[37] For example memory and cpu.
[38] See Appendix B to view the default configuration file.

[39] There is nothing "bad" about using sudo to view logs, however I've discovered the minor inconvenience of requiring sudo to view logs can be annoying. I review logs more frequently when I have direct access without the use of sudo..

Make two changes to the /etc/mail/aliases file. Forward root email to an email address you normally use and forward user joe's email to root.[40]

(file: /etc/mail/aliases)
```
joe: root
root: joe@mail.university.edu
```

The aliases database needs to be rebuilt for the changes to take effect.

(command)
```
/usr/bin/newaliases
```

Step 12. Install Additional software

Step 12.1 SUDO

Step 12.1.1 Mount the Patch/sudo CD.

(Command)
```
mount /cdrom
```

Step 12.1.2 Copy sudo source files to hard drive.

Copy the sudo source file (sudo-1.6.5p2.tar.gz from the CD to /usr/var/tmp/

(command)
```
cp /cdrom/sudo-1.6.5p2.tar.gz /usr/var/tmp/
```

Step 12.1.3 Compile and install sudo.

(commands)
```
cd /usr/var/tmp
gunzip sudo-1.6.5p2.tar.gz
tar -xvf sudo-1.6.5p2.tar
cd sudo-1.6.5p2
./configure
make
make install
```

Step 12.1.4 visudo

Grant joe root access to the system. Add joe to the sudoers file with allow ALL access[41].

(ommand)
```
/usr/local/sbin/visudo
```

The command visudo opens the /etc/sudoers file with vi. The following line will grant Joe permission to execute any command with root privilege.

---

[40] You could enter the same email address for both root and joe, but it is easier to maintain if you point all aliases to root.

[41] This will grant full system privileges without letting him interactively login to the root account.

(file: /etc/sudoers edited via command visudo)
```
joe  ALL=(ALL) ALL
```

Step 12.2 Install logcheck

Copy the logcheck source file (logcheck-1.1.1.tar.gz from the CD to /usr/var/tmp/

(command)
```
cp /cdrom/logcheck-1.1.1.tar.gz /usr/var/tmp/
```

Step 12.2.1 Compile and install logcheck.

(commands)
```
cd /usr/var/tmp
gunzip logcheck-1.1.1.tar.gz
tar -xvf logcheck-1.1.1.tar
cd logcheck-1.1.1
make bsdos
```

Add cron entry to check logs every hour

(file: /etc/crontab)
```
0 * * * * root /bin/sh /usr/local/etc/logcheck.sh
```

Step 13. System security scripts

BSDi runs a nightly cron job (/etc/daily). This script rotates various logfiles and executes /etc/security. The rotate command will create new files with the default permissions 644, which grant world read access.

Step 13.1 /etc/daily

Step 13.1.1 Change rotate default mode

Modify the daily scripts to create files with permission setting 640 by adding "-m 640" to each line that executes the command "rotate"

(file: /etc/daily)
```
rotate -r 3 -m 640 /var/account/acct
rotate -z -r 7 -m 640 /var/log/maillog
```

Step 13.1.2 Increase the number of daily logs retained.

The log rotation configured in /etc/daily only retains 3 days. During long holiday weekend, logs can be overwritten before they can be reviewed. This machine will not produce large logs except during heavy testing. Increasing the number of logs to retain to 9 should be adequate.

(file: /etc/daily)
```
rotate -r 9 -m 640 /var/account/acct
```

Step 13.2 /etc/weekly

Step 13.2.1 Change rotate default mode

Set the mode for the "rotate" command to 640

(file:/etc/weekly)
```
rotate -z -r 3 -m 640 /var/log/messages
```

```
rotate -z -r 3 -m 640 /var/log/daemon.log
rotate -z -r 3 -m 640 /var/log/cron
rotate -z -r 3 -m 640 /var/log/ftpd/xferlog
rotate -z -r 3 -m 640 /var/account/gettyd
rotate -z -r 3 -m 640 /var/log/httpd/access_log
rotate -z -r 3 -m 640 /var/log/httpd/error_log
```

Step 14.3 /etc/monthly

Step 13.3.1 Change rotate default mode

(file: /etc/monthly)

```
rotate -z -r 5 -m 640 /var/log/$i
rotate -z -r 5 -m 640 /var/log/ftpd/ftp.log
```

Step 13.3.1 Enable connect time summary.

Uncomment the following lines in the /etc/monthly file to generate a summary of user account usage for the month

(file: /etc/montly)

```
echo ""; echo "Doing login accounting:"
ac -p | sort -nr +1
```

Step 13.4 /etc/security

The /etc/security script runs nightly[42]. This script verifies file and directory permissions, file changes, changes in binary files, looks for. rhosts files, checks password files for unusual or invalid accounts, and more. This is a wonderful script. Configuration files for the /etc/security script are in the directory /etc/mtree.
The default settings should work fine.

Step 14. Warning banners

The login banner is in /etc/motd file. This file is recreated when the system boots. Do the following to have your warning recreated when the system boots.

Create a file /etc/motd.site with your site specific warning banner.

Edit /etc/rc.local. To append your site warning banner to the motd file, add the last line below to the file /etc/rc.local. It is important that the line be placed AFTER the line "cat $T > /etc/motd.

(file /etc/rc.local)

```
T=/etc/motd.tmp
(sysctl -n kern.version | head -1; echo ""; sed '1,/^$/d' < /etc/motd) > $T
cat $T > /etc/motd
rm -f $T
cat /etc/motd.site >> /etc/motd
```

[42] It is executed from /etc/daily

Step 15. Ongoing Maintenance.

Step 16.1 Backups

System backups will not be done on this system for several reasons.

- An exploit launcher is a volatile system.
- If a compromise were suspected, there are immediate resources available for comparison and analysis on four similar servers[43]. If necessary, system files/binaries can be restored from another machine.
- It would take more time to restore the system from a DAT backup tape than it would to rebuild the system
- The alternative to using an external DAT tape drive is network backup There are few benefits to keeping backups of exploit code. The code should be used for testing, then deleted. Test results should not be kept on the launcher.
- All information that requires preserving will be moved to another machine.
- Exploit code that is found to be "safe" and may be useful in the future will be moved to another machine and archived either to CD or floppy disk.

Step 16.2 Backup tips for Lazy or Paranoid Administrators

Consider saving copies of high-risk files to another location on your system. For example, most rootkits replace the "ps" command. Copy the ps command to another directory with a different name.

(command)
```
cp /bin/ps /usr/home/joe/bin/sp
```

This command may be executed using sudo without requiring the sgid permission bit.[44]
If the ps command is modified, the sp command allows for immediate comparison.

Step 16.3 Patches

Patches are downloaded to a machine on our network, then distributed to all BSDi servers via scp and applied locally.

Step 16.4 Exploit Code

Exploits will also be downloaded in a similar manner and copied to the launcher for testing.

Step 16.5 Log Monitoring

Step 16.5.1 Logs will be reviewed locally and remotely.

Logcheck is installed locally and will check system logs every hour for suspicious activity.
Swatch is installed on our remote syslog server.

---

[43] All these servers are intel processors running the same version BSDi.

[44] The ps command is normally executed with group kmem permissions. This allows a non-priviledged user to access the information from /dev/mem. Removing the sgid from the ps command restricts usage to users that are members of the kmem group.

Step 16.5.2 Enable addition logging.

Edit the root account's login script .cshrc

(file: /root/.cshrc)
```
echo "Root is interactive on the Launcher!"|mail joe@university.edu
```

Step 16.6 System Accounting

BSDi installs and enables system accounting by default.
This information can be used to identify unusual activity on your machine.[45]

Step 17 Detecting Trojans

Exploit code can do bad things to your system. To monitor changes that may be made to your system, create a shell script to record the md5 checksum value of the high-risk files.[46]

For example to create a list of all the files in the /bin and /sbin directories, execute the following command:

(command)
```
md5 -i /bin/* /sbin/* > /root/md5list.out
```

After you test your exploit create another list and compare the two for differences.

(command)
```
md5 -i /bin/* /sbin/* > /root/md5list2.out
diff /root/md5list.out /root/md5list2.out
```

Step 17.1 Create a script to automatically check files.

(command)
```
vi /usr/local/bin/getmd5.sh
```

(file: /usr/local/bin/getmd5.sh)
```
#!/bin/sh
# syntax: getmd5.sh [start|stop]

MDDIR=/usr/local/etc
MDFILE=$MDDIR/$1.md5
case $1 in
 start)
   md5 -i /bin/* /sbin/* /usr/bin/* /usr/sbin/* usr/contrib/bin/* > $MDFILE
  ;;
  stop)
   md5 -i /bin/* /sbin/* /usr/bin/* /usr/sbin/* usr/contrib/bin/* > $MDFILE
   diff $MDDIR/start.md5 $MDDIR/stop.md5
   ;;
 *) echo "Syntax: $0 [start|stop]"
    exit 0;;
esac
```

[45] See Appendix C for examples.

[46] This is essentially the same thing tripwire does, but we are using the tools supplied by BSDi to do it.

Step 18. Additional steps for the paranoid (schg)

Use the chflags command to prevent files from being modified. A file with the schg flag set can not be modified until the flag is removed. The flag can only be removed when the system is in single user mode.

Step 18.1 Replace inetd.conf with an empty file

(command)

```
mv /etc/inetd.conf /etc/inetd.conf.dist
touch /etc/inetd.conf
chflags schg /etc/inetd.conf
```

Step 18.2 Set schg flag on commonly targeted files.

Keep a list of the flagged files[47]. Use this list to write a quick script to remove the schg flags. You will need to remove flags before applying some patches or upgrading the operating system.

/etc/rc
/etc/rc.local
/etc/rc.hardware
/etc/netstart
/etc/security
/etc/daily
/etc/weekly
/etc/monthly
/etc/mtree
/etc/crontab
/etc/passwd
/etc/hosts.equiv
/root/.profile
/root/.login
/root/.cshrc
/root/.rhosts
/root/.shosts
/etc/ftpusers
/etc/sshd_config
/etc/syslog.conf

Step 19. Verify your work

Step 19.1 Reboot your machine

Step 19.1.1 Login to the console with the root account.

☑ Did your site's warning banner display?

[47] The command "ls -alo" will display the flag settings of files.

☑ Did you get email when root logs in interactively. The log indicates the mail was "accepted for delivery". If you did not receive the email, the problem is not on the launcher.

(log entries from /var/log/maillog)

```
Feb 14 12:00:35 launcher sendmail[207]: g1EI0Yl00207: from=root, size=60, class=0,
nrcpts=1, msgid=<200202141800.g1EI0Yl00207@launcher.university.edu>,
relay=root@localhost
Feb 14 12:00:35 launcher sendmail[211]: g1EI0Yl00207: to=joe@university.edu,
ctladdr=root (0/0), delay=00:00:01, xdelay=00:00:00, mailer=esmtp, pri=30060,
relay=mail.university.edu. [192.168.121.60], dsn=2.0.0, stat=Sent (g1EI2M209108 Message
accepted for delivery)
```

Step 16.1.2 Login via ssh with the root account.

☑ SSH denies root login.

(output from client side)

```
ssh root@launcher.university.edu
root@launcher.university.edu's password:
Permission denied, please try again.
root@launcher.university.edu's password:
Permission denied, please try again.
root@launcher.university.edu's password:
Permission denied (publickey,password,keyboard-interactive).
```

((logfile entry from launcher local logs)

```
Feb 14 12:06:10 launcher sshd[214]: Failed password for ROOT from 192.168.28.212 port
1264 ssh2
Feb 14 12:06:15 launcher sshd[214]: Failed password for ROOT from 192.268.28.212 port
1264 ssh2
Feb 14 12:06:31 launcher last message repeated 1 times
Feb 14 12:06:31 launcher sshd[214]: Connection closed by 192.168.28.212
```

☑ *Verify remote logging.*

(logfile from remote syslog server)[48]

```
Feb 14 11:58:18 launcher init: kernel security level changed from 0 to 1
Feb 14 12:00:33 launcher login: ROOT LOGIN (root) ON console
Feb 14 12:55:18 launcher sshd[266]: Accepted password for joe from 192.168.28.212 port
1277 ssh2
```

☑ *Check for error messages during reboot*

(command)

```
dmesg
```

☑ *Verify the patches were applied*

Review the list of patches in /usr/src/sys/PATCHLOG

[48] I actually discovered the syslog sever was not logging the facility.level that the launcher had ssh configured for. This accounts for the time differnce in the local and remote logs. The remote log enter was from the second test.

☑ *Verify the boot kernel is the patched kernel.*

The command uname will display when the kernel was built. This date should be after the last patch listed in the PATCHLOG file.

(command)
```
| uname -a
BSD/OS launcher.university.edu 4.2 BSDI BSD/OS 4.2 Kernel #1: Thu Feb 14 13:25:53 CST 2002
root@launcher.university.edu:/usr/src/sys/compile/LOCAL  i386
```

The stat[49] command will provide additional information about the kernel file.

(command)
```
| ./stat /bsd
  File: "/bsd"
  Size: 1903926      Allocated Blocks: 3744         Filetype: Regular File
  Mode: (0555/-r-xr-xr-x)         Uid: (    0/    root)  Gid: (    0/   wheel)
Device:  3,0   Inode: 5         Links: 1
Access: Thu Feb 14 13:34:15 2002
Modify: Thu Feb 14 13:31:17 2002
Change: Thu Feb 14 13:31:17 2002
```

☑ *Verify sudo is configured correctly*

Login with joe's account and execute the following command:[50]

(command)
```
| sudo -l
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:

        #1) Respect the privacy of others.
        #2) Think before you type.

Password: *****
User joe may run the following commands on this host:
    (ALL) ALL
```

☑ *Verify sshd configuration file*

(Command)
```
| sshd -dt
debug1: Seeding random number generator
debug1: sshd version OpenSSH_3.0.2p1
debug1: read PEM private key done: type DSA
debug1: private host key: #0 type 2 DSA
```

☑ *Verify ssh version*

(command)
```
| ssh -V
```

[49] Stat is not in the BSDi basic distribution. It is included on the CD that came with the book Unix Power Tools.

[50] This is requesting joe to type in his password.

```
OpenSSH_3.0.2p1, SSH protocols 1.5/2.0, OpenSSL 0x0090581f
```

☑ *Verify syslog configuration*

I use the script syslogconf.pl[51] to verify I am logging everything.

(command)

```
./syslogconf.pl
Report for launcher.university.edu:/etc/syslog.conf

Event:  auth.alert             Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  auth.crit              Action: @192.168.28.80, root, /dev/console, /var/log/messages
Event:  auth.debug             Action:  @192.168.28.80, /var/log/messages
Event:  auth.emerg             Action:  *, @192.168.28.80, root, /dev/console, /var/log/messages
Event:  auth.err              Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  auth.info              Action:  @192.168.28.80, /var/log/messages
Event:  auth.notice            Action:  @192.168.28.80, /dev/console, /var/log/messages
Event:  auth.warning          Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  authpriv.emerg         Action:  , /var/log/secure
Event:  cron.alert             Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  cron.crit             Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  cron.debug             Action:  @192.168.28.80, /var/log/messages
Event:  cron.emerg             Action:  *, @192.168.28.80, root, /dev/console, /var/log/messages
Event:  cron.err              Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  cron.info              Action:  @192.168.28.80, /var/log/messages
Event:  cron.notice            Action:  @192.168.28.80, /var/log/messages
Event:  cron.warning          Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  daemon.alert           Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  daemon.crit            Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  daemon.debug           Action:  @192.168.28.80, /var/log/messages
Event:  daemon.emerg           Action:  *, @192.168.28.80, root, /dev/console, /var/log/messages
Event:  daemon.err             Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  daemon.info            Action:  @192.168.28.80, /var/log/messages
Event:  daemon.notice          Action:  @192.168.28.80, /var/log/messages
Event:  daemon.warning         Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  kern.alert             Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  kern.crit              Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  kern.debug             Action:  @192.168.28.80, /var/log/messages
Event:  kern.emerg             Action:  *, @192.168.28.80, root, /dev/console, /var/log/messages
Event:  kern.err               Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  kern.info              Action:  @192.168.28.80, /var/log/messages
Event:  kern.notice            Action:  @192.168.28.80, /var/log/messages
Event:  kern.warning           Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  local0.alert           Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  local0.crit            Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  local0.debug           Action:  @192.168.28.80, /var/log/messages
Event:  local0.emerg           Action:  *, @192.168.28.80, root, /dev/console, /var/log/messages
Event:  local0.err             Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  local0.info            Action:  @192.168.28.80, /var/log/messages
Event:  local0.notice          Action:  @192.168.28.80, /var/log/messages
Event:  local0.warning         Action:  @192.168.28.80, root, /dev/console, /var/log/messages
Event:  local1.alert       Action:  @192.168.28.80, root, /dev/console, /var/log/messages
<clipped>
```

☑ *Verify network services running*

(command)

```
lsof -i
```

[51] Hill, Michael, "Understanding syslog.conf.", Sys Admin Magazine December 1996

```
COMMAND PID USER   FD   TYPE     DEVICE SIZE/OFF NODE NAME
syslogd 146 root    7u  IPv4 0xf0ecbd00      0t0  UDP *:syslog
sshd    561 root    3u  IPv4 0xf10eab00      0t0  TCP *:ssh (LISTEN)
```

☑ *Verify kernel network settings*

(command)

```
| sysctl net.inet.icmp.rediraccept
net.inet.icmp.rediraccept = 0
| sysctl net.inet.ip.forwarding
net.inet.ip.forwarding = 0
```

☑ *Verify that cron is logging.*

Cron will log to the files defined in syslog. According to the information from syslogconf.pl, cron should be logging to /var/log/messages.

(command)

```
| grep cron /var/log/messages
Feb 14 14:00:00 nicolas cron[448]: (root) CMD (/bin/sh /usr/local/etc/logcheck.sh)
Feb 14 14:00:00 nicolas cron[449]: (root) CMD (/usr/bin/at)
Feb 14 14:15:00 nicolas cron[558]: (root) CMD (/usr/bin/at)
Feb 14 14:30:00 nicolas cron[600]: (root) CMD (/usr/bin/at)
Feb 14 14:45:00 nicolas cron[610]: (root) CMD (/usr/bin/at)
```

☑ Verify email forwarding.

(command)

```
| /usr/sbin/sendmail -bv -d21.0 root
joe@mail.university.edu... deliverable: mailer esmtp, host mail.university.edu., user
joe@mail.university.edu

| /usr/sbin/sendmail -bv -d21.0 joe
joe@mail.university.edu... deliverable: mailer esmtp, host mail.university.edu., user
joe@mail.university.edu
```

☑ Verify widepassword is enabled
From joe's account execute the passwd command.[52]

(command)

```
| passwd
Changing local password for bug.
New password (128 significant characters):
```

☑ Verify accounting is on by executing the command "lastcomm".

☑ List all processes with "ps -aux"

☑ Verify External firewall is blocking traffic. Execute lynx (a web browser) or ftp to ftp.bsdi.com.

[52] To leave the password unchanged, press enter when prompted for a new password.

☑ Verify schg flags are set by listing the directory "ls -alo"

Appendix A: gpg --key-gen (command output)

```
gpg (GnuPG) 1.0.3; Copyright (C) 2000 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details

Please select what kind of key you want:
(1) DSA and ElGamal (default)
(2) DSA (sign only)
(3) ElGamal (sign and encrypt)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
                    minimum keysize is  768 bits
                        default keysize is 1024 bits
       highest suggested keysize is 2048 bits
What keysize do you want? (1024)
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct (y/n)?
You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"
Real name: Joe Test
Email address: joe@university.edu
Comment: Testing for GCUX certification
 You selected this USER-ID:
 "Joe Test (Testing for GCUX certification) <joe@university.edu>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

Enter passphrase: type in a passphrase you will not forget

We need to generate a lot of random bytes. It is a good idea to perform some other action
(type on the keyboard, move the mouse, utilize the disks) during the prime generation;
this gives the random number generator a better chance to gain enough entropy.
++++++++++++++++++++++++++++++++++++++++++.++++++++++..+++++...+++++++++++++++.........
.............................................................+++++
We need to generate a lot of random bytes. It is a good idea to perform some other action
(type on the keyboard, move the mouse, utilize the disks) during the prime generation;
this gives the random number generator a better chance to gain enough entropy.
+++++++++..+..+++++++++...+++++++++++++++..+++++..++++++++++..+++++++++++++++.++++++++++
++++++++++>+++++>..+++++....<.+++++...........................+++++^^
public and secret key created and signed.
```

Appendix B - /etc/login.conf

```
#    BSDI login.conf,v 2.31 2000/01/27 23:28:10 prb Exp
#
# Sample login.conf file.
#

#
# Standard default entry.  Use compiled in system defaults.
# Enable all the various login programs
#
# krb-or-pwd First try kerberos password, then local password file
# passwd    Use only the local password file
# kerberos  Use only the kerberos password
# chpass    Do not authenticate, but change users password (change
#           the kerberos password if the user has one, else change
#           the local password)
# lchpass   Do not authenticate, but change users local password
#
# rpasswd   Password auth using /etc/rpasswd.db instead of /etc/passwd
# rchpass   Do not authenticate, but change password in /etc/rpasswd.db
#
# skey             Use S/Key MD5 authentication
# activ            ActivCard X9.9 token authentication
# crypto    CRYPTOCard X9.9 token authentication
# snk       Digital Pathways SecureNet Key authentication
# token            Generic X9.9 token authentication
#
# DO NOT ALTER the following lines:
#
auth-bsdi-defaults:auth=krb-or-pwd,kerberos,passwd,activ,crypto,snk,chpass,lchpass,token:
auth-ftp-bsdi-defaults:auth-ftp=passwd,activ,crypto,snk,token:
radius-bsdi-defaults:auth-radius=rpasswd,activ,crypto,snk,token:

#
# The default values
# To alter the default authentication types change the line:
#    :tc=auth-bsdi-defaults:\
# to be read something like: (enables passwd, "myauth", and activ)
#    :auth=passwd,myauth,activ:\
# Any value changed in the daemon class should be reset in default
# class.
#
default:\
      :path=/bin /usr/bin /usr/contrib/bin /usr/X11/bin:\
      :datasize-max=64M:\
      :datasize-cur=16M:\
      :maxproc-max=128:\
      :maxproc-cur=64:\
      :openfiles-cur=64:\
      :radius-challenge-styles=activ,crypto,skey,snk,token:\
      :stacksize-cur=2M:\
      :tc=auth-bsdi-defaults:\
      :tc=auth-ftp-bsdi-defaults:

#
```

```
# Settings used by /etc/rc and root
# This must be set properly for daemons started as root by inetd as well.
# Be sure reset these values back to system defaults in the default class!
#
daemon:\
      :path=/sbin /usr/sbin /bin /usr/bin /usr/contrib/bin /usr/X11/bin:\
      :widepasswords:\
      :ignorenologin:\
      :datasize=infinity:\
      :maxproc=infinity:\
      :openfiles-cur=128:\
      :stacksize-cur=8M:\
      :umask=022:\
      :tc=default:

#
# Settings used by rc.local to start netnews (INN)
#
news:\
      :path=/bin /usr/bin /usr/contrib/bin /var/news/bin:\
      :datasize=infinity:\
      :maxproc=256:\
      :openfiles-cur=256:\
      :stacksize-cur=16M:\
      :tc=default:

#
# To alter the list of styles, replace the line
#     :tc=radius-bsdi-defaults:
# with something like:
#     :auth-radius=passwd,myauth,rpasswd:\
#
RADIUS:\
      :tc=radius-bsdi-defaults:
#
# full default system settings
# Note that entries like "cputime" set both "cputime-cur" and "cputime-max"
# Several hard limits are determined by the system configuration (amount
# of memory and max users) and are not reflected here.
#
full-default:\
      :auth=passwd:\
      :coredumpsize=infinity:\
      :cputime=infinity:\
      :datasize-cur=16M:\
      :datasize-max=64M:\
      :filesize=infinity:\
      :maxproc-max=128:\
      :maxproc-cur=64:\
      :memorylocked-cur=10M:\
      :openfiles-cur=64:\
      :priority=0:\
      :requirehome:\
      :stacksize-cur=2M:\
      :stacksize-max=16M:\
      :term=unknown:\
      :umask=022:
```

```
#
# Staff have fewer restrictions and can login even when nologins are set
# All login methods are enabled
#
staff:\
      :datasize-cur=64M:\
      :datasize-max=infinity:\
      :ignorenologin:\
      :requirehome@:\
      :tc=default:

#
# The dialer class should be used for PPP and SLIP login account.
# This will suppress login messages
#
dialer:\
      :hushlogin:\
      :widepasswords:\
      :requirehome@:\
      :tc=default:


#
# The restricted class lowers limits
# requires the use of kerberos
# only allows 1 1/2 hours of CPU time
# The C-name entries are samples of entries, remove the C- to enable them
#
restricted:\
      :C-approve=/usr/local/bin/approve:\
      :C-copyright=/etc/copyright-local:\
      :C-shell=/usr/local/bin/meta-shell:\
      :C-welcome=/etc/motd-restricted:\
      :auth=passwd:\
      :coredumpsize=8M:\
      :cputime=1h 30m:\
      :datasize=8M:\
      :filesize=8M:\
      :maxproc=20:\
      :memorylocked=4M:\
      :memoryuse=8M:\
      :nologin=/etc/nologin-restricted:\
      :openfiles=20:\
      :path=/bin /usr/bin /usr/contrib/bin /usr/local/bin /usr/X11/bin:\
      :priority=4:\
      :requirehome:\
      :stacksize=2M:\
      :umask=0027:\
      :tc=default:
```

Appendix C - Accounting information

A summary by user.

(command)
```
| sa -m
user       cpu       re        k       avio      cnt
root        179.68 3786888.53     1.1      450.8   30448
joe           1.55   1722.67    15.0      442.6     393
nobody        1.18     26.98     8.9    18201.6      60
uucp          0.10      0.68     3.2      475.6     116
daemon        0.02    454.80     4.5      757.3      54
www           0.01      0.02     3.1       35.7      52
sys           0.00      0.00     9.0      128.0       5
```

Resources usage by command.

(command)
```
| sa
command      cpu       re        k       avio      cnt
randomd       127.31  31590.40     1.8      264.0      32
find            3.11     26.37     3.0   236163.7      35
runmod          2.18      4.91     9.2    25258.7      42
cc1             2.12      2.26    42.2      280.2     294
ssh-keygen      1.46      1.60    22.3     4117.3       3
gpg             1.31     22.17     9.7     1568.0       4
perl            1.14      3.87     6.2     2262.0     201
md5             1.13      3.10     3.2    53025.4      23
makewhatis      0.81      4.08     3.1   112725.3       6
sh              0.65 1117934.93     3.3       73.2    3560
gunzip          0.50      3.37     7.1       61.6      53
sshd            0.50  31863.47    22.5     1133.4      79
sort            0.48     27.30    46.6      186.6     179
gettyd          0.41  31598.93    11.5      772.9      13
ld              0.36      0.98    76.1     2000.6     154
ls              0.34      3.05     7.1      133.2     812
cpp             0.32      0.55     6.1      292.0     398
pax             0.32      3.48    15.5     8304.0      52
sendmail        0.30    498.13    22.7      623.8     265
du              0.30      2.03     3.2    57028.9      13
vi              0.26    257.47     8.9     1794.2     144
tr              0.24      7.30     1.4       14.7     252
tcsh            0.22 1117934.93    17.1      488.8     215
sed             0.21     21.35     1.6       23.9     741
cron            0.20  31607.47     8.8        6.4    2174
ssh             0.19      0.46    25.3     5024.0       4
locate          0.17      0.20     1.9      201.7      46
configure       0.15      2.77     4.2        2.0    3014
sa              0.14      0.33     4.3      464.4      39
syslogd         0.14  31598.93     6.8    17462.2      13
as              0.13      0.19    10.7      342.9     285
more            0.12    260.13     2.1      137.0     242
file            0.11      0.12     8.9      326.7      19
awk             0.10      0.58     4.1       53.3     406
lsof            0.09      0.13    11.0      443.5      43
gcc             0.09      2.46     2.6      381.6     352
```

```
make            0.08      3.87    11.0      557.7      14
perl5.0050      0.08      0.19     3.8     1952.0      22
apropos         0.08      0.11     2.2      212.7      65
gzip            0.08      0.33     6.8      360.9      61
<clipped>
```

sa -u
To list all the commands executed and sort by the number of times they were executed:

(command)
```
sa -u | awk '{print $1}' | sort | uniq -c | sort -n
   1 Mail
   1 adduser
   1 apmstart
   1 as
   1 calendar
   1 cc
   1 cc1
   1 checkpc
   1 chkconfig
   1 cpp
   1 csctl
   1 daily
   1 dev_mkdb
   1 dump
   1 finger
   1 getty
   1 head
   1 install
   1 jot
   1 kill
   1 kvm_mkdb
   1 lastcomm
   1 ld
   1 ldconfig
   1 netstat
   1 nm
   1 objdump
   1 od
   1 perl5.0050
   1 pwd_mkdb
   1 quotacheck
   1 quotaon
   1 recover
   1 rotate
   1 ruptime
   1 squid.dail
   1 swapon
   1 umount
   1 uuparams
   1 view
   1 w
   1 wall
   1 who
   2 ac
   2 accton
   2 dmesg
   2 getmd5.sh
   2 gettyd
   2 gunzip
   2 gzip
   2 ifconfig
   2 inetd
```

```
  2 init
  2 logger
  2 mount_mfs
  2 mtree
  2 perl
  2 scp
  2 setcons
  2 syslogd
  2 telnet
  2 wc
  2 xargs
  3 chflags
  3 cmp
  3 date
  3 fortune
  3 locate
  3 lsof
  3 mailq
  3 randomd
  3 route
  3 shutdown
  3 tail
  3 tee
  3 tset
  4 chsh
  4 df
  4 dirname
  4 find
  4 join
  4 make
  4 tar
  5 apropos
  5 asyncd
  5 m4
  5 mount
  5 syslogconf
  5 uname
  6 csh
  6 mkdir
  6 nfsiod
  6 sp
  6 sshd
  7 biff
  7 mail
  7 mesg
  7 sysctl
  8 ln
  8 ps
  9 id
  9 mv
 10 expr
 11 cput
 11 echo
 11 tr
 12 hostname
 13 uniq
 15 cat
```

```
  16 login_krb-
  16 pwd
  16 stty
  17 sort
  21 grep
  22 rm
  22 sendmail
  23 awk
  23 tcsh
  23 vi
  25 man
  28 printf
  30 at
  31 mathlink
  31 sa
  33 diff
  37 cron
  41 chown
  45 more
  46 egrep
  51 chmod
  54 sed
  56 cp
  68 ls
 141 basename
 183 sh
```

To list all the commands executed by a user

(command)

```
| sa -u | grep joe | sort | uniq -c | sort -n
   1 apropos    joe
   1 cp         joe
   1 diff       joe
   1 gunzip     joe
   1 locate     joe
   1 mail       joe
   1 man        joe
   1 sendmail   joe
   1 sh         joe
   2 getmd5.sh  joe
   2 scp        joe
   2 sp         joe
   2 tar        joe
   3 biff       joe
   3 fortune    joe
   3 hostname   joe
   3 id         joe
   3 mesg       joe
   3 pwd        joe
   3 stty       joe
   3 tset       joe
   4 vi         joe
   5 more       joe
   7 tcsh       joe
  14 ls         joe
```

To find out when joe executed gunzip:

(Command)

```
lastcomm joe | grep gunzip
gunzip     -       joe              ttyp0      0.08 secs Thu Feb 14 08:43
```

This is the report generated from the accounting summary command in the /etc/monthly script.

(command)

```
ac -p |sort -nr +1
        total     44.19
        joe               25.03
        root              19.16
```

References

BSDi. BSDOS 4.2 Administrator's Guide, v1.0, Colorado Springs: Berkeley Software Design, Inc., 2000

Garfinkel, Simson and Spafford, Gene. Practical Unix Security, Sebastopol: O'Reilly & Associates, Inc. 1991

Peek, Jerry; O'Reilly, Tim; and Loukides, Mike. Unix Power Tools, Sebastopol: O'Reilly & Associates, Inc. 1993. Page number 387.

Beck, David F., " SuSE Linux on a PowerBook G4 Workstation". December 6, 2001. URL: http://www.giac.org/practical/David_Beck_GCUX.doc