Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

Securing AIX5L, Version 5.1 on an RS/6000 E30

Austin H. Gresham, III

This purpose of this paper is to discuss the installation and configuration of an IBM RS/6000 model E30 for use as a Sendmail bastion server. The configuration of the E30 used in this paper is:

RAM: 256MB
Disk: (2) Internal 2.0GB SCSI disks
Processor:
Bus Architecture: PCI
Network: 10/100 Ethernet adapter
Console: Graphics console attached to SVGA adapter
Operating System: AIX 5L, version 5.1
Email Software: Sendmail 8.11.6
Secure Administration: OpenSSH 3.0.2

This purpose of this server is to accept incoming mail from the Internet and deliver it to a central mail server on the organization's trusted internal network. This mail server will set inside the organization's demilitarized zone, or DMZ. In it's most basic form, the DMZ is a network between the Internet and the internal network that houses public servers such as web servers, email servers, news servers and the like. Any information flowing between the DMZ and the internal network is ideally carefully controlled via firewalls and/or routing software and hardware which lessens the chance that a compromise of a server in the DMZ will lead to the compromise of the internal network. A more detailed discussion of how the Sendmail software is configured for this task is in the Sendmail Installation and Configuration section. The configuration of the DMZ and the organization's network will not be discussed in this paper: it is assumed that the reader already has a fundamental knowledge of these concepts or can pass that responsibility to another party which will ensure that the installation of this server into the DMZ is done properly.

Due to the high cost of IBM pSeries (formerly RS/6000) hardware, the machine used for this paper is refurbished and no longer sold by IBM. Please note, however, that the information provided in this document can be used on any PCI-based RS/6000 or pSeries standalone workstation or server, from Entry to

Midrange, but not on High-End or Clustered servers[1]. Applying the following steps on a MicroChannel-based RS/6000 will not yield the same results, however, some of the information provided can be used on any AIX system, regardless of architecture. In order to keep this paper focused, MicroChannel vs. PCI differences will not be discussed. If the reader chooses to use all or part of this paper for installing and securing a MicroChannel-based RS/6000 he or she does so at their own risk.

Assumptions made are that the user has already physically installed the hardware and any necessary peripherals. The physical installation of this system is outside the scope of this document. Also, until a later section, make sure that the system is <u>not</u> plugged into any network jack and is not configured with an IP address for communication via the TCP/IP protocol suite. Once the system is properly configured it will be plugged into the network and a valid IP address will be assigned.

Before installation, it is important to discuss the how physical security will play a role in this system. It is unfortunately not enough to just "harden" the operating system and the software that runs on it. Not properly securing the server from physical attack could make all the other steps in this document worthless. The first consideration should be were to place this server when it is in production. Ideally, your organization already has a secured, "raised floor," environment with the proper power, cooling, humidification and security. All these things are important for a system's well-being. Not having the proper electrical or environmental systems could cause the server to meet and untimely death, again rendering everything in this document worthless. A server that isn't running isn't a server at all!

If your organization does not already have such a facility, or you would like to make sure that what your organization provides is adequate, here are some points to consider when choosing a server location:

- Is the server in a room which has physical access controls in place? The best methods combine types of access, like photo ID badges and password controls, but even a door with a combination lock is better than nothing.

- If your server is in a raised floor environment, are there adequate structural barriers in place to prevent someone from just lifting a floor tile and "tunneling" their way into the server room?

---

[1] Please see http://www.rs6000.ibm.com for a description of the differences between Entry, Midrange, High-end and clustered servers for the purpose of this paper.

- The server should also be placed in a rack that has a locking door of some type. Even if someone gets access to the server room, a locked door on the rack may be enough to deter an attacker. Also, if the server room houses systems for multiple agencies or departments, it may make sense to lock your department's servers in a separate rack to deter curious people from "playing" with your system, which is sometimes worse than a malicious attacker. Also, if an attacker gets possession of the key to another department's rack, they won't necessarily have access to yours, too.

- Lastly does the server room provide adequate power and climate control? Ideally the server room will be serviced by a backup generator with batteries that will power all the equipment in the room until the generator comes online. Also, it is important to properly humidify a server room to lessen the risk of static electric shock, which could cause serious damage to sensitive electrical equipment. Conversely, a room with too much humidity can damage equipment too.

These are just a few points to consider when choosing a location for the server. In some environments, some of these options may not be possible but it is important to understand the risks associated with the physical placement of the server. Once you have chosen a location for this server, proceed with the installation. Again, do not attach any network cables to this server until later in the document. Doing so may invite a breach of the system even while you are installing it.

Initial Installation - Power on and NVRAM Setup

Power on the RS/6000. As the machine performs its Power-On Self Test (POST) you will see alphanumeric codes cycle in the LCD on the front of the server. After a couple of minutes, the monitor attached to the SVGA connection will sync, and you will see a graphical menu. Also, the code E1F1 will be displayed on the LCD.

On the console, you will see icons appear from left to right at the bottom of the screen. When you see the keyboard icon appear, press the F1 key on the console keyboard. After the icons finish, you will here a loud beep from the RS/6000 and you will be placed in the Systems Management Services (SMS) Menu.

Using the arrow keys on the keyboard, highlight the "Start Up" icon and press enter. At the next screen, use the arrow keys to highlight the "Default" icon and press enter. You should see a display. Make sure that the CD-ROM icon appears before any internal hard disk icons.

If, for some reason, the default bootlist shows an internal hard disk before the CD-ROM, use the arrow keys to select the "Select" icon and press enter. At the next screen, use the arrow keys to select the icon for the CD-ROM and press the spacebar to select it as the first boot device.

Use the arrow keys to select the "OK" icon and press enter. You do not need to choose any other boot devices at this time, just the CD-ROM is sufficient for the installation. During the installation of AIX, the bootlist is automatically set with whatever disk AIX has been installed on.

Continue selecting the "OK" icon at each menu you are returned to until you are back at the main SMS menu. At the main menu, use the arrow keys to select the "Tools" icon. You will be presented with a menu.

Use the arrow keys to select the "Privileged" icon. This will enable a password lock on the NVRAM settings of the RS/6000 so that SMS cannot be accessed, and, therefore, settings cannot be changed without knowing this password. The password can be up to 8 characters, and any combination of letters, numbers, and certain punctuation characters, like parentheses and the exclamation point. The reader is encouraged to use all 8 characters, use a combination of letters, numbers and punctuation, and even vary upper- and lower-case. This will make the privileged password much harder for a cracker to guess.

As you type the password, each box on the screen is filled with a key icon. If you type an invalid character, no key is displayed in that box and you are free to re-type another character. When finished typing the password, press enter. Another 8-character box is displayed, which is a confirmation of the password that was just entered. Again, press enter when finished. You are then returned to the Tools menu. Use the arrow keys to highlight the "Exit" icon and press enter.

At this point, insert the CD labeled "**AIX 5L for POWER V5.1 5765-E61, Volume 1 of 5**" into the system's CD-ROM drive. At the main SMS menu, highlight the "Exit" icon using the arrow keys and press enter. The system will continue to boot using the CD-ROM

AIX Base Installation

After the basic AIX kernel and supporting installation programs are loaded, the LCD will display code C31 and graphics console will display a screen asking you to define the system console.

Press F1 and Enter to make this graphics console the system console. At the next screen, you are asked for the language you would like to see during the installation. Choose English by pressing 1 and enter.

The next menu that will be displayed is "Welcome to Base Operating System Installation and Maintenance." Choose option 2 from this menu. At the "Installation and Settings" menu (Figure 8), press 1 and enter, which will take you to the "Change Method of Installation" menu. Again, choose option 1 and press enter. Option 1, "New and Complete Overwrite" will do exactly what it says: the disk will be formatted and a fresh copy of AIX will be installed with no data from previous installations being preserved.

At the "Change Disk(s) Where You Want to Install" menu, choose only to install to the first internal hard disk, usually called hdisk0. De-select any other internal hard disks that appear. To select or deselect any disk, press the number key corresponding to the number that appears to the left of that entry. A double ">" will appear or disappear to tell you that the disk is selected or not.

Press 0 (the number zero) and enter to return to the "Installation and Settings" menu. If all the information in option 2, "Primary Language Environment Settings (AFTER Install)" are correct, choose option 3, Advanced Options. If not, press 2 and enter to change the language settings. For this paper, all language options should be set to English and US. When finished, press 0 and enter to return to the "Installation and Settings" menu. Choose option 3, Advanced Options. At the "Advanced Options" menu, press 1 and enter repeatedly to cycle through the desktop options. Since this will be a mail server and not a workstation, we do not need to install any graphical desktops. Continue cycling through the options until "NONE" appears. Press 2 and enter until the "Enable Trusted Computing Base" option is "Yes."

Press 0 (the number zero) and enter to return to the "Installation and Settings" menu. Press 0 and enter again to begin the installation.
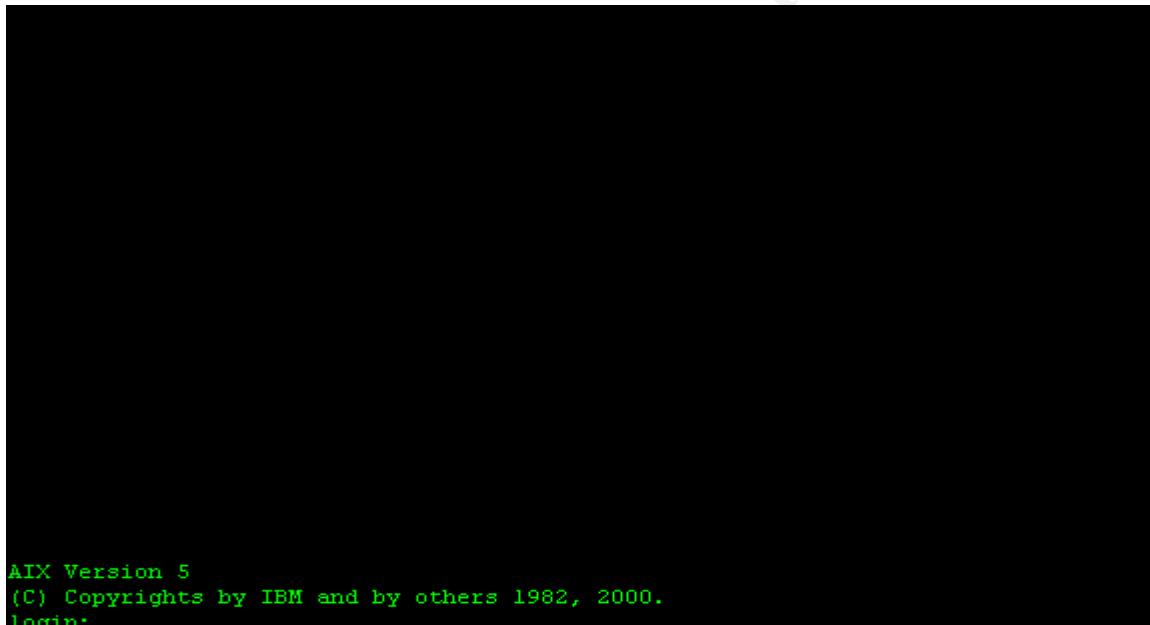
The system will now perform a base AIX installation from the CD-ROM. On the E30, a typical installation will take about an hour. You are free to do other tasks while this installation is running, you will not need to answer any additional questions or swap CD's during this portion. When the installation is finished the system will reboot. The bootlist will have been changed to only hdisk0, so if you leave the CD in the CD-ROM drive, the system will ignore it on boot up.

AIX Additional Installation

When the RS/6000 has rebooted after the initial installation, you will be presented with a graphical screen titled "Configuration Assistant." This initial

screen asks you to accept the terms of the AIX and installed software license agreements. Click "View Licenses" to review the license agreements for the software that has been installed. When you have fully reviewed the license agreements, click "Close" and you will be returned to the Configuration Assistant window. Click on "Accept" if you accept the terms of the license agreements. If you choose not to accept these agreements, you will not be able to proceed.

After clicking "Accept" you are presented with another Configuration Assistant window. Instead of using the graphical Configuration Assistant, this paper will do configuration manually, at the command prompt. Click the "Cancel" button to exit the Configuration Assistant. The graphical subsystem, X Windows, will shut down and you will be presented with a command-line login prompt like the one below:

```
AIX Version 5
(C) Copyrights by IBM and by others 1982, 2000.
login:
```

You may get some boot-up messages on the console that overwrite the login prompt. If the screen becomes garbled, just press enter and it will refresh.

At the login prompt login as the account **root**. Since this is the initial installation, root has no password, which is not a good state to be in. This is one of the reasons for keeping the system disconnected from the network until we are finished completely installing and securing the system.

The first task to be done is to choose a password for the root account. It is vital that you choose a password that is difficult to guess. The root account is the "superuser" of the AIX operating system, and a compromise of the root account is a compromise of the entire system: if an intruder has access to the root

account, they have access to everything! Choose a password of 8 characters, which is the maximum number of characters that AIX can use for passwords. Your password can be longer, but AIX will omit anything after the 8th character when creating the encrypted version of the password that is stored on disk. Also try and use combinations of letters and numbers or even punctuation marks like semicolons, periods, etc. Change the password using the passwd command:

```
# passwd
```

You are then prompted to choose a new password for the root account, and then verify that password again. When typing in the new password you will not see any output to the screen as you type, as a safety measure. As the root user you are not required to "know" the current password. The system assumes that if you are running the passwd command as the root user you have already satisfactorily authenticated as the root user. For non-root users, the passwd command asks for the current password before the user is allowed to pick a new one. The root user can also change passwords for non-root users without knowing the user's current password. By default, when root changes a user's password, a flag is set in the authentication subsystem that requires the user change their password immediately on the next login.

Next, we will stop the Configuration Assistant from starting every time we reboot. The startup of the Configuration Assistant is in the system initialization file /etc/inittab. The command to remove this entry is:

```
# rmitab install_assist
```

The next step is to set the system date, time, and time zone information. This particular server is installed in the Eastern Time zone, and uses Daylight Savings time when appropriate. AIX will know when to switch from EST to EDT, unlike some operating systems. First, change the time zone:

```
# chtz EST5EDT
```

The chtz command changes the "TZ" environment variable in the global /etc/environment file. In order for the change to take effect, log out and log back in as root. Next, we will configure the current date and time with the date command. The date command has the following syntax:

```
# date mmddHHMMccyy
```

where:
        mm   Two digit month
        dd   Two digit day
        HH   Two digit hour

MM   Two digit minute
ccyy   Four digit year

For our system, the command will be:

```
# date 032918292002
```

Make sure the output of the command is correct!

Next, we will setup the Network Time Protocol daemon (ntpd) so that the system clock is always in sync with time on the Internet. It is important that the time of the system is as accurate as possible. In the event of a system compromise, the timestamps in the system's logs may be compared with timestamps on other systems to establish an audit trail which can possibly be used to track down the source of the intruder. If the timestamps on those disparate systems don't match, it may make it more difficult, or even impossible, to establish that trail. Also, for forensic evidence to even be considered in a court of law, it is imperative that the accuracy of the timestamps in the logs can be verified: using NTP is one way of ensuring this.

Edit the file **/etc/ntp.conf**. There should already be some entries in this file, but we are only going to keep a few. Erase all the lines except the following:

```
driftfile /etc/ntp.drift
tracefile /etc/ntp.trace
```

Now, add the following line:

```
server <serverip> version 3
```

Where <*serverip*> is the IP address of an appropriate NTP server[2].

Save this file. Since we are not connected to the network yet, we will not be able to sync the time with the appropriate NTP server. We will perform that in a later step.

Finally, we will set the number of AIX user licenses so that more than two users can log in at the same time. In the past, the cost of AIX was based partly on the concept of user licenses, but this restriction has been lifted. The default installation of AIX, however, still only allows two users to log in. This includes the same user logging in via two or more different sessions, like telnet. We will change this restriction with the chlicense command. For this example, we will

---

[2] To find an appropriate NTP server on the Internet, and for a more detailed discussion of NTP, please see http://www.eecis.udel.edu/~ntp/

change the number of users allowed to log in to 50. Your installation may require more than 50 simultaneous logins, or maybe much less. The command to change the licenses to 50 is:

```
# chlicense -u 50
```

We can verify this change with the lslicense command:

```
# lslicense

Maximum number of fixed licenses is 50.
Floating licensing is disabled.
```

Now that the date, time, time zone and licensed users have been set properly, we need to reboot so that all processes, such as daemons, are aware of the new settings. Since no one else is on the system, we can shut down and reboot quickly:

```
# shutdown -Fr
```

Once the system has rebooted log back in as root.

Software to Install

Install the following software with the command:

```
# installp -acqgQNX -d /dev/cd0 bos.net.ipsec.rte
```

The IP Security fileset contains programs for securing an AIX system, which we will configure in a later section.

Software to Uninstall

The software in the next table should be uninstalled with the following command:

```
# installp -u fileset
```

Substitute the filesets in the table below for the *fileset* in the command above. If the fileset in the table has an asterisk ("*") type that as well, it serves as a wildcard for the installp command.

IMNSearch*	Tivoli_Management_Agent	rsct.msg.en_US.core.gui
rsct.core.gui	sysmgt*	Java*

X11*	bos.mh	ifor_ls
bos.net.nis*	bos.sysmgt.nim.client	bos.net.nfs*
rsct.msg.EN_US.core.gui	bos.powermgt.rte	bos.net.ncs
bos.net.snapp	bos.msg.en_US.svprint	bos.net.uucp printers*
bos.Dt	bos.X11	

The filesets in the next table should be uninstalled with this command:

```
# installp –ug fileset
```

Again, if there is an asterisk in the fileset name in the table, type the asterisk on the command line as well

bos.svprint*	bos.docregister*	bos.docsearch*	printers*	bos.txt*

Applying Current Maintenance Level Patches

Just like any other operating system, AIX has its share of bugs. When a bug is reported to, or discovered by, IBM it is assigned a number to track it, called an Authorized Program Activity Report ("APAR") number. When the bug has been fixed, the fixed filesets required to safely implement the fix are bundled together into a Program Temporary Fix ("PTF"). Each quarter, if applicable, IBM bundles all the current recommended PTF's into a Maintenance Level, which is also assigned a PTF number.   As of this writing, AIX 5L has one maintenance release, PTF number U480124.

To find and download AIX maintenance levels, use this URL from a workstation that has Internet access:

http://techsupport.services.ibm.com/server/nav?fetch=ffa5e

Click on the "New! Fix Delivery Center for AIX 5.1" link, then the "Download maintenance packages" link and follow the instructions. You can also have IBM send you the PTF's on physical media, such as CD-ROM by clicking on the "Order on physical media by PTF or APAR number" link.

For this paper we will make the assumption that the fixes are on CD-ROM, either from ordering a CD-ROM from IBM, or by "burning" a CD-ROM from another

computer that has access to the IBM Internet site above. The creation of this CD-ROM is beyond the scope of this paper, however.

Insert the CD-ROM in the CD-ROM drive. At the command prompt type:

```
# smitty update_all
```

You will be presented with the following screen:


```
                              Install Software

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]
* INPUT device / directory for software         []                        +















F1=Help              F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset          F6=Command          F7=Edit             F8=Image
F9=Shell             F10=Exit            Enter=Do
```
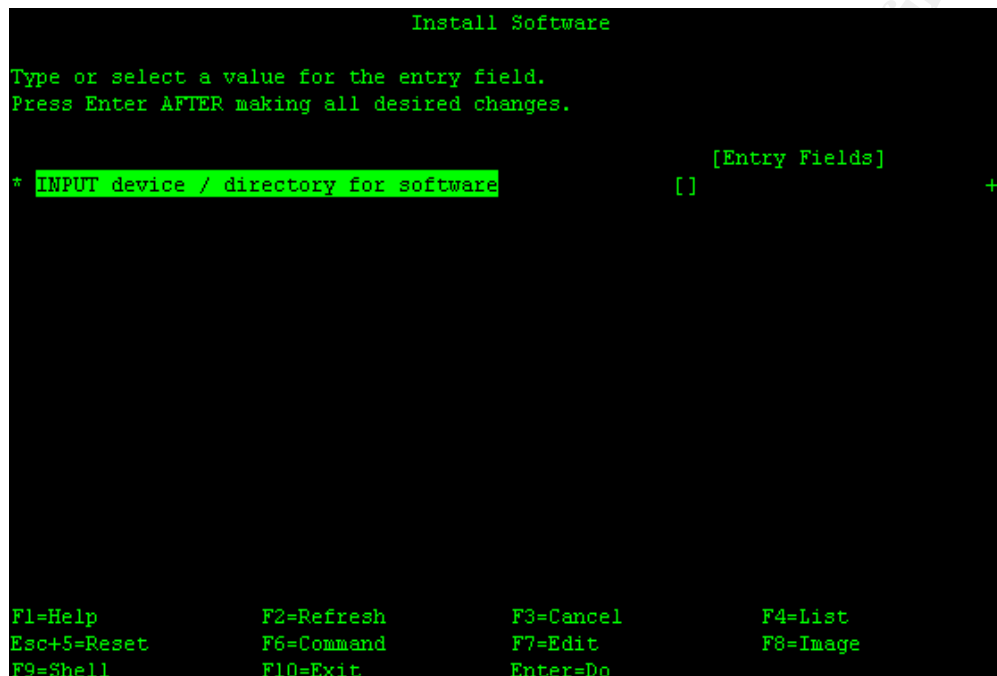
Type "/dev/cd0" in the entry box (the [ ] area) and press enter. You will see the confirmation screen:

```
                          Install Software
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* INPUT device / directory for software             /dev/cd0
* SOFTWARE to install                               [_all_latest]           +
  PREVIEW only? (install operation will NOT occur)  no                      +
  COMMIT software updates?                          yes                     +
  SAVE replaced files?                              no                      +
  AUTOMATICALLY install requisite software?         yes                     +
  EXTEND file systems if space needed?              yes                     +
  OVERWRITE same or newer versions?                 no                      +
  VERIFY install and check file sizes?              no                      +
  Include corresponding LANGUAGE filesets?          yes                     +
  DETAILED output?                                  no                      +
  Process multiple volumes?                         yes                     +
  ACCEPT new license agreements?                    no                      +
  Preview new LICENSE agreements?                   no                      +


F1=Help           F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset       F6=Command          F7=Edit             F8=Image
F9=Shell          F10=Exit            Enter=Do
```

Press enter to accept the defaults. You will be prompted with an "ARE YOU
SURE?" message. Just press enter again to proceed with the installation. As the
installation runs you will see the COMMAND STATUS window which will show you
detailed information about the running process. When finished, check the
success of the command by paging through the output in the COMMAND STATUS
window. You may see failures for "missing requisites." The maintenance release
contains fixes for all the filesets that you may have installed from the base AIX
installation media. Since we uninstalled some of those filesets in the section
above, they are not able to be patched, so a requisite failure is recorded. To
make sure that no "real" failures occurred, scroll to the bottom, and make sure all
the filesets listed have "SUCCESS" for the APPLY and "SUCCESS" for the COMMIT.
When finished, exit to a command prompt by pressing F10 and run this
command:

```
# instfix -i | grep ML
```

You should see output similar to the following:

```
All filesets for 5.0.0.0_AIX_ML were found.
All filesets for 5.1.0.0_AIX_ML were found.
All filesets for 5100-01_AIX_ML were found.
```

If any of the output begins with "Not all filesets for…" then something in the
patching process did not work. Unfortunately, this kind of problem determination
is outside the scope of this paper: we will assume that all the patches were
installed correctly.

Since the maintenance level also include some kernel fixes, we need to reboot the machine to make sure everything is up to date:

```
# shutdown -Fr
```

When the machine has rebooted, log in as the root account and proceed to the next section.

Note: Anytime you install additional software from the base AIX installation media, you must perform the steps above to make sure the current maintenance patches are applied. Failure to do so may result in an unstable system.

Adjusting the Size of Filesystems

Unlike some Unix variants, AIX does not give you the option at install time to select the size and name of filesystems. AIX will install with 6 default filesystems: /, /usr, /var, /tmp, /home, /opt. There is also a new filesystem in AIX 5L called /proc. This filesystem resides in memory and is not a filesystem on disk. Normally the default sizes of the filesystems are not adequate to handle optionally-installable components. The author recommends the following filesystem sizes for this system:

Filesystem	Size	Size in blocks
/ (root)	32 MB	65536
/usr	1 GB	2048000
/var	100 MB	204800
/tmp	100 MB	204800
/home	100 MB	204800
/opt	(remove)	(n/a)

The first thing we will do on this system is remove the /opt filesystem. For the purposes of this mail server, /opt is not needed and we could use the space for some other purposes. Run the following commands to remove /opt:

```
# umount /opt
# rmfs /opt
```

The reason for making the root ("/") filesystem so small is that software should not install directly in the root filesystem. Software which is needed at boot-time, before the other filesystems are mounted, needs to reside in directories like /bin, /sbin, and /etc, but these are usually AIX-specific programs which are installed at

the initial install and rarely change. When patches are applied, some additional space is required, but not a significant amount. Disk space, at a premium on this system, is better utilized elsewhere, like /usr where the majority of programs are installed.

When sizing filesystems in AIX, sizes are given in blocks, which are 512 bytes, or ½ of a kilobyte. The easiest way to convert Megabytes to blocks is this:

(Size in MB) * 2048 = (Size in blocks)

and to convert Gigabytes:

(Size in GB) * 2048 * 1000[3] = (Size in blocks)

Therefore, a 1GB filesystem can be represented as 2048000 blocks. The command to change the size of a filesystem is:

```
# chfs –a size=sizeinblocks filesystem
```

Where *sizeinblocks* is the new size of the filesystem in blocks, and *filesystem* is the full path to the filesystem's mount point (i.e. /usr/local). Run the chfs command for all the filesystems listed in the table above.

 When finished, output from the df command shows the new filesystem sizes:

```
# df –k

Filesystem      1024-blocks      Free %Used     Iused %Iused Mounted on
/dev/hd4            32768       25212   24%       894     6% /
/dev/hd2          1024000      784036   24%      9267     4% /usr
/dev/hd9var        102400       88872   14%       226     1% /var
/dev/hd3           102400       98944    4%        43     1% /tmp
/dev/hd1           102400       99092    4%        22     1% /home
/proc                  -           -    -          -      - /proc
```

Increasing Paging Space

The next step is to increase the size of our paging space to better match the size of this server. One commonly used method is to double the size of physical RAM[1]. For a server with 256MB of RAM, this would seem a good place to start.   Resizing or creating paging spaces is done in units of Physical Partitions.

---

[3] Note that this is not 100% accurate because there are 1024 KB in a Megabyte, and 1024 MB in a Gigabyte, however AIX will automatically round size to the nearest multiple of the Physical Partition size of a volume group, so the "1000" is to make it easier on the systems administrator for readability.

To find the Physical Partition size of a Volume Group in which a paging space is to be created, use this command:

```
# lsvg vgname | grep "PP SIZE"
```

In our case, the paging space hd6 in the rootvg Volume Group is going to be resized, so the command will be:

```
# lsvg rootvg | grep "PP SIZE"
```

The output shows:

```
VG STATE:     active        PP SIZE:            4 megabytes
```

So we see that our physical partition size is 4 MB.   From here we need to find out how large our current paging space is. The command to list paging space(s) is:

```
# lsps -a
```

This will list all paging spaces defined to the system. In this case there is only one, hd6:

```
Page Space    Physical Volume    Volume Group    Size    %Used    Active    Auto    Type
hd6           hdisk0             rootvg          64MB    6        yes       yes     lv
```

From here we can see that the default paging space is 64MB. We already know that this system has 256MB of physical RAM but, if we were not sure, how would we see this while the system is running? The bootinfo command will show us, among other things, the RAM size in kilobytes:

```
# bootinfo -r
```

The output is:

```
262144
```

If we divide 262144 by 1024, we get 256MB. If we double that we see that 512MB is required for our paging space, so we need to add 448MB to hd6. To get the number of physical partitions we need to add to hd6, divide 448 by 4, yielding 112 physical partitions. Now, to increase the size of hd6 to 512MB, use the chps command:

```
# chps -s112 hd6
```

Running the lsps command, we see that our hd6 paging space is now 512MB:

```
# lsps -a
```

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
hd6	hdisk0	rootvg	512MB	1	yes	yes	lv

Reorganize the Disk for Performance

The next step is to reorganize our filesystems on disk for the best performance. If we look at the physical layout of hdisk0, we can see that when we resized all the filesystems, we also caused some fragmentation:

```
# lspv -p hdisk0

hdisk0:
PP RANGE   STATE   REGION         LV NAME         TYPE      MOUNT
POINT
  1-2      used    outer edge     hd5             boot      N/A
  3-3      used    outer edge     hd9var          jfs       /var
  4-103    used    outer edge     hd6             paging    N/A
104-119    used    outer middle   hd6             paging    N/A
120-170    used    outer middle   hd2             jfs       /usr
171-176    used    outer middle   hd4             jfs       /
177-180    used    outer middle   hd6             paging    N/A
181-206    used    outer middle   hd2             jfs       /usr
207-207    used    center         hd8             jfslog    N/A
208-208    used    center         hd4             jfs       /
209-223    used    center         hd2             jfs       /usr
224-226    used    center         hd9var          jfs       /var
227-231    used    center         hd3             jfs       /tmp
232-232    used    center         hd1             jfs       /home
233-233    used    center         hd6             paging    N/A
234-234    used    center         hd4             jfs       /
235-309    used    center         hd2             jfs       /usr
310-392    used    inner middle   hd2             jfs       /usr
393-399    used    inner middle   hd6             paging    N/A
400-412    free    inner middle
413-433    used    inner edge     hd9var          jfs       /var
434-453    used    inner edge     hd3             jfs       /tmp
454-477    used    inner edge     hd1             jfs       /home
478-515    free    inner edge
```

The goal here is to de-fragment our filesystems so the when searching a filesystem for data, the physical read/write head of the drive will not have to move all over the disk platter. This reduces what's called the seek time. Another factor to consider is rotational delay. Rotational delay is a measure of how long it takes for a piece of data on the disk to "spin around" until it reaches the read/write head. The outside edge of the disk has the least delay, because it is spinning the fastest. The inside edge of the disk has the longest rotational delay. This becomes a factor mostly in sequential reads and writes. Random reads and writes, which will be the majority of the data on this system, will benefit more from reducing the seek time. The center of the disk is the best place to "hedge our bets" to reduce seek time. Since at any given moment the read/write head may

be anywhere on the disk, it would make sense that the middle of the disk is the "average" spot where the read/write head would be. The read/write head may be at either edge of the disk at any time, but the center of the disk is "halfway" in between the two. We want our most performance-critical data here, at the center of the disk[2].

To make this task simple, we will break down the usage of each filesystem to get an idea of the most dynamic filesystems and the most static filesystems.

Filesystem	Logical volume	Usage	Seek Time Critical?	Best placement
/home	hd1	Home directories for user's data.	No, moderately static	Inner edge
/usr	hd2	Most system binaries and programs.	Yes, static	Outer edge
/tmp	hd3	Temporary data.	Yes, dynamic	Outer-middle
/	hd4	Root filesystem, /etc and programs needed for boot up before /usr is mounted.	No, static	Outer Edge
/var	hd9var	/var fileystem: process locks, process PID files, logs, shared user data	Yes, dynamic	Inner-middle
-	hd5	Boot logical volume	Cannot be relocated	PP 1, Outer Edge
-	hd6	Default Paging Space	Yes, dynamic	Center

| - | hd8 | JFS filesystem log for rootvg | Yes, dynamic | Center |

As we can see, a few of our filesystems and logical volumes have similar requirements, with a handful favoring the center of the disk. In AIX there are 5 distinct placement areas for data, as you can see from the "lspv –p" output above: Outer Edge, Outer Middle, Center, Inner Middle and Inner Edge. Now we will place the filesystems and logical volumes in one of these five areas of the disk, using the following points:

1. The / (root) filesystem is not very seek-critical. Most files in root are system programs or configuration files (/etc) that are read during system initialization or when a program starts. From then on they reside in memory, unless the program is terminated. Since hd5 is the first spot accessed on disk during system IPL, it would make sense to place root near hd5, in the outer edge.
2. hd8, the JFS log, is touched every time there is a change to a filesystem in rootvg[3]. It would make sense to place hd8 near the filesystems that are the most dynamic to reduce seek time.
3. hd6, our paging space, is probably going to be the busiest logical volume we have. We'll prioritize hd6 to the center-most area on the disk
4. h5 is only read at boot up, so is not performance-critical at all. Also, it has to reside in physical partition 1 on the boot disk, so we can't move it anyway.
5. /usr is similar to the root filesystem. Program binaries for the operating system reside here, and are usually read once, when the program executes, and from there resides in memory unless the program is terminated. We can place this one out by the root filesystem on the outer edge.
6. /tmp and /var are similarly dynamic. We'll place these toward the center of the disk with hd6 and since these are near the center, hd8 will be as well.
7. /home is somewhat dynamic on most systems, especially if it is a development system where users are compiling programs, etc. It is not, however, system-critical for performance. Once developers have tested their code and it moves into "production" it should normally go somewhere in the /usr filesystem (i.e. /usr/local/ by convention). We can place /home on the inner-edge of the disk.

The following commands will set the "intra-policy allocation" flag for all of our filesystems and logical volumes. This flag tells AIX where our "first choice" is to place data on the disk. These commands will not actually relocate the data, we will do that in the next step.

```
# chlv -a ie hd1
# chlv -a e hd2
# chlv -a m hd3
# chlv -a e hd4
# chlv -a im hd9var
# chlv -a c hd6
# chlv -a c hd8
```

This will set the flags that the "reorgvg" command will use to manipulate the physical partitions on the disk. To reorganize each logical volume, run this command:

```
# reorgvg rootvg
```

When finished, we can see that all the filesystems are no longer fragmented, and they are placed in the best spot for performance:

```
# lspv -p hdisk0

hdisk0:
PP RANGE  STATE   REGION         LV NAME          TYPE     MOUNT
POINT
  1-2     used    outer edge     hd5              boot     N/A
  3-10    used    outer edge     hd4              jfs      /
 11-103   used    outer edge     hd2              jfs      /usr
104-206   used    outer middle   hd2              jfs      /usr
207-260   used    center         hd2              jfs      /usr
261-309   used    center         hd6              paging   N/A
310-388   used    inner middle   hd6              paging   N/A
389-389   used    inner middle   hd8              jfslog   N/A
390-412   used    inner middle   hd9var           jfs      /var
413-414   used    inner edge     hd9var           jfs      /var
415-439   used    inner edge     hd3              jfs      /tmp
440-464   used    inner edge     hd1              jfs      /home
465-515   free    inner edge
```

Mirroring the rootvg Volume Group

The next step will be to mirror our disk, hdisk0, so that a failure in the disk will not cause the system to crash. AIX has built in software RAID, ranging from RAID0 to RAID5. In a two disk configuration, RAID1 (Mirroring) is our best option. This is easily done by first adding the unused hdisk1 into the rootvg volume group:

```
# extendvg -f rootvg hdisk1
```

Then we tell AIX to mirror hdisk0 to hdisk1:

```
# mirrorvg -m rootvg hdisk1
```

The "-m" flag tells the mirrorvg command to make an exact map of the source disk on the destination disk. This way we can be assured that there is no fragmentation on the mirror disk either. On the E30, this takes about 15 minutes to complete. Now we must build a new boot logical volume on hdisk1:

```
# bostboot -ad /dev/hdisk1
```

And finally, change the default bootlist so that we may boot from either disk in case of a failure:

```
# bootlist -m normal hdisk0 hdisk1
```

Create the System Dump Logical Volume

The next step will be to create a logical volume for unexpected system dumps. To find out how big we should make this dump device, we use the sysdumpdev command:

```
# sysdumpdev -e
```

And the output:

```
0453-041 Estimated dump size in bytes: 46137344
```

This shows us that if the system were to dump, the dump image would take approximately 46MB of space. This estimate, of course, is made while the system is relatively idle. When we have a fully-configured system, this number could be much larger (i.e. when Sendmail is running). We will start with this estimate, and increase the size later after the system has been fully configured and Sendmail is installed.

The first step is to make a dump logical volume that will hold the data. The dump logical volume is created just like any other logical volume, except the "type" is going to be "sysdump" as we see in the following command:

```
# mklv -t sysdump -y dumplv -c1 -a ie rootvg 12 hdisk0
```

This will create a logical volume, called "dumplv" which will be 12 logical partitions in size, or 48MB. Notice that we specified only one copy of the logical volume with the "-c1" flag (no mirroring). This is because the dump device is not critical to normal system operation. If the system crashes and initiates a dump, it does not matter that the device is not mirrored because the operating system is effectively "down" at that point. Also, we specified via the "-a ie" parameter, that the logical volume is to be placed on the inner edge of the disk. Performance of this logical volume is not a factor here.

The final step is to assign dumplv to be the primary dump device for this system:

```
# sysdumpdev -P -p /dev/dumplv -d /var/adm/ras -C
```

This will assign the dumplv device to be our primary dump device, and will compress the image to maximize space. Upon reboot, the system will not force us to copy the dump to a file in /var/adm/ras as it would if the primary device was a paging space, like hd6. When the system is back online, we can examine the dump using the iadb command, as long as dumplv is big enough to hold the dump. We will not go into the use of the iadb command in this paper, though. The next step is to build this information into the kernel:

```
# bosboot -ad /dev/hdisk0
# bosboot -ad /dev/hdisk1
```

Then reboot the system:

```
# shutdown -Fr
```

Configuring the System Log Daemon (syslogd)

The syslog daemon is the primary program responsible for taking status and other messages from different software running on a system and putting those messages into a log. In its most basic form syslog takes input from messages that are logged to the /dev/log socket via a syslog API that is compiled into programs that wish to use the syslog service. When syslog receives a message on this socket, it consults its configuration file, /etc/syslog.conf, to determine what to do with the message. There are two main options: log it somewhere or ignore it. To ignore a message, simply leave the message's facility and level out of the syslog.conf. If syslog has been instructed to log the message, there are a number of different paths that it can take: log to a file, log to a user's terminal, log to all users' terminals or log to a remote syslog server.   For a given message, one or more of these events may be configured. For instance, you can have a particular warning message sent to all the users' terminals to notify them, and have the message saved in a file for future reference.

The version of syslog that comes with AIX 5L also has options of how to handle the rotation of log files that are under its control. For the purposes of this server, we are going to rotate the log files weekly, compress them to save space, and keep 8 versions which is roughly 2 months. We are also logging most of the important messages to a central syslog server. The configuration of this server will not be covered here, though. If your site does not have a central syslog server omit the "loghost" line from the sample configuration below.

Edit the /etc/syslog.conf file and add the following lines. Delete any other lines that are not comments[4]:

```
local4.debug        /var/adm/ipsec rotate time 1w files 8 compress
auth.info           /var/adm/secure rotate time 1w files 8 compress
mail.info           /var/adm/maillog rotate time 1w files 8 compress
daemon.info         /var/adm/syslog rotate time 1w files 8 compress
*.info              @loghost
```

Make sure there is a tab between the first and second columns.   The @loghost entry means that messages from any facility that are level "info" or higher will be logged to the remote server called loghost. This is just a hostname of a server, not a special keyword. In this example, we are going to add an entry into the local /etc/hosts file which tells this server what IP address belongs to loghost.

```
# vi /etc/hosts
```

Add this line, substituting <*ipaddress*> with the IP address of your central syslog server:

```
<ipaddress> loghost
```

Next make sure all the local files referenced in the syslog.conf exist and have the appropriate permissions. We don't want everyone to be able to read the logs, but we do want systems administrators to be able to read them without having to "su" to the root account. Therefore, these commands will create the files appropriately:

```
# cd /var/adm
# touch ipsec secure maillog syslog
# chmod 740 ipsec secure maillog syslog
# chown root:sysadm ipsec secure maillog syslog
```

The next step will be to change the startup of the syslog daemon so that it will not accept syslog messages from other remote hosts. If left in the default state, an attacker who is able to connect to UDP port 514 could flood our syslog server with fake messages. Not only could this confuse the systems administrator, by possibly inserting messages with false information, etc, an attacker could potentially cause a Denial of Service attack either by flooding the syslog daemon with more messages than it can handle, or by simply filling up the filesystems on which the logs reside (/var in our case).

---

[4] For a detailed discussion of the syslog daemon and the configuration options, see AIX 5L Version 5.1 Commands Reference, Volume 5: syslogd Daemon
http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/cmds/aixcmds5/syslogd.htm

AIX controls syslogd with the System Resource Controller, which is similar in function to the inetd super server. The startup options for each subsystem, like syslogd, are kept in AIX's Object Data Manager (ODM). The ODM is a database which houses information, mainly on system configuration. The ODM is used for some configuration options instead of test files, like /etc/syslog.conf for instance. To change the syslogd subsystem so that syslogd is run with a "-r" parameter we use the "chssys" command:

```
# chssys -s syslogd -a "-r"
```

Finally, restart the syslog daemon for the changes to take effect:

```
# refresh -s syslogd
```

If you would like to verify that the "-r" parameter was passed successfully to the syslogd subsystem, run this command:

```
# ps auxw | grep syslog
```

And the output:

```
root 6978  0.0  1.0  588  600 - A    22:35:34  0:00 /usr/sbin/syslogd -r
```

Tying the Error Report into Syslog

AIX's built in error reporting daemon, errdaemon, has the ability to log many different types of events from hardware and software failures to systems administrator informational messages. The syslog daemon can also log its messages to the errdaemon by specifiying "errorlog" instead of a filename in the /etc/syslog.conf[4]. Conversely, the errdaemon can be coaxed into sending its message to the syslog daemon. We will add the information on how to do this into the ODM with a template developed by Andreas Seigert in his book "The AIX Survival Guide" which is cited in the References section.

First, create a temporary file /tmp/errlog. Add the following lines to the file:

```
errnotify:
    en_pid = 0
    en_name = "syslog"
    en_persistenceflg = 1
    en_label = ""
    en_crcid = 0
    en_class =""
    en_type = ""
    en_alertflg = ""
    en_resource = ""
    en_rtype = ""
    en_rclass = ""
```

```
        en_method = "/usr/bin/errpt -l $1 | /usr/bin/tail -1 |
/usr/bin/logger -t errpt -p daemon.notice"
```

Save this file and incorporate it into the ODM with this command:

```
# odmadd /tmp/errlog
```

To test this, use the errlogger command:

```
# errlogger "test message"
```

You should see a message in the /var/adm/syslog file, that says there was an
operator message logged, and also in the syslog file of your central syslog
server. To view the actual message, use the errpt command:

```
# errpt

AA8AB241    0620213302 T O OPERATOR       OPERATOR NOTIFICATION
```

Then, use another form of the errpt command to view the details:

```
# errpt -aj AA8AB241

----------------------------------------------------------------
LABEL:          OPMSG
IDENTIFIER:     AA8AB241

Date/Time:      Thu Jun 20 21:33:30 EDT
Sequence Number: 49
Machine Id:     00200003C000
Node Id:        availaix
Class:          O
Type:           TEMP
Resource Name:  OPERATOR

Description
OPERATOR NOTIFICATION

User Causes
ERRLOGGER COMMAND

        Recommended Actions
        REVIEW DETAILED DATA

Detail Data
MESSAGE FROM ERRLOGGER COMMAND
test message
```

Removing Default User and Group Accounts

Now that we have completed the installation tasks, we can begin securing the AIX operating system. The first step will be to remove unused/unnecessary user and group accounts to limit the server's exposure to attempts to access it via "back door" or "default" accounts. First we will remove these unused user accounts:

daemon	**guest**	**lpd**
lp	**nuucp**	**imnadm**

Use the rmuser command to remove the user accounts, substituting the user id in the table above with the *userid* parameter in the command:

```
# rmuser -p userid
```

Next we can also remove some of the unused AIX groups:

ecs	**usr**	**perf**
shutdown	**lp**	**imnadm**

Before removing the shutdown group, execute the following command:

```
# find -group shutdown -print
```

You should see the following output:

```
/usr/sbin/exec_shutdown
/usr/sbin/reboot
/usr/sbin/fastboot
/usr/sbin/shutdown
```

If this is the case, run this command to change the group ownership of these files to the system group:

```
# find / -group shutdown -exec chgrp system {} \;
```

Now proceed with removing the groups from the list above. This time, use the rmgroup command, substituting the group names above for the *groupid* parameter:

```
# rmgroup groupid
```

The rmgroup and rmuser commands may leave some files without an owner, like the imnadm files in /var/docsearch. To find these files, run the following commands:

```
# find / -nouser

# find / -nogroup
```

Any files with "docsearch" in the path can be safely removed, as well as the /home/guest directory (and all files, if any) and the /var/spool/uucppublic/.profile file. If there are other files in your output, please be sure to verify that those files can be deleted before proceeding with the next set of commands. If you have followed all the steps up until now, the list above should be the only files you will come across. Run the following commands to delete these files:

```
# find / -nouser -exec rm -fr {} \;

# find / -nogroup -exec rm -fr {}\;
```

Securing Remaining User and Group Accounts

Now that we have removed some of the unused accounts to limit the server's exposure, we need to set some restrictions and options for the remaining user accounts. Even though this server will not be a server that users routinely log into, except administrators, it is good practice to set up these restrictions in case the role of this server changes in some way, and to protect the system from weakly-defined administrator accounts.

The first step will be to set parameters on account passwords. AIX has a number of options that define how passwords are implemented. The settings we are concerned with are:

Parameter Name	Definition	Recommended Setting[5]
dictionlist	Comma-separated list of filenames of dictionary files to check a user's password against.	/usr/share/dict/words
histexpire	How many weeks after a password has been used can it be reused.	26
histsize	Defines the number of previous passwords that cannot be reused.	8

---
[5] Some recommendations taken from Additional AIX Security Tools on IBM eServer pSeries, IBM RS/600 and SP Cluster in the References section.

loginretries	How many invalid login attempts until the user account is locked.	3
maxage	Number of weeks until a password expires and must be changed.	6
maxexpired	Number of weeks after a password expires that the user is still allowed to change it.	2
maxrepeats	How many times a given character can appear in a password.	2
minage	Minimum number of weeks between password changes.	0
minalpha	Minimum number of alphabetic characters that can appear in a password.	4
mindiff	The minimum number of characters in the new password that were not in the old password.	3
minlen	Minimum password length.	8
minother	Minimum number of non-alphabetic characters that can appear in a password.	1

pwdwarntime	Number of days before a user's password expires that they will be warned of the expiration.	7

The dictionlist parameter takes a comma-separated list of files that contain listings of dictionary words. If the user's choice of passwords matches any word in a dictionary file, the password is rejected as a valid choice. The format of the dictionary files is one word per line with no leading or trailing white space (tabs, spaces) and all words should be 7-bit ASCII characters. There are numerous resources on the Internet for obtaining dictionary files that match these requirements. For now, create an empty file called /usr/share/dict/words. When the system is connected to the network, obtain "real" dictionary files from trusted sources and place them in the /usr/share/dict directory and add them to the dictionlist parameter.

```
# mkdir –p /usr/share/dict
# > /usr/share/dict/words
# chmod 0644 /usr/share/dict/words
```

In order to implement the password restrictions above, edit the /etc/security/user file, after making a backup copy:

```
# cp /etc/security/user /etc/security/user.aix51
# vi /etc/security/user
```

Find the "default:" stanza after the comments at the beginning of the file. Find the lines with each of the parameters above and change them to the suggested values above. If a parameter does not appear at all, just add a new line and insert it. Take care not to add any blank lines between each parameter.

Find the stanza for "root:" and make sure that the loginretries parameter for root is 0, meaning that it does not apply to root. This will prevent the root account from being locked out of the system by accident or malicious intent.

Change all of the default system accounts that are left so that each of the accounts is locked down from interactively logging into the system. This is an added measure to patch as many ways into the system via default accounts as possible. First, list all the accounts that are still on the system:

```
# cat /etc/passwd | cut –f1 –d:
```

For each account, except root and any systems administrator accounts you have created, run the following commands:

```
# chuser account_locked=true <username>
# chuser login=false <username>
# chuser rlogin=false <username>
```

Lastly, we will modify the /etc/profile to include an idle timeout that will close login sessions after a certain time with no activity. This is a way to ensure that, for instance, a systems administrator does not leave herself logged in after going home for the night. If someone were to find her open login session on her workstation they could perform actions with her account. Since each shell uses a different environment variable for this, we'll set both in the /etc/profile. Make a backup copy of the /etc/profile and add the following lines. The TMOUT variable will already be present in the file: just uncomment it and change the value and add TMOUT and TIMEOUT to the last "export" line:

```
TMOUT=600
TIMEOUT=600
export TMOUT TIMEOUT
```

Modifying and Securing the root Account

The next step will be to secure the root account, to make it more difficult for someone to gain direct access to root. We want anyone who wants to access the root account to have to log in as another user first. No one will have direct access to log in as root, even if they know the password, via the network:

```
# chuser rlogin=false root
```

This will allow only local console logins for root, or access via the "su" command from another account. We also want only certain users to be able to perform the su command to the root account, like systems administrators. By doing this we add a layer of protection such that if someone knows the root password, they must also have either physical access to the server (to log in on the console directly) or know the account name <u>and</u> password of a systems administrator, in addition to the root password, in order to gain root access.

The first step is to create a group for systems administrators. This group will be called "sysadm" and all new users who are also systems administrators will be added to this group.

```
# mkgroup id=210 sysadm
```

This creates the sysadm group with the gid of 210, which should be available on a default AIX system like the one we are installing. If you need to choose another gid, feel free to do so. Now we will set permissions so that only members of the sysadm group can "su" to the root account:

```
# chuser sugroups=sysadm root
```

When you add new administrator accounts to the system, remember to make them members of the sysadm group.

Next, we will set the default umask of the root account so that newly-created files are not automatically world-readable. This ensures that if an administrator using the root account forgets to explicitly set permissions on a new file that only the root account will be able to access it. Run this command to set the default umask for root:

```
# chuser umask=077 root
```

Security-related Files

The next step will be to modify or create certain security-related files in AIX. Some of these files directly affect the security of certain network daemons, while others indirectly affect security by imposing limits on certain AIX operating parameters.

The first file we will create is the /etc/ftpusers file. This file is a list of user names that are not allowed to log into the server via FTP. By default, all users, especially the root account and other system accounts, should be listed. If you have a need for specific users to be able to log in via FTP, you should grant them access explicitly by removing them from the /etc/ftpusers file. For the purposes of this paper, we are not using FTP at all, so all users will be listed in the /etc/ftpusers file. The reason for listing all the users even though we will not be running the FTP daemon is that if an administrator accidentally "turns on" the FTP daemon, we are still a little safer knowing that none of the accounts have access to FTP to be able to retrieve any files.

Run the following command to add all the current AIX users to the /etc/ftpusers file and set appropriate permissions:

```
# cat /etc/passwd | cut -f1 -d: > /etc/ftpusers
# chmod 600 /etc/ftpusers
```

We also do not want rogue cron or at jobs to be run without our knowledge. On this system the only user that will need to submit jobs via cron and at is the root account. To disable all other accounts from submitting cron or at jobs, add the user accounts to two files: cron.deny and at.deny.

```
# cat /etc/passwd | cut -f1 -d: | grep -v ^root \
> /var/adm/cron/cron.deny
# cp /var/adm/cron/cron.deny /var/adm/cron/at.deny
# chmod 000 /var/adm/cron/*deny
```

The next file we will modify is the /etc/security/limits file. This file sets all the default, per-user restrictions on items such as maximum file sizes, core file sizes and CPU utilization. We are going to restrict the creation of core files so that no users are allowed to create them. This decreases the risk of a misconfiguration, or a malicious attacker, from creating a Denial of Service attack by filling up key filesystems with useless core files. Also, core files can be analyzed with simple commands such as the strings command. An analyzed core file may leak private information such as user accounts or passwords on the system. To stop the ability for programs to create core files, edit the /etc/security/limits file and change the line with "core =" to:

```
core = 0
```

Also, add a new line below it that reads:

```
core_hard = 0
```

This will set the hard and soft limits for core files to zero blocks (blocks are 512 bytes).

Next, create default login banners that will be displayed when someone connects to the machine via a remote session or the console. A sample warning banner would be:

```
********************************  WARNING  ********************************
*
*  This system is restricted solely to Acme Company authorized personnel
*  for legitimate business purposes only.  The actual or attempted unauthorized
*  access, use or modification of this system is strictly prohibited.
*  Unauthorized personnel are subject to disciplinary proceedings and/or
*  criminal and civil penalties under state, federal or other applicable
*  domestic and foreign laws.  The use of this system is monitored and recorded
*  for administrative and security reasons.  Anyone accessing this system
*  expressly consents to such monitoring and is advised that if such monitoring
*  reveals possible evidence of criminal activity, Acme Company may
*  provide the evidence of such activity to law enforcement officials.
*
*  Acme Company, LLC
*  security@acme.com
*
*
**************************************************************************

Login:
```

The file that is responsible for this pre-login banner is /etc/security/login.cfg. This file, like most configuration files in AIX, is made of different stanzas with a set of statements for each stanza. Look for the default stanza, which is directly after the first comment block and starts with "default:".

Add "herald" and "herald2" statements to the default stanza. If we use the example message above, it would look like this:

```
default:
        herald =
"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
\n\n\n\n\n********************************  WARNING
**********************************\n*\n*\n*  This system is restricted
solely to Acme Company authorized personnel\n*  for legitimate business
purposes only.  The actual or attempted unauthorized\n*  access, use or
modification of this system is strictly prohibited.\n*  Unauthorized
personnel are subject to disciplinary proceedings and/or\n*  criminal
and civil penalties under state, federal or other applicable\n*
domestic and foreign laws.  The use of this system is monitored and
recorded\n*  for administrative and security reasons.  Anyone accessing
this system\n*  expressly consents to such monitoring and is advised
that if such monitoring\n*  reveals possible evidence of criminal
activity, Acme Company may\n*  provide the evidence of such activity to
law enforcement officials.\n*\n*  Acme Company, LLC\n*
security@acme.com\n*\n*\n*********************************************
*********************************\n\nLogin: "
herald2 =
"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n
\n\n\n\n\n********************************  WARNING
**********************************\n*\n*\n*  This system is restricted
solely to Acme Company authorized personnel\n*  for legitimate business
purposes only.  The actual or attempted unauthorized\n*  access, use or
modification of this system is strictly prohibited.\n*  Unauthorized
personnel are subject to disciplinary proceedings and/or\n*  criminal
and civil penalties under state, federal or other applicable\n*
domestic and foreign laws.  The use of this system is monitored and
recorded\n*  for administrative and security reasons.  Anyone accessing
this system\n*  expressly consents to such monitoring and is advised
that if such monitoring\n*  reveals possible evidence of criminal
activity, Acme Company may\n*  provide the evidence of such activity to
law enforcement officials.\n*\n*  Acme Company, LLC\n*
security@acme.com\n*\n*\n*********************************************
*********************************\n\nYou Entered an invalid login or
password.\n\nLogin: "
```

The "herald" parameter is what is displayed when a user makes an initial connection to a telnet port on this server, or attempts to log in at the console. The "herald2" statement is what is displayed if the user enters an incorrect user ID or password in response to the Login: prompt. Be sure there are no blank lines between the herald and herald2 statements. The text in herald and herald2 can be any text enclosed in quotations. To force line breaks or tabs, use the "printf" structure (i.e. \n for carriage returns, \t for tabs). The purpose of adding 40 carriage returns to the beginning of the output is twofold: it "cleans up" the screen by flushing it before displaying the banner so it stands out more, and many "banner grabbing" programs usually only grab the first few lines of output.

In this case, they would get nothing but blank lines. Do not exit editing this file yet.

Now we will change some settings to make it harder for "brute force" password crackers to compromise this server. A number of different statements in the default: stanza affect this:

<table>
<tr><td>logindelay</td><td>How many seconds to delay the display of the Login: prompt between unsuccessful login attempts. This delay is multiplied by the number of unsuccessful logins. For instance, if this is the 2nd attempt, the delay is multiplied by 2, if this is the 4th attempt, the delay is multiplied by 4.</td></tr>
<tr><td>logindisable</td><td>The number of unsuccessful login attempts before this port is locked. When locked, it can only be unlocked by an administrator. Take care with this option because it could lead to a denial of service attack which could render no one able to legitimately log into this server.</td></tr>
<tr><td>logininterval</td><td>The number of seconds during which logindisable unsuccessful login attempts must occur before the port is locked.</td></tr>
<tr><td>loginreenable</td><td>The number of minutes that will elapse before the port is automatically re-enabled. If this is set to zero, it will need to be unlocked by an administrator.</td></tr>
</table>

For our purposes, we'll set the following:

logindelay	2
logindisable	3
logininterval	14
loginreenable	5

This means that each time an unsuccessful login is received on a given port, a multiple of 2 seconds will elapse before the next Login: prompt is displayed. If 3 unsuccessful login attempts are received within 14 seconds, the port is locked for 5 minutes until it is automatically re-enabled. Three attempts in 14 seconds is most likely not a legitimate user, but some type of brute force password generator trying to "guess" passwords for a user. How do we know this?

Because of the settings for logindelay, a user who is on their third attempt at logging in will have waited 12 seconds total for AIX to redisplay the Login: prompt (2 seconds for the first, 4 for the second and 6 for the third). With the average human delaying at least a second or two between seeing the new Login: prompt and actually typing it in, versus a computer program who's response to the Login: prompt is instantaneous, we can see that if a password is typed wrong 3 times within a 14 second period, it is most likely not a human doing the password entry. These settings are a compromise between irritating or inconveniencing legitimate users and thwarting "casual" hackers looking for easy prey. You could, obviously, be more strict or less strict depending on your needs.

In the /etc/security/login.cfg there are separate stanzas for each port you can login in to. The default stanza gives the values for all ports unless those statements are redefined for a particular port. You could, if you wanted, have different banners appear to local (console) users versus remote users. For more information, see reference [5].

Lastly, while still editing the /etc/security/login.cfg file, we will remove some shells from the usw: stanza in the shells statement. We are not using any dial-up connections on this server, so there is no need for the /usr/sbin/sliplogin to be listed as a valid shell. You may also wish to remove other shells if you want to limit the types of login shells that a user may invoke. This will not stop someone from logging in via a Korn Shell, for instance, and switching to a Bourne Shell. It will only stop someone from using a non-listed shell as their login shell. Remove the /usr/sbin/sliplogin, /bin/ksh93 and /usr/bin/ksh93 from the list. Save the changes to this file.

One more spot for banners is the famous /etc/motd, or Message of the Day. The contents of this file are displayed on the user's terminal after successfully logging in either via a remote session, like telnet, or a console. The default message in this file gives information on the AIX version level as well as some helpful hints on what file to look at for pertinent information on AIX 5.1. This file should either reiterate the security policy for a site, or provide some helpful information to users, or just be left blank. For the purposes of this server, we'll echo the same information as we did in the herald banner, minus the 40 blank lines and the Login: prompt. This will ensure that any user accessing the system sees the information at least once, and cannot claim "ignorance" of the fact that this is a private system and that it is monitored. Edit the /etc/motd file and place the contents, line by line, that you want to display. There is no need to use the printf format as this file is just basically cat'd out as a standard text file.

```
# vi /etc/motd
# chmod 444 /etc/motd
```

Network Security

The next phase will be to secure the network aspects of this server. So far, all of our administration has been at the console, and the server is still not plugged into the network (is it?). Before we connect this server to the network and open it up for service, we need to make sure we have the least amount of exposure possible. Since this is going to be a Sendmail server, all we really need is a Sendmail daemon listening on TCP port 25, a way to synchronize the system clock (ntp) and some way to administer this server over the network, like telnet or Secure Shell. First, let's turn off all the unneeded services that run under the control of the inetd super-server. We don't need any of these services, really, so running the following commands will comment-out all the lines in the /etc/inetd.conf configuration:

```
# cp /etc/inetd.conf /etc/inetd.conf.aix51
# cat /etc/inetd.conf | sed /^/##/ > /tmp/inetd.conf
# mv /tmp/inetd.conf /etc/inetd.conf
# chmod 000 /etc/inetd.conf
```

In fact, since we are not running any services out of /etc/inetd.conf, we can turn off inetd completely. It is still a good idea to comment-out all of the services, though, so that in case inetd is started by accident the server will not be exposed.

```
# stopsrc -s inetd
```

You may notice that the telnet daemon is controlled by inetd. Not to worry, we are going to install and configure Secure Shell (SSH) to be used in place of telnet for security reasons[6]. That will be done later.

Next, we need to get rid of unneeded entries in the /etc/inittab file. The inittab is run when the system is booting, at different run levels. Some network-related services are started here, so we'll remove those entries:

```
# cp /etc/inittab /etc/inittab.aix51
# rmitab qdaemon
# rmitab writesrv
```

Then we'll manually stop these services:

```
# stopsrc -s qdaemon
# stopsrc -s writesrv
```

The qdaemon is the backend queuing service for remote print jobs. We won't be servicing any remote printing, so we can safely remove this service. The writesrv

---

[6] For more information on Secure Shell, see http://www.openssh.org

daemon allows users to send and receive messages from users on remote
systems and also from the printing subsystem. Again, this service is not needed
because not only are we not servicing print requests but no users, other than the
administrator(s), will be logging into this server.

The next step will be to edit the /etc/rc.tcpip file to remove even more
unnecessary network services. Edit the /etc/rc.tcpip file, after making a backup,
and comment-out the following lines. Some may already be commented-out, but
for consistency they are all listed:

```
start /usr/sbin/dhcpcd "$src_running"
start /usr/sbin/autoconf6 ""
start /usr/sbin/ndpd-host "$src_running"
start /usr/sbin/ndpd-router "$src_running"
start /usr/sbin/lpd "$src_running"
start /usr/sbin/routed "$src_running" -q
start /usr/sbin/gated "$src_running"
start /usr/lib/sendmail "$src_running" "-bd -q${qpi}"
start /usr/sbin/portmap "$src_running"
start /usr/sbin/inetd "$src_running"
start /usr/sbin/named "$src_running"
start /usr/sbin/timed "$src_running"
start /usr/sbin/rwhod "$src_running"
start /usr/sbin/snmpd "$src_running"
start /usr/sbin/dhcpsd "$src_running"
start /usr/sbin/dhcprd "$src_running"
start /usr/sbin/dpid2 "$src_running"
start /usr/sbin/hostmibd "$src_running"
start /usr/sbin/mrouted "$src_running"
start /usr/sbin/pxed "$src_running"
start /usr/sbin/binld "$src_running"
```

Make sure the following lines are not commented out:

```
start /usr/sbin/syslogd "$src_running"
start /usr/sbin/xntpd "$src_running"
```

Save the changes to this file and run the following commands to stop any
services that are running that we commented out:

```
# lssrc -a | grep active
```

Make note of the name of each service in the first column. For each service that
we commented out in the /etc/rc.tcpip file, run the following command. If there
are any services listed which you are not familiar with, just ignore them for now:

```
# stopsrc -s  servicename
```

If the syslogd service does not show as active, run this command to activate it:

```
# startsrc –s syslogd
```

The next step will be to change some of the running kernel parameters, both for security and administration. AIX uses the no command (Network Options) to modify certain kernel parameters that deal directly with the network. To view a list of all the options that are settable via the no command, use this:

```
# no –a | more
```

There will be quite a bit of output, so it is not included here. The parameters we are going to be modifying are:

Parameter Name	Description	Recommended Setting
bcastping	Respond to broadcast ICMP ECHO packets?	0
clean_partial_conns	Avoid SYN attacks by cleaning partial connections to make room for new connections.	1
extendednetstats	Report extensive statistics for network memory services?	1
icmpaddressmask	Respond to ICMP address mask requests?	0
ipforwarding	Forward IP packets not destined for this system?	0
ipsrcrouteforward	Forward source-routed packets not destined for this system?	0
ipsrcrouterecv	Accept source-routed packets destined for this system?	0

ipsrcroutesend	Allow applications to send source-routed packets?	0
nonlocsrcroute	Allow strictly source-routed packets to be addressed to outside hosts?	0
tcp_pmtu_discover	Allow Path MTU discovery for TCP?	0
udp_pmtu_discover	Allow Path MTU discovery for UDP?	0

The no command also sets the values for these parameters. Unfortunately, the parameters only affect the running kernel and the settings are lost when the system is rebooted. We need to create a script that will run when the system reboots to change these back to the values listed above. First, let's create a script called /etc/rc.local. In other Unix variants, like Linux, the rc.local file already exists. Its purpose is to add "local" commands (i.e. specific to a particular host) that are run during the boot phase by the init daemon. AIX does not ship with an rc.local, so we will create it and then add it to the init daemon's configuration file, /etc/inittab.

First, create the /etc/rc.local file with the no commands to set the values listed above:

```
# vi /etc/rc.local
```

Add the following lines:

```
#!/bin/ksh
#
#  rc.local
#
#  acme.com's customized rc.local file
#
##################################################
#
#  Change run-time kernel networking parameters
#
##################################################
/usr/sbin/no -o bcastping=0
/usr/sbin/no -o clean_partial_conns=10
```

```
/usr/sbin/no -o extendednetstats=1
/usr/sbin/no -o icmpaddressmask=0
/usr/sbin/no -o ipforwarding=0
/usr/sbin/no -o ipsrcrouteforward=0
/usr/sbin/no -o ipsrcrouterecv=0
/usr/sbin/no -o ipsrcroutesend=0
/usr/sbin/no -o nonlocsrcroute=0
/usr/sbin/no -o tcp_pmtu_discover=0
/usr/sbin/no -o udp_pmtu_discover=0
exit 0
```

Save the file and change the permissions so that it is executable:

```
# chmod 700 /etc/rc.local
```

Then, run the script to make sure it works and that it returns with an exit code of zero:

```
# /etc/rc.local
# echo $?
0
```

If the script worked, now it's time to add it to the inittab so it runs on system boot. We want this to be the last rc script that is run. Do this with the following command:

```
# mkitab 'rclocal:2:wait:/etc/rc.local >/dev/console 2>&1'
```

Make sure to enclose the "rclocal:2:wait:/etc/rc.local >/dev/console 2>&1" in <u>single</u> quotes.

Configuring the Network

Okay, at this point we need to configure the network in order to get our software from the Internet: Secure Shell (and supporting software) and Sendmail. At this point the only services that should be running are the NTP daemon (xntpd) and the syslog daemon (syslogd). We will assume, for the purposes of this paper, that whatever body or organization is responsible for assigning TCP/IP addresses on your network has already given you the following information for this system:

    Hostname
    IP Address
    Subnet mask
    Nameserver IP address(es)
    Domain name
    Default Gateway IP Address

At this time, go ahead and plug the network adapter cable into the network. With the information above we can quickly configure TCP/IP via smit:

```
# smitty mktcpip
```

The first menu will ask which interface you want to configure. This system is running Ethernet, so we will choose the en0 device using the arrow keys, and press ENTER. You will be presented with the following menu:

```
                         Minimum Configuration & Startup

         To Delete existing configuration data, please use Further Configuration menus

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* HOSTNAME                                           []
* Internet ADDRESS (dotted decimal)                  []
  Network MASK (dotted decimal)                      []
* Network INTERFACE                                   en0
  NAMESERVER
          Internet ADDRESS (dotted decimal)          []
          DOMAIN Name                                []
  Default Gateway
      Address (dotted decimal or symbolic name)      []
      Cost                                           [0]                              #
      Do Active Dead Gateway Detection?               no                              +
  Your CABLE Type                                     N/A                             +
  START Now                                           no                              +



F1=Help                 F2=Refresh              F3=Cancel               F4=List
Esc+5=Reset             F6=Command              F7=Edit                 F8=Image
F9=Shell               F10=Exit                Enter=Do
```

Fill in the appropriate information, and leave the options under "Default Gateway" "Cost" and "Do Active Dead Gateway Detection" at the defaults of "0" and "no" respectively.

For the HOSTNAME field, put in just the short hostname, like "mail" if your server's hostname is mail.acme.com. Put the acme.com part in the DOMAIN Name field.

Leave the "START Now" field at "no." This tells AIX that you want to start all the TCP/IP daemons listed in the /etc/rc.tcpip. We commented-out most of them, and the others are already running, so we don't need to re-run the /etc/rc.tcpip script.

Press ENTER to confirm these settings and bring up the adapter.

Test your setup by first pinging the default gateway's IP address:

```
# ping <ip_of_gateway>
```

If the output looks similar to the output below, press CTRL-C to stop the ping and proceed to the next step. If not, check all the settings above and make sure the adapter is connected properly to the network.

```
PING 192.168.10.1: (192.168.10.1): 56 data bytes
64 bytes from 192.168.10.1: icmp_seq=0 ttl=255 time=1 ms
64 bytes from 192.168.10.1: icmp_seq=1 ttl=255 time=1 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=255 time=1 ms
```

Next, make sure that the Domain Name Server (DNS) can be reached and that hostnames can be resolved from this system. Try to resolve a hostname that you know you can resolve from other hosts on your network, like www.ibm.com:

```
# host www.ibm.com
www.ibm.com is 129.42.18.99
```

If this is not successful, work with your networking group to track down the source of the problem. When you are able to properly resolve hostnames, proceed to the next step.

Now that the network is running, we need to make sure we can synchronize our date and time with the NTP server that is configured in the /etc/ntp.conf from previous steps. First, stop the xntpd daemon by issuing this command:

```
# stopsrc -s xntpd
```

Then, query the time on the NTP server from the /etc/ntp.conf file:

```
# ntpdate <ipaddressofNTPserver>
```

If you get a message similar to the one below then you can synchronize to this server. If you get an error message, you may have to use another NTP server: some servers require pre-arranged agreements between you and the NTP server owner and some do not service public NTP clients at all. You should see a message similar to this:

```
10 Jun 23:07:15 ntpdate[6772]: adjust time server 0.0.0.0 offset 0.029086
sec
```

If the ntpdate command was successful, then run this command to set the time to that of the NTP server:

```
# ntpdate -s <ipaddressofNTPserver>
```

Then, check the current date and time to make sure it is correct:

```
# date
```

If so, restart the xntpd daemon so that the time will automatically be synchronized periodically:

```
# startsrc -s xntpd
```

Getting Secure Shell and Sendmail

First, create a directory on the AIX system where we can house programs for installation. Create a /usr/local/src directory using the following commands:

```
# mkdir -p /usr/local/src
# chmod 0755 /usr/local/src
```

From a workstation with Internet access, connect via a web browser to http://www.bull.de/pub. Bull is one of the largest and most trusted websites for pre-compiled AIX shareware and freeware. Since we are not installing a compiler on this system, we need to get software that is ready to be installed on AIX 5.1 from a site such as this. Follow the links to download software for AIX 5.1. Get the following software[7]:

> OpenSSH-3.0.2.1
> OpenSSL-0.9.6.3
> zlib-1.1.4.0
> Sendmail-8.11.6

Place these files on a server or workstation running an FTP daemon that can be reached from the newly-installed AIX server. Specifics on how to do this are not within the scope of this document. Once the software is in an accessible place via FTP, go back the console of the AIX system. FTP the software from the FTP server to the /usr/local/src directory of the AIX system. When finished, "cd" into the /usr/local/src directory and run the following commands to extract the installable images from the .exe files that were downloaded:

```
# chmod 700 *exe
# ./openssh-3.0.2.1.exe
# ./openssl-0.9.6.3.exe
# ./zlib-1.1.4.0.exe
```

---

[7] The software listed is the most current as of the time this paper was written. More recent software is uploaded to Bull each day, so newer versions may exist. Before downloading a newer version, research the interaction it may have with other software in this list, especially OpenSSH, OpenSSL and zlib.

```
# ./sendmail-8.11.6.exe
```

To save room, you may delete the .exe files:

```
# rm -i *exe
```

You are now left with a .bff and .asc file for each software package. Next, verify that the integrity of each .bff (Backup File Format) files is intact. View the .asc file that came with each .bff file. Look for the section that says "The sum(1) checksum for <*filename*> is:". Write down the two numbers on the next line.

Run the following command to print out the checksum of each .bff file. Compare these to the numbers you wrote down from the .asc files and make sure they match before proceeding. If the numbers are different it could mean that the .bff file was tampered with, or that the FTP download somehow corrupted the file. Try FTP'ing the file from the Bull website again. If the file checksums do not match and you suspect that the file was tampered with, contact Bull immediately. Instructions on how to do this appear on their website.

```
# sum *bff
```

Note that the fact that the checksums match is still no guarantee that the files were not tampered with. It is still possible to modify the files in such as way that the checksum is still the same. This step is simply a quick check and is easily done for a little more peace of mind.

Lastly, run the following command to build a Table of Contents (.toc) file in the /usr/local/src directory so that the installp program knows which installable filesets reside in this directory:

```
# inutoc /usr/local/src
```

Installing and Configuring OpenSSH

To install OpenSSH, you must also install two pre-requisite software packages: zlib, which is a compression library used in OpenSSH; and OpenSSL, which is the cryptography package that provides the libraries for OpenSSH's encryption and decryption mechanisms. Using the following command, installp will install all three filesets:

```
# installp -acqgQNX -d /usr/local/src freeware.openssh
```

Make sure that the Installation Summary shows SUCCESS next to each APPLY operation for all three filesets. The OpenSSL software does need any post-installation configuration, unless you are going to use the software to generate

certificates, like for a web server. In our case, we will not use that functionality.
Also, the zlib libraries do not need any post-installation configuration.

We do, however, need to configure the OpenSSH software. For this version of
OpenSSH the configuration files and keys exist in the /etc/openssh directory.
The installation should have added an /etc/rc.openssh script for starting and
stopping the OpenSSH daemon. Also, an entry was added to the /etc/inittab to
automatically start the daemon in run-level 2 (the default AIX runlevel).

To stop the currently running SSH daemon (sshd), which was started
automatically after the installation, run this command:

```
# /etc/rc.openssh stop
```

Now, edit the /etc/openssh/sshd_config file to setup operating values for the
OpenSSH daemon. First, make a backup copy:

```
# cp /etc/openssh/sshd_config /etc/openssh/sshd_config.aix51
# vi /etc/openssh/sshd_config
```

Change the following entries, or add them if they do not exist:

Parameter	Description	Recommended Value
AllowGroups	Members of these group(s) are allowed to log in.	sysadm
Banner	Output the contents of this file as a pre-login banner.	/etc/motd
ChallengeResponseAuthentication	Is it allowed or not?	no
ClientAliveInterval	Time to send Client Alive messages to the client.	30
ClientAliveCountMax	Number of non-received responses to Client Alive messages before disconnecting the client.	2

GatewayPorts	Can other clients connect to forwarded ports for a client?	no
HostbasedAuthentication	Use ~/.rhost and /etc/hosts.equiv with public key encryption?	no
IgnoreRhosts	Do not use .rhosts in authentication mechanisms that can use it.	yes
LoginGraceTime	Timeout for user authentication.	60
LogLevel	The syslog level at which sshd will log.	INFO
PasswordAuthentication	Use the built-in /etc/passwd file?	yes
PermitEmptyPasswords	Do we?	no
PermitRootLogin	Can root login directly via SSH?	no
PrintLastLog	Print the lastlog entry when a user logs in?	yes
Protocol	What SSH protocol versions can we use?	2
RhostsAuthentication	Can we use it?	no
StrictModes	Check permissions on users' home directories and key files?	yes

SyslogFacility	What syslog facility to use when logging?	AUTH
UseLogin	Use the built-in AIX login facility?	no
X11Forwarding	Do we forward X11 displays through the encrypted channel?	no

When all the options have been set according to the table above, restart the SSH daemon:

```
# /etc/rc.openssh start
```

Test the configuration by using an SSH client[8] to connect to the AIX server using password authentication. You will not be able to log in as the root account, you will need to use the account of a systems administrator that has been defined on the server.

As a side note, the use of public key authentication is more secure than password-based authentication but the configuration and installation of those facilities is not covered in this document. To read more about how to use public key authentication, check out the resources on http://www.openssh.org.   The reason that password authentication is weaker is not because of the encryption: session encryption in SSH is the same for all authentication mechanisms. The problem is that password authentication is subject to password "brute force" attacks in much the same way a standard telnet or FTP daemon is subject to those attacks. Anyone with a properly-configured SSH client (which are freely available on the Internet) can access your SSH server and attempt to brute force user accounts and passwords fairly easily. Public key authentication, however, is not subject to the same attack.

Installing and Configuring Sendmail

Now for the final configuration to make this server operational for its intended purpose. As stated in the first section of this paper, this server will be configured as a Sendmail bastion server. The bastion server is the first line of defense between the Internet and our "real" internal mail server. It does not matter for the

---

[8] A list of Unix, Macintosh and Windows clients can be found on the OpenSSH website at http://www.openssh.org under the Alternatives section.

purpose of this paper what the internal mail server is running, as long as it accepts incoming SMTP connections. We will not discuss the security of the internal mail server at all, only how to get mail from the Internet, through this server, and onto the internal mail server. What this does is add a layer of protection such that if this Sendmail server was compromised in some way, there are no mailboxes or mail stores on this server at all for an intruder to look at: all the mail is forwarded to the internal mail server for processing and only briefly may touch the incoming /var/spool/mqueue directory while on its way. Also, since no users log into this system, except systems administrators, an intruder would not have a list of all the user accounts for the mail users since none of them reside on this server either.

A brief diagram will show the configuration:
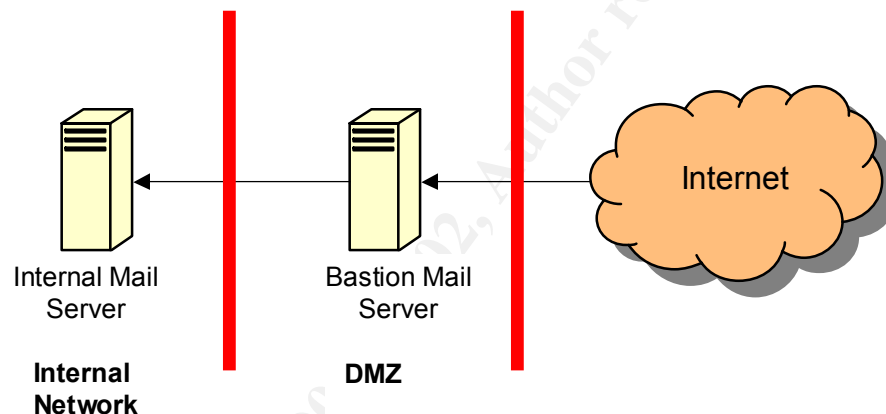
The bastion mail server will reside in the network's DMZ, which is one step from the Internet and considered "untrusted" to the protected internal network. Rules in the firewall and routing software between the DMZ and the internal network will only allow specific protocols to pass from the bastion mail server to the internal network. The protocols that would be required are:

TCP port 25	To deliver mail to the internal mail server
TCP port 22	To enable administrators to use SSH to log into the server
UDP and TCP port 53	To enable the bastion server to resolve IP addresses and hostnames. If the DNS server is on the DMZ, then this rule is not required. Also, if the DNS server is on the Internet, outside the DMZ, then this rule would apply to the DMZ->Internet connection instead of the DMZ->internal connection

UDP port 123	Needed only if the NTP server is inside the internal network. If it is on the Internet, then this rule would apply to the DMZ->Internet connection
UDP port 514	To enable the syslog server to pass events to the internal syslog host, unless that host is also on the DMZ (not recommended)

The firewall and routing software for the hardware between the bastion server and the Internet need to only allow for TCP port 25 so that email from the Internet will be able to make it to the DMZ and onto the bastion sendmail server, unless one of the exceptions is met for DNS and/or NTP in the table above. The configuration of the firewall and router software is not a topic in this paper, but the table above is enough to relay the basic information on the role this bastion server will play in the delivery of email to the organization.

First, we need to do some preliminary tasks for Sendmail to make sure we have the properly configured environment prior to running the Sendmail daemon. Make sure that all the directories that Sendmail uses have the proper permissions. Run the following commands to set this in accordance with the recommendations provided by the developers of Sendmail [6]:

```
# chmod 0700 /var/spool/mqueue
# touch /var/adm/maillog
# chmod 600 /var/adm/maillog
# chmod 700 /etc/mail
# chown root:system /etc/mail
# rm /etc/mail/*
# vi /etc/netsvc.conf
```

Add this line to the /etc/netsvc.conf. Remove any other lines:

```
hosts=local,bind
```

Make sure there are no spaces or tabs anywhere in this line!

To install Sendmail, run the following command:

```
# installp –acqgQNX –d /usr/local/src freeware.sendmail
```

Now we will build a Sendmail configuration file that will meet the needs of this server. "cd" into the /usr/local/lib/sendmail-8.11.6/cf/cf directory. Create a new file called <hostname>.mc where <hostname> is the fully qualified hostname of this server (i.e. mail.acme.com.mc).

```
# vi mail.acme.com.mc
```

Add the following lines to this file:

```
include(`../m4/cf.m4')dnl
OSTYPE(`aix5')dnl
FEATURE(`dnsbl',`relays.ordb.org')dnl
undefine(`confFORWARD_PATH')dnl
undefine(`ALIAS_FILE')dnl
define(`confTO_IDENT',`0')dnl
define(`confSMTP_LOGIN_MSG', `$j Sendmail; $b')dnl
FEATURE(`no_default_msa')dnl
FEATURE(`use_cw_file')dnl
FEATURE(`stickyhost')dnl
define(`MAIL_HUB', `internal-mail')dnl
define(`SMART_HOST', `internal-mail')dnl
FEATURE(`access_db',`dbm /etc/mail/access')dnl
FEATURE(`blacklist_recipients')dnl
LOCAL_USER(`root')dnl
define(`confPRIVACY_FLAGS',`goaway')dnl
MAILER(`smtp')dnl
```

Substitute the "*internal-mail*" with the fully qualified hostname of your internal mail server.

The explanation of each line is:

Line	Explanation	Rationale
include(`../m4/cf.m4')dnl	Standard "include" line for the m4 preprocessor to be able to build Sendmail configuration files (.cf)	Required
OSTYPE(`aix5')dnl	Defines the proper constructs and operating system specific values.	Required
FEATURE(`dnsbl',`relays.ordb.org')dnl	Uses the ORDB "realtime blackhole" list[9]	Assists in stopping unwanted SPAM from "open relays".
undefine(`confFORWARD_PATH')dnl	Turns off the use of .forward files	No users have accounts on the system, so .forward files are not required and could pose a security risk if misconfigured.

---
[9] See http://www.ordb.org for a discussion of Blackhole lists and their function

undefine(`ALIAS_FILE')dnl	Turns off expansion of Aliases	Not needed since no users have accounts on this system. Gives a small performance boost.
define(`confTO_IDENT',`0')dnl	Turns off IDENT to remote mail servers.	Speeds up mail delivery by not requiring an IDENT response from remote mail server, which is easily spoofed anyway.
define(`confSMTP_LOGIN_MSG ', `$j Sendmail; $b')dnl	Defines the message that is seen when clients connect to port 25.	The default message give AIX version and Sendmail version statistics. This is a more "sanitized" version giving only the hostname ($j) and the date/time ($b).
FEATURE(`no_default_msa')dnl	Turns off the Mail Submission Agent which listens on port 897.	Users are not sending mail from this system, so an MSA is not required. Saves us from a possible vector of attack by reducing the number of daemons listening on TCP/IP ports.
FEATURE(`use_cw_file')	By adding hostnames to the file /etc/mail/local-host-names we can accept mail not addressed directly to mail.acme.com.	We need to accept mail for the entire acme.com domain, so we need this option.
FEATURE(`stickyhost')dnl	Turns on the "stickyhost" feature.	Mail addressed to user@acme.com will not be re-written as user@internal-mail-server. Keeps the recipient's address the same after being fowarded to the internal mail server.

FEATURE(`access_db',`dbm /etc/mail/access')dnl	Allows the use of access lists to control who we accept mail from/to via /etc/mail/access file.	Let's us block specific senders, like spammer@spam.com if we choose to. Used with blacklist_recipients it lets us also block mail to a specific address.
FEATURE(`blacklist_recipients') dnl	Extends the access list to include recipient addresses as well as sender addresses.	If we have local distribution lists that are for internal use only, we can block incoming Internet email from being sent to that list. This is another way of blocking unwanted email.
define(`SMART_HOST', `internal-mail')dnl	Defines where mail addressed for other hosts is sent.	Incoming mail addressed for acme.com instead of mail.acme.com (this system's name) gets sent to the internal mail server.
define(`MAIL_HUB', `internal-mail')dnl	Defines where mail addressed for this host is sent.	Incoming mail with the fully-qualified name of this host, user@mail.acme.com is sent to the internal mailer because no users have accounts on this server. Mail sent locally, without any @ at all, like 'root' are still delivered locally.
LOCAL_USER(`root')dnl	Any mail addressed to root will be delivered locally and not forwarded to the internal mail server.	It is important to keep local mail for root on this system and not forward it to another system where it may get confused with that systems "root" mail.

define(`confPRIVACY_FLA GS',`goaway')dnl	Turns off commands such as EXPN and VRFY.	The EXPN and VRFY commands can be used maliciously to "harvest" email addresses or check for valid user accounts. Turning these features off means that in order to send mail to acme.com through this server you need to already know the email address of your recipient.
Mailer(`SMTP')dnl	Defines the SMTP mailer	Required in order to send and receive SMTP mail.

Then compile the .mc file with the m4 command to create the configuration file for Sendmail, /etc/mail/sendmail.cf:

```
# m4 mail.acme.com > /etc/mail/sendmail.cf
```

Before starting Sendmail for the first time, we need to create a few files that we defined in the configuration file. The first is the access database, or map. We are not going to add any recipients or senders right now, but the file does have to exist for Sendmail to work properly, even if it's empty. First, create the empty /etc/mail/access file:

```
# > /etc/mail/access
```

Then, create the database (dbm) version of the file, called the map:

```
# /usr/local/bin/makemap dbm /etc/mail/access < /etc/mail/access
```

If you look at the /etc/mail directory, you should have three files for the access map: access, access.pag and access.dir. The .pag and .dir files are used by the DBM routines in Sendmail to quickly search the list of allowed/denied addresses. The text file is where you add new addresses. Don't forget to re-run the makemap when you make changes to the /etc/mail/access file. For a complete discussion of how to use the access map with Sendmail, see http://www.sendmail.org/m4/anti-spam.html.

The next file we need to create is /etc/mail/local-host-names. This is a simple text file, which lists one host or domain name on each line that this server will accept mail for. Sendmail automatically adds (internally) the name of the local

server, like mail.acme.com to this list. If you want senders to be able to address mail to user@acme.com instead of user@mail.acme.com, then you need to use this feature. Just add the string "acme.com" to the top of this file. There is no makemap command necessary:

```
# echo "acme.com" >> /etc/mail/local-host-names
```

Lastly, make sure that all the files in /etc/mail are accessible only by root:

```
# chmod 600 /etc/mail/*
```

Now we need to have sendmail automatically start when the system reboots. The /etc/rc.tcpip is the script where the version of sendmail that comes with AIX is started, so we'll use that for our new version. First, edit the /etc/rc.tcpip file. Find the line where we commented-out sendmail earlier. Replace the line with this one, and make sure it's not commented-out:

```
start /usr/local/bin/sendmail "$src_running"
```

Next, we need to place the new Sendmail under the control of SRC. Since the old Sendmail is currently configured for that, we need to first remove the configuration of the "old" Sendmail:

```
# rmssys -s sendmail
```

Then, create the new SRC definition:

```
# mkssys -s sendmail -p "/usr/local/bin/sendmail" -u root \
-a "-bD" -S -n 15 -f 9 -G mail
```

Finally, start it:

```
# startsrc -s sendmail

0513-059 The sendmail Subsystem has been started. Subsystem PID is 7576.
```

And to check that it's running ok:

```
# lssrc -s sendmail

Subsystem         Group           PID     Status
 sendmail         mail            7576    active
```

Now that Sendmail is running, we need to make sure that the System Dump device we defined is large enough so that if the system dies, we can save the dump for analysis. Use the sysdumpdev command to check this:

```
# sysdumpdev -e

453-041 Estimated dump size in bytes: 47185920
```

We created a 48MB dump device, so we're still ok. If it was not big enough, you could add more physical partitions to the dumplv logical volume like we did in the installation section.

Testing Sendmail

Now we should test that our Sendmail server is working correctly, before we add it to DNS as the 'MX' record for acme.com. The most simple way to test is to telnet to port 25 of this server from a remote client, like a workstation. When you connect to port 25, you should see the custom login banner that we configured:

220 availaix.acme.com ESMTP Sendmail; Fri, 20 Jun 2001 18:20:31 -0400

Test that the Privacy Options are set correctly. Type these commands in the telnet session. The lines in **bold** are what the server should send back to you:

```
help
```
502 5.3.0 Sendmail 8.11.6 -- HELP not implemented

```
expn root
```
502 5.7.0 Sorry, we do not allow this operation

```
vrfy root
```
252 2.5.2 Cannot VRFY user; try RCPT to attempt delivery (or try finger)

So far so good. Now test that mail cannot be relayed through our server. Relaying is used by spammers to hide their identity when sending unsolicited email. A misconfigured Sendmail server could be sending messages on behalf of someone else without even knowing it! Use these commands to test this:

```
helo acme.com
```
250 mail.acme.com Hello workstation [0.0.0.0], pleased to meet you

```
mail from: <test@acme.com>
```
250 2.1.0 <test@acme.com>... Sender ok

```
rcpt to: <test@hotmail.com>
```
550 5.7.1 <test@hotmail.com>... Relaying denied

```
rcpt to: test%hotmail.com@acme.com
```
550 5.7.1 <test%hotmail.com@acme.com>... Relaying denied

Everything looks good again. Now test sending an email to a real user at acme.com:

```
rcpt to: realuser@acme.com
250 2.1.5 <realuser@acme.com>... Recipient ok

data
354 Enter mail, end with "." on a line by itself
Subject: test email
From: me
To: you
Test test test
.

250 2.0.0 g5LMd5t12905 Message accepted for delivery

quit
221 2.0.0 mail.acme.com closing connection
Connection closed by foreign host.
```

Now check the mail server's log /var/adm/maillog for a message similar to this one:

```
Jun 01 15:21:33 mail sendmail[8492]: g5JJLWO08492:
to=<realuser@acme.com>, delay=00:00:01, xdelay=00:00:01,
mailer=relay, pri=30128, relay=internal.mail.acme.com
[24.95.72.95], dsn=2.0.0, stat=Sent (g5JJLXX31884 Message
accepted for delivery)
```

This shows that the email was sent to "internal.mail.acme.com" which in our case is the name of the internal mail server. Verify with the user that the mail was delivered. After testing more deliveries, and only when you are comfortable that everything works properly, inform the group responsible for maintaining your organization's DNS server that the new Mail Exchanger (MX) record should be changed to the name of this mail server: mail.acme.com in our case.

Configuration of the Trusted Computing Base

The Trusted Computing Base is a facility in AIX that is responsible for enforcing the information security policies on the system [7]. Items like hardware configuration, file permissions and other attributes are maintained by the TCB. An administrator can add or subtract from this list, and periodically run reports to assess the integrity of the system. For instance, if a compromise of the system is suspected, the systems administrator can run the "tcbck" program to see if any critical operating system files had been modified. Now that we have configured AIX and all the supporting software, we need to tell TCB how our configuration has changed, and add some new options for our environment.

First, we need to tell TCB that some of the files that used to be owned by the shutdown group are now owned by the system group. To modify the entries for the files that were owned by the shutdown group, run these commands:

```
# tcbck -y /usr/sbin/exec_shutdown
# tcbck -y /usr/sbin/reboot
```

You will receive 2 warning messages when you run each command. This is tcbck's way of telling you that the configuration of these files is not what it was when AIX was installed. You can safely ignore these. The next step is to add our Sendmail binaries and the /etc/mail/sendmail.cf and all the /etc/mail/ files. This way we will know if one of these critical files has been changed:

```
# tcbck -a /usr/local/bin/sendmail checksum size group owner \
mode symlinks="/usr/local/bin/hoststat,/usr/local/bin/mailq,\
/usr/local/bin/newaliases,/usr/local/bin/purgestat"

# tcbck -a /usr/local/bin/hoststat mode owner group
target="/usr/local/bin/sendmail"
# tcbck -a /usr/local/bin/mailq mode owner group
target="/usr/local/bin/sendmail"
# tcbck -a /usr/local/bin/purgestat mode owner group
target="/usr/local/bin/sendmail"
# tcbck -a /usr/local/bin/newaliases mode owner group
target="/usr/local/bin/sendmail"

# tcbck -a /usr/local/bin/makemap checksum size group owner mode
# tcbck -a /etc/mail/sendmail.cf checksum size group owner mode
# tcbck -a /etc/mail/access checksum size group owner mode
# tcbck -a /etc/mail/access.pag checksum size group owner mode
# tcbck -a /etc/mail/access.dir checksum size group owner mode
# tcbck -a /etc/mail/local-host-names checksum size group \
owner mode
```

This will add information to the TCB configuration database for each file that will make sure there were no changes to the file itself (checksum,size), the owner of the file (owner), the group owner of the file (group) or the permissions (mode). Additionally, for the sendmail binary itself, we want to make sure no links are made to it from other programs, except the ones we already know about (symlinks). And, for those programs that are linked to Sendmail, we want to know if someone changes that link to point to another file (target). Since those files are links to the Sendmail program, we can't use the size or checksum attributes.

To check this, try this exercise. First, run the tcbck command to verify the integrity of the /etc/mail/sendmail.cf file:

```
# tcbk -n /etc/mail/sendmail.cf
```

As long as there is no output, the file integrity is assured by TCB. Next, edit the /etc/mail/sendmail.cf file with vi. Just add a new blank line at the bottom of the file and save it. The, run the tcbck program again:

```
# tcbck -n /etc/mail/sendmail.cf

3001-028 The file /etc/mail/sendmail.cf has the wrong checksum value.
3001-049 The file /etc/mail/sendmail.cf has the wrong file size.
```

To verify the integrity of all the files on the AIX server, run tcbck this way:

tcbck –n ALL

This will report on all exceptions for all files managed by TCB. Some errors, such as owner/group errors on a /dev/pts device can be ignored most of the time. When a user logs into the system, the ownership of their terminal device in /dev is changed so that the user can read and write to their terminal. This is normal behavior for the system.

It is good practice to periodically check the status of the files in TCB. You may want to set up a scheduled job in cron that runs tcbck periodically and emails the results to a systems administrator.

Setting Up Packet Filtering

The IP Security filesets that were installed in the beginning of this paper will allow us to do packet filtering on the AIX system. By "shutting off" all the unnecessary TCP/IP ports at the network level, we can better ensure that a configuration error of a network service, like accidentally turning on Telnet, will not expose the system. We will set up the IP Security software to filter out all but the most necessary ports to incoming traffic, namely TCP port 25 for Sendmail, TCP port 22 for Secure Shell, UDP port 123 for NTP and UDP port 514 for remote syslogging. All other ports will be denied incoming traffic regardless of whether or not a network service is "listening" on the port. The original idea for this section of the paper, using the AIX IP Security features, came from reference [8].

For this section, perform all the commands at the console, not via an SSH session. During configuration, network services will be interrupted and if you are logged in via SSH, your session will be disconnected.

First, we want to setup our syslog daemon to actively capture events that are generated by the IP Security software on the local4 facility. Add the following line to /etc/syslog.conf file if it was not already added:

```
local4.debug        /var/adm/ipsec
```

Next, make sure the /var/adm/ipsec file exists, and is only accessible by root and the sysadm group:

```
# touch /var/adm/ipsec
# chown root:sysadm ipsec
# chmod 0640 /var/adm/ipsec
```

Then restart the syslog daemon so that the changes will take effect:

```
# refresh -s syslogd
```

To configure the IP Security software, first flush all the default rules so that we are sure we are starting from a clean slate:

```
# rmfilt -v4 -n all
```

Then, configure the software to automatically start the IP Security "device" when the system reboots:

```
# smitty ips4_start
```

You will see the following screen:

Change the "Deny all Non_Secure IP Packets" to "yes". This sets the default rule to deny all packets unless we have specifically allowed them. Press ENTER to

configure the changes. To make sure that the ipsec device was loaded successfully, run the following command:

```
# lsdev -Cc ipsec
```

And the output should be:

```
ipsec_v4 Available  IP Version 4 Security Extension
```

Now, we will start configuring our rules. Let's start by allowing Secure Shell access:

```
# genfilt -v4 -a P -s 0.0.0.0 -m 0.0.0.0 -d 0.0.0.0 -M 0.0.0.0 \
-c tcp -O eq -P 22 -r B -w I -i all
```

A brief explanation of each parameter follows:

Parameter	Value	What it means
-v	4	What IP version are we working with?
-a	P	Permit this packet.
-s	0.0.0.0	Source address. 0.0.0.0 means "any".
-m	0.0.0.0	Network mask of source address. 0.0.0.0 is no mask.
-d	0.0.0.0	IP address of destination. 0.0.0.0 means any destination (our IP address).
-M	0.0.0.0	Network mask of destination IP address. 0.0.0.0 means no mask.
-c	TCP	What protocol? TCP, UDP, ICMP, and others.
-O	eq	When we define the destination port, how do we want the rule to compare it? We want the destination port to be "equal to" what we define with the "-P" parameter.

-P	22	Destination port.
-r	B	What type of packets will this rule apply to: forwarded, local or both? B is both.
-w	I	Direction of traffic: Inbound or Outbound?
-I	all	Interfaces to apply this rule to?

This rule turns on access for TCP packets originating from anywhere to local port 22. If you want to define rules so that only particular IP addresses or subnets are allowed access, change the –s and –m parameters to your IP address and subnet values.

Now we need to set up a reflexive rule so that traffic originating from the local TCP port 22 can get back to the remote client:

```
# genfilt –v4 –a P –s 0.0.0.0 –m 0.0.0.0 –d 0.0.0.0 –M 0.0.0.0 \
-c tcp -o eq -p 22 -r B -w O -i all
```

Notice the options that changed: the "-O" from the first command is "-o" now, the "-P" from the first command is now "-p" and the "-w" value is now "O" for outbound.

Now, set up the rest of the rules for Sendmail, syslog and NTP:

Sendmail:

```
# genfilt –v4 –a P –s 0.0.0.0 –m 0.0.0.0 –d 0.0.0.0 –M 0.0.0.0 \
-c tcp -O eq -P 25 -r B -w I -i all

# genfilt –v4 –a P –s 0.0.0.0 –m 0.0.0.0 –d 0.0.0.0 –M 0.0.0.0 \
-c tcp -o eq -p 25 -r B -w O -i all
```

NTP:

```
# genfilt –v4 -a P –s <ipofNTPserver> –m 255.255.255.255 \
-d 0.0.0.0 -M 0.0.0.0 -c udp -O eq -P 123 -r B -w I -i all

# genfilt –v4 -a P –s 0.0.0.0 –m 0.0.0.0 -d <ipofNTPserver> \
 -M 255.255.255.255 -c udp -o eq -p 123 -r B -w O -i all
```

Notice for this rule we are only specifying the IP address of the NTP server that was configured in the /etc/ntp.conf file. Substitute that IP

address in place of *<ipofNTPserver>*. The network mask of
255.255.255.255 says "only this host, not hosts in the same subnet."

Syslog:

```
# genfilt -v4 -a P -s <ipofsyslogserver> -m 255.255.255.255 \
-d 0.0.0.0 -M 0.0.0.0 -c udp -O eq -P 514 -r B -w I -i all

# genfilt -v4 -a P -s 0.0.0.0 -m 0.0.0.0 -d <ipofsyslogserver> \
 -M 255.255.255.255 -c udp -o eq -p 514 -r B -w O -i all
```

As with NTP, we are going to lock down the syslog port so that we only
communicate with our internal loghost.

Another item that we will need, which is sometimes forgotten, is the ability to
perform DNS requests so that we can resolve hostnames. The following rules
will take care of that as well:

```
# genfilt -v4 -a P -s 0.0.0.0 -m 0.0.0.0 -d <ipofDNSserver> \
-M 255.255.255.255 -c udp -O eq -P 53 -r B -w O -i all

# genfilt -v4 -a P -s <ipofDNSserver> -m 255.255.255.255 \
-d 0.0.0.0 -M 0.0.0.0 -c udp -o eq -p 53 -r B -w I -i all
```

If you have multiple DNS server defined in the /etc/resolv.conf, make rules for
each one.

Oh, one more thing: at this point everything is denied except the 5 items above.
If you want ICMP to function normally (i.e. ping requests) we need to allow ICMP
traffic as well:

```
# genfilt -v4 -a P -s 0.0.0.0 -m 0.0.0.0 \
-d 0.0.0.0 -M 0.0.0.0 -c icmp -r B -w B -i all
```

To activate the rules:

```
# mkfilt -v4 -u
```

To list all of the rules:

```
# lsfilt -v4
```

To deactivate the IP Security Code:

```
# mkfilt -v4 -d
```

Make sure you can still synchronize time with the NTP server, look up hostnames, and receive incoming email. If something doesn't work correctly, deactivate the IP Security code and check the rules with the lsfilt command to see what's wrong. Check the /var/adm/ipsec log for informational messages.

When everything is correct, you're finished! The filters you have configured will take affect on each reboot, unless you change the startup options via smit or the mkfilt program.

Closing

At this point you should have a relatively secure server. The problem with security, though, is that it's cyclical. You can't stop here and rest on your backside and assume that all is. First, take the time to check the logs on your server periodically for any signs of malicious activity. If you are running a central syslog server, as this paper recommends, you can view all the logs from all of your systems at one time. This is a convenience, especially if you have a large site.

On the server itself, a number of commands can help you audit your system. The who[10] command has options to show you currently logged in users, past user logins, and past failed logins. The who command can be used to see if someone is actively trying to compromise your system via a user or system account. The file /var/adm/sulog[11] also keeps tabs on who's been using the su command, and if they were successful or not. This can also be helpful to provide an audit trail of user or malicious activity. Lastly, don't forget to look at the system error log. We configured the errdaemon to log events to syslog, but it is still wise to view the error log periodically to get the details on each event. Use the errpt command, like we did in the section "Tying the Error Report into Syslog."

Make sure that you stay abreast of operating system fixes and maintenance levels, and most importantly find a way to be notified of security patches that are released for software or configuration bugs. The Computer Emergency Response Team at Carnegie Mellon university is an excellent place to start. The website for CERT is: http://www.cert.org. There are many links to current vulnerabilities and their fixes, as well as interesting reading on Security Best

---

[10] AIX 5L Version 5.1 Commands Reference Volume 6: who Command
http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/cmds/aixcmds6/who.htm
[11] AIX 5L Version 5.1 Commands Reference Volume 5: su Command
http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/cmds/aixcmds5/su.htm

Practices and other items of interest. For IBM-specific issues, you can subscribe to a multitude of email lists at http://techsupport.services.ibm.com/server/listserv.

Again, security is an ongoing effort. It takes diligence to stay on top!

References

Cited References

[1] "Adding and Activating a Paging Space." <u>AIX 5L Version 5.1 System Management Guide, Operating Systems and Devices</u>.
<<u>http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/baseadmn/baseadmntfrm.htm</u>>

[2] "Reorganizing Logical Volumes." <u>AIX 5L Version 5.1 Performance Management Guide</u>.
<<u>http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/prftungd/2365c86.htm</u>>

[3] "Reorganizing JFS Log and Logical Volumes." <u>AIX 5L Version 5.1 Performance Management Guide</u>.
<<u>http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/prftungd/2365c88.htm</u>>

[4] Seigert, Andreas. <u>The AIX Survival Guide</u>. Addison-Wesley, 1998.

[5] "login.cfg File." <u>AIX 5L, Version 5.1 Files Reference</u>.
<<u>http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/files/aixfiles/login.cfg.htm</u>>

[6] Allman, Eric. <u>Sendmail Installation and Operation Guide</u>. Vers. 8.317.4.71. Sendmail, Inc.

[7] "The Trusted Computing Base." <u>AIX 5L Version 5.1 Systems Management Concepts: Operating Systems and Devices</u> .
<<u>http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/admnconc/tcb.htm</u>>

[8] Sklar, Sandor. <u>Packet Filtering with IPSec</u>.
<<u>http://www.ibm.com/servers/esdd/tutorials/ipsec/index.html</u>>

Additional References

Farazdel, Abbas, et al. <u>Additional AIX Security Tools on IBM eserver pSeries, IBM RS/6000 and SP/Cluster</u>. IBM Corporation, International Technical Support Organization, 2000.

"no Command." <u>AIX 5L Version 5.1 Commands Reference, Volume 4.</u>
<<u>http://publibn.boulder.ibm.com/doc_link/en_US/a_doc_lib/cmds/aixcmds4/no.htm</u>>