Global Information Assurance Certification Paper

Securing UNIX GCUX Practical Assignment
Version 1.8 (revised October 26, 2001)
Option 1- Securing UNIX Step by Step
AIX Version 4.3.3 on Power2 3xx Series RS/6000

John Jenkinson March, 2002

Table of Contents

Overview

The assignment is to take a "host from out-of-the-box to a secure, operational state." Most of the other papers do this with a new system. We don't have that luxury. This paper will address some of the issues in taking a used machine to replace a more used machine while doing some server consolidation AND preserving legacy applications. This probably represents a more real-world assignment, at least in our case. The paper will cover all the aspects of a new system install, but will add issues dealing with the additional aspects given above. Thus the step-by-step guide will not be as strict as one can be with a new system installation and the other aspects of the assignment will be a superset of the new system installation.

Description of System

The system is currently an IBM RS/6000 Model 360 running AIX Version 4.3.3. The system is to be an IBM RS/6000 Model 39H running AIX Version 4.3.3. This is not a simple hardware upgrade as the model 360 is Power based while the model 39H is Power 2, thus a different kernel which is close to a new Operating System installation in the manner it is applied. The RS/6000 model line was renamed by IBM to be the e(logo) server pSeries recently, but these
two systems are still badged RS/6000 and were withdrawn from marketing before that rename. The model 39H has enough performance improvement features to justify the upgrade as seen in the table below, but the reason for this upgrade is the availability of spare parts for the hardware as we are hardware and software self-maintenance.

Attribute	Model 360	Model 39H
Processor	Power 50MHz	Power2 67MHz
Peripheral Bus	Micro Channel	Micro Channel
Instruction Cache	32KB two-way set-associative	32KB two-way set-associative
Data Cache	32KB four-way set-associative	128KB four-way set-associative
Level 2 Cache	Optional	N/A
Memory	128MB ECC	128MB ECC 4-word wide
Disk Controller	SCSI	SCSI-2 fast-wide
Ethernet	10Mbit AUI/bnc	10Mbit AUI/bnc

The security aspects of the upgrade are many. We have the project to cover any security enhancements to the current system. We are under cost challenges thus the costs of any enhancements to security, though justifiable, are more easily chartered under this other project.
The older RS/6000 models have quite a few security features in the hardware that are not generally available in other systems.

1) A real key – this key is part of a lock that locks the system case, implements one of three operating modes for the machine (Secure, Maintenance, Normal), is furnished in duplicate, made of metal, has the notches cut on angle so they can't be easily duplicated, and protects the power switch. If the key is in the secure position and removed, the power switch will not function. This prevents someone with physical access but no key from powering the system on or off which enhances security but could cause a problem it the machine were to catch fire. From the factory the key and its duplicate come

with a tag with the key number stamped on it to facilitate obtaining a replacement key. The keys for machines obtained from used computer sources are usually an issue, be sure to get a machine with both keys and the correct tag with the key number. In the 5xx series of the same machine, i.e. model 59H, the key also locks the back cover in place which prevents the power cord or the network connection from being pulled. The 39H does not have that protection. The key position change generates an event that can trigger an alarm as we have a daemon to poll the switch position.

2) The fans are more protected than on most systems, but still vulnerable to having their motion stopped by an implement which will cause the machine to shutdown. The error logger will indicate this condition.
3) A 3 digit 7 segment LED display provides the status of boot up checks, but can be also used to do real-time displays of aspects of the machine once booted and running. We use the display to show the number of tasks in the CPU run queue. This gives a quick sanity check on the machine's operation by observing the LED display without having a session on the machine. Thus operations can do a quick visual inspection of a rack of these and observe any increased level of CPU activity and notify us or logon to check the reason for the increased level of CPU activity.
4) Another feature of sort is the response to someone disconnecting the keyboard from a running system. When this occurs the system changes from the keyboard map it loaded at boot up after querying the keyboard for its type to a generic keyboard map. The change is quite noticeable and can be used as an indication that someone may have inserted a keyboard sniffer into the keyboard cord. These devices are getting quite small and hard to detect since the standard connector at the back of the machine.

The system is both a workstation for a system administrator and a server for several infrastructure services like SMTP mail gateway, secondary BIND/DNS server, slave NIS (Network Information Services) server, sysback server, NFS server for the filesystem holding patches, and other tasks that do not fit well elsewhere. Most of our other RS/6000s are production servers so when new applications are being evaluated they start life on this machine. This spans the spectra from games to very intense graphical applications specific to our industry. This system has been in service on older models since the RS/6000 was introduced in 1991, thus a lot of old applications and utilities are on the system and need to survive the hardware upgrade. If you have an old system over 10 years old you probably know the problem. An example is an application that takes old graphical formats and produces PostScript. That application is sans source code so it can not be rebuilt. IBM as done a very good job of forward compatability as this specific example and several other similar applications have been runniing on AIX since version 3.1 without a recompile or rebuild from object libraries.

As services always grow in demand for resources, the disk capacity of the replacement system will increase. This increase in disk capacity has some security benefit, as more log space will be available to prevent an attack from filling the currently available disk space. The increase in speed for the SCSI controller will help reduce system disk latency if the system were to be under attack.

Risk Analysis of System

The system needs a high level of security for the key services it performs.

The system needs to run X Windows to be useful on the system administrator's desk.

Not the best of configurations, but not that unusual in practice.

The system is on a different network topology (ethernet) than the other critical servers (FDDI) such that we can survive the main computer room failure using this machine as servers for NIS, DNS, NFS, ntp, etc. – the business continuity aspect of security. It is hard to test such a configuration, so our computer room UPS failed recently to allow this testing in a real world case.

For component risk analysis we will use the SANS method of severity of an exploit

(Criticality + Lethality) – (System + Network)

where:

Criticality is a measure of how critical the component is to the system thus the enterprise

Lethality is a measure of how lethal a compromise of the component would be to the system thus the enterprise

System is a measure of how well equipped the system is to deal with the security of the component

Network is a measure of how well the network is configured to deal with the security of the component

This method is part of the SANS GIAC ciriculum for Intrusion Analyst. More detail can be found in the student's posted practical assignments. Althought the method is used for vulnerability assessment, it has the components of risk assessment as well.

The rating from 1 to 5, with 5 being the highest for each measure. Thus the range will be from –10 (where the component has no crticality and no lethality with maximum protection for both system and network) to +10 (maximum criticality and lethality and no system or network protection)

SMTP

The SMTP gateway is for smtp mail to/from the corporate standard of Microsoft Exchange for email. Thus email inbound is filtered for viruses and malware by other servers which also do other security aspects of email such as blocking attachments. DNS MX records do not involve this AIX gateway machine and its level of criticality to the company's operation is low. Sendmail V8.9.3 is the MTA (Mail Transport Agent). The system is not visible on the Internet but does provide the gateway to the company WAN (Wide Area Network). Thus the criticality is rated 2.

Sendmail has been vulnerable to a large number of exploits due to its complexity and its ability to be configured to do most anything for anybody. It is also undergone a lot of scrutiny due to those reasons and the long history of the application, thus lethality gets a 3.

AIX 4.3.3 uses sendmail 8.9.3 and IBM is fairly fast on issuing PTF (Program Temporary Fixes) for its sendmail giving a system rating of 4.

This server is on a private switched LAN behind a robust set of firewalls, border routers, and IDS (Intrusion Detection Systems) so we rate network a 4.

(2 + 3) – (4 + 4) = -3

DNS/BIND

The system is a secondary DNS server for the private namespace for the local site's zone. Though a secondary server, it is the only secondary server so criticality is rated a 4.

The version of BIND supported by IBM for AIX V4.3.3 is 8.2.2-P5. In recent BIND vulnerabilities IBM has been slow to get PTFs out, though emergency fixes have been provided. This makes the lethality rating a 4

AIX seems to have implemented most of the options for BIND V8, has the BIND subsystem under the control of SRC (System Resource Controller), and supports both BIND V4 and BIND V8 via a link to the named daemon so system rates a 3.

This server is on a private switched LAN behind a robust set of firewalls, border routers, and IDS so we rate network as a 4.

(4 + 4) – (3 + 4) = 1

NIS

For historical reasons and our need to support a large user base across a very heterogeneous mix of systems and platform types we still use NIS (Network Information System) on the LAN. We do have plans to look at Kerberos for authentication once our Windows 2000 backend is converted. All of our other NIS servers are on a different network topology (FDDI) so this is our only NIS server on Ethernet giving the criticality a 5.

NIS is lethal for exploits and information leaks making that a 5 also.

System gets a 3 as we do have a lot of logging and monitoring of NIS.

This server is on a private switched LAN behind a robust set of firewalls, border routers, and IDS so we rate network a 4.

(5 + 5) – (3 +4) = 3

NFS

The one filesystem exported from this system is a staging area for PTF downloads, emergency fixes from IBM, and we use it as a vehicle to get files to the other systems on the LAN via NFS. While we will soon use the SP filesystem to hold PTFs for the AIX nodes as well as the SP AIX nodes (a brief note about the SP is in Appendix A), the need for such a filesystem for the other uses will continue. This makes criticality rate a 2.

NFS is lethal. Authentication is via UIDs, stateless handling via filehandles, ability to spoof IP addresses, etc. Rate lethality a 4

AIX allows some security for NFS. We turn on portmon, use secure ports, etc. giving system a 3 rating

This server is on a private switched LAN behind a robust set of firewalls, border routers, and IDS so we rate network a 4.

(2 + 4) – (3 + 4) = -1

X Windows

X Windows is not critical to the machine, but needed by the single user of the machine as a workstation. X Windows is lethal so we have two 5's.

System gets a 4 rating as the single user of the machine is aware of the security problems and issues with X Windows and the system is not available for logins from normal users.

This server is on a private switched LAN behind a robust set of firewalls, border routers, and IDS so we rate network as 4.

(5 + 5) – (4 + 4) = 2

HARDWARE

As shown before the RS/6000 machine for this upgrade does have some physical security features. Hardware criticality should be a 5 as should the lethality rating.

System rating is given a 4 for the physical and security features of the replacement machine given above. The system is also behind locked doors with a keycard access control for off-shift hours, weekends, and holidays.

Network countermeasures get a 4 as before though it probably does not get the weight as it would for software.

(5 + 5) – (4 + 4) = 2

AIX Version 4.3.3

Security risks of an operating system are very subjective. AIX does have security features like the security administrator concept, ACL (Access Control Lists) for files, a journaled filesystem, resource limits, the Trusted Computing Base (TCB), access restrictions, password policies, security auditing, secondary authentication steps, etc. which gives it an overall risk rating of 2 in my opinion.

A summary risk rating (risks are additive) of 6 (-3 + 1 +3 –1 +2 +2 +2) on a scale of –10 to 10 with 10 being high risk.

Step by Step Guide

Most of the commands done in this effort will be done via SMIT (System Management Interface Tool). SMIT is a graphical and/or menu front-end to utilities on AIX. SMIT also does some rudimentary checks on parameters to make such commands less prone to user input errors. The command actually invoked by the system is available from the SMIT screens via the F6 function key as well as ~/smit.script thus for repetitive tasks SMIT can do the sanity and parameter checking, provide online help, providing a list of options for the parameters, and fill in fields so you do less typing. Once it is done once successfully, F6 gives the complete command so it can be repeated on the same or different machines. Since the command itself takes less space to show in documentation like this paper and is the actual command given to the system, the command will be used in this paper instead of the screen shots from SMIT.

Making the clone

As we are building a replacement system, it will be off the network during the build just as a new install would. Even if we changed the IP address and nodename to allow the replacement and current system to coexist, there is no reason to provide another target while the replacement is built and configured.
Also, as with a new build, the replacement system can be placed on a secure network segment if network connectivity is needed. In our case the machine is placed in a locked workroom inside the secured data center. We have such a facility to protect the new equipment until it is configured and deployed but it helps in the case of used equipment not yet deployed as well.

To make a clone when the machine architecture is different we use a combination of the mksysb command and the AIX V4.3.3 distribution CD-ROM set. To start the process take two mksysb backups and two sysback backups to removable tape such that the tape transport can be moved between the current and replacement machines or a like tape transport exists on both. Once these backups are made, protect them as anyone in physical possession of these backups has access to the files on the tapes, i.e. any file on the current system. The mksysb will be used to make the clone system, the sysback will provide a recovery of the original system. Two different commands in case one of the commands or utilities fails. Two tapes in case a tape fails.

> Note: mksysb is a bootable backup of the rootvg volume group and installation image. It is part of AIX and must be run on the local machine.
> sysback is an optional product that also makes a bootable installation image, but it can also capture other volume groups, allow manipulation of
> volume and filesystem sizes, and can run on another machine.

The mksysb command on the current machine:

```
/usr/bin/mksysb   '-i'  /dev/rmt1
```

The corresponding sysback command:

```
/usr/sbin/sbsmitout -s7 sysback -f'/dev/rmt0'  '-x' '-T rs6k' '-k up'
```

The replacement system is configured with the hardware and peripherals required and a diagnostic suite run to find any problems with the used equipment.
While the diagnostic supervisor is running take the time to ensure the microcode for all the components is up to date. Then the bootlist is setup to allow booting from the CD-ROM, the attached tape drive, the floppy, then the system disk. Then the AIX 4.3.3 installation CD-ROM Volume 1 is inserted, the key turned to the service position, and the replacement system powered on. Watch the LED display for the normal sequence through the POST (Power On Self Test) then the boot up states until the LED shows a console command

prompt. At this point the system should be displaying the text on the graphics display and the terminal(s) connected to the asynchronous serial ports. The screen/terminal should look similar to:

```
 ******* Please      define the System Console. *******

Type a 2 and press Enter to use this terminal as the
  system console.
Typ een 2 en druk op Enter om deze terminal als de
  systeemconsole to gebruiken.
Pour definir ce    terminal comme console systeme, appuyez
  sur 2    puis sur Entree.
Taste 2    und anschliessend die Eingabetaste druecken, um
  diese   Datenstation als Systemkonsole zu verwenden.
Premere   il tasto 2 ed Invio per usare questo terminal
  come console.
Escriba   2 y pulse Intro para utilizar esta terminal como
  consola del sistema.
Escriviu 1 2 i premeu Intro per   utilitzar aquest
  terminal com a consola del sistema.
Digite um 2 e pressione   Enter para utilizar este terminal
  como console do sistema.
```

which will be on each attached display device with a different number for each. Select the display of choice and the next screen appears:

```
>>>  1 Type 1 and press Enter to have English during install.
     2 Entreu 2  i premeu Intro per veure la installaci en catal.
     3 Entrez 3  pour effectuer l'installation en franais.
     4 Fr Installation in deutscher Sprache 4 eingeben
       und die Eingabetaste drcken.
     5 Immettere 5 e premere Invio per l'installazione in Italiano.
     6 Digite 6  e pressione Enter para usar Portugus na instalao.
     7 Escriba 7 y pulse Intro para usar
       el idioma espaol durante la instalacin.
     8 Skriv 8 och tryck ned Enter = Svenska vid installationen.
```

Selecting English then gives:

```
                    Welcome to Base Operating System
                    Installation and Maintenance

Type the number of your choice and press Enter.  Choice is indicated by >>>.

>>> 1 Start Install Now with Default Settings
    2 Change/Show Installation Settings   and Install
    3 Start Maintenance   Mode for System Recovery
```

We want Maintenance Mode for System Recovery so:

```
Type the number of your choice and press Enter.

>>> 1 Access a Root Volume Group
    2 Copy a System Dump to Removable Media
    3 Access Advanced Maintenance Functions
    4 Install from a System Backup
```

Select option 4, then:

```
                    Choose mksysb Device

Type the number of the device containing the system backup to be
installed and press Enter.

      Device Name                   Path Name
```

```
>>>   1 tape/scsi/ost                /dev/rmt0
      2     cdrom/scsi/enhcdrom          /dev/cd0
```

Select the correct device with the mksysb tape. This process then loads the appropriate kernel and supporting files for the new hardware and "clones" the system by copying the rootvg volume group to the selected disk(s) on the replacement system. You can place the key to the normal position so the system will reboot and configure the devices once the mksysb tape is restored. The copying of the system in this way gets all the filesystems, the files, the directory structure, etc. on the replacement system with a few minor changes. Terminal attributes are reset, the devices are rebuilt even if the device existed on the current system. The security issues here are the permissions on the devices and the terminal characteristics – thus you'll need to (re)set these as desired. Examples are taking world access from floppy disks, tape transports, etc. and keeping modem control on terminals such that disconnecting and reconnecting a serial line will drop the current terminal session.

Other tasks required for a new install are not required in our case of updating the hardware. These tasks not related to security include setting the timezone,
adding licenses, adding printers and print queues, etc. Tasks related to security taken care of by cloning include setting the root password, ntp configuration, setup of accounting, applying mandatory updates, applying maintenance levels etc.

Making a clone takes time. That time is ticking by after the initial mksysb tape is made and files are being written and modified on the current system in the interim. Also, on the replacement system are files and data that are not relevant to the replacement system. On the replacement system /smit.log, /smit.script, errlog data, etc. need to be removed. On the replacement system things like /var/spool/mail/*, syslog files, etc. need to be updated from the current system before actual replacement.

Installing Optional Software

AIX has filesets, which contain a specific function. Then there are packages, which is a group of installable filesets to provide a set of like functions. Then there are bundles, which are a grouping of filesets, packages, and/or files. Doing an install from scratch would require a review of the supplied bundles, packages, and filesets to meet the requirements of the machine. In that exercise you would want to install only those items needed to perform the tasks required for the system. We have the opposite problem. A system with bundles, packages, filesystems, and files installed that might not be needed. How to find such items?

One way is to run a local filesystem directory listing with `ls -laR` and compare against a fresh install. This can be time consuming unless you use some tricks like dismissing subdirectories with equivalent summary lines – if the number of files and the space taken is the same, the contents are probably the same. Scripts to do an enhanced `diff` will help with this task. AIX in version 4.3 does have the ability to remove bundles, packages, and filesets with installp, which did not exist in earlier versions of AIX. We are lucky in that the disks supporting this system for its long life have been small thus the temptation to install more than necessary or be lazy and install more than needed was never, till now, possible. Another way to accomplish this review is to get a `lslpp` installation and modification history via the `lslpp -h` command, then review each of the module histories. Most of the filesets will not have a security aspect with their removal like the unused language environments, but some will like the old X11 compatibility sets for X11R4. As an example we think the fileset
X11.samples.apps.demos is a potential security problem and is not needed on this system.
First see if the fileset is loaded

```
lslpp -h X11.samples.apps.demos
  Fileset         Level     Action       Status       Date         Time
  ----------------------------------------------------------------------------
```

```
     Path: /usr/lib/objrepos
       X11.samples.apps.demos
                   4.3.3.0   COMMIT       COMPLETE     11/28/00     03:05:54
                  4.3.3.10   COMMIT       COMPLETE     02/11/01     11:11:25
```

Now check if this fileset has fileset dependencies

```
    Command_to_Execute follows below:
>> /usr/lib/instl/sm_inst installp_cmd -u -f'X11.samples.apps.demos' '-p'


    Output from Command_to_Execute follows below:

---- start ----

installp -up -f File 2>&1

File:
    X11.samples.apps.demos

********************************************************************************
installp PREVIEW:  deinstall operation will not actually occur.
********************************************************************************

+-----------------------------------------------------------------------------+
                    Pre-deinstall Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-deinstall verification
  and will be removed.

  Selected Filesets
  -----------------
  X11.samples.apps.demos 4.3.3.0              # AIXwindows Sample X Consorti...

  << End of Success Section >>



FILESET STATISTICS
------------------
    1  Selected to be deinstalled, of which:
        1  Passed pre-deinstall verification
  ----
    1  Total to be deinstalled


********************************************************************************
End of installp PREVIEW.  No deinstall operation has actually occurred.
********************************************************************************
```

This tells us there are no filesets dependent on the selected fileset so we can remove it from the system

```
    Command_to_Execute follows below:
>> /usr/lib/instl/sm_inst installp_cmd -u -f'X11.samples.apps.demos'    '-V2'


    Output from Command_to_Execute follows below:

---- start ----
```

```
installp -u -V2 -f File 2>&1

File:
    X11.samples.apps.demos

+-----------------------------------------------------------------------------+
                    Pre-deinstall Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...

SUCCESSES
---------
  Filesets listed in this section passed pre-deinstall verification
  and will be removed.
    -- Filesets are listed in the order in which they will be removed.
    -- The reason for deinstalling each fileset is indicated with a keyword
       in parentheses and explained by a "Success Key" following this list.

  X11.samples.apps.demos 4.3.3.0 (Selected)
    AIXwindows Sample X Consortium Demos Source


  Success Key:
  Selected  -- Explicitly selected by user to be deinstalled.
  Dependent -- Dependent of other filesets being deinstalled; dependents
               are always deinstalled when "auto-deinstall" (-g flag) is
               specified.

  << End of Success Section >>

FILESET STATISTICS
------------------
    1  Selected to be deinstalled, of which:
        1  Passed pre-deinstall verification
  ----
    1  Total to be deinstalled

+-----------------------------------------------------------------------------+
                          Deinstalling Software...
+-----------------------------------------------------------------------------+

installp:  DEINSTALLING software for:
        X11.samples.apps.demos 4.3.3.0

Finished processing all filesets.  (Total time:  13 secs).

+-----------------------------------------------------------------------------+
                                Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
X11.samples.apps.demos      4.3.3.0         USR         DEINSTALL   SUCCESS
```

Recall we do this from SMIT where the path is SMIT -> `Software Installation and Maintenance -> Software Maintenance and Utilities -> Remove Installed Software`

Which gives this screen:

```
                Commit Applied Software Updates (Remove Saved Files)

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

                                                        [Entry Fields]
* SOFTWARE name                                        [all]                   +
  PREVIEW only? (commit operation will NOT occur)       no                     +
  COMMIT requisites?                                    yes                    +
  EXTEND file systems if space needed?                  yes                    +
  DETAILED output?                                      no                     +

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

At the SOFTWARE name filed use the PF4 key to get a list.

SCSI Disk Configuration

AIX can use mirrored disks to make the rootvg. This gives the ability to loose a system disk and continue to run the system if quorum is adjusted or
turned off. We elect to use another configuration as mirroring has drawbacks like the system having to keep the mirror consistent, the write penalty since all the mirrors have to be written, corruption or inadvertent removal of files is done on all the mirror members, etc. We setup the critical systems to have a like disk making the rootvg as an external attached disk in no volume group. A cron job does a `dd` copy of the rootvg physical disk every day at the most idle time. Then if the system disk fails a quick reboot restores the system as it was at the time of the `dd` copy. This method has drawbacks also like having to recover from the checkpoint/snapshot some hours ago, a need for a reboot, the possible need to readdress the SCSI ID of the disk, inability to have both disks up concurrently due to the volume group structure, etc. BUT it is immune to corruption, accidental file removal, etc. With such a rootvg disk configuration it makes the ability to do a sanity check for exploits like trojan and root kits. This is possible since the disk is normally offline except for the `dd` copy to the raw device. The ideal setup would have the running system disk in the external enclosure with the copy in the internal peripheral bay. Then we could disable the failed external, boot the internal `dd` copy, then work on the failed disk in the external enclosure while the system is running. AIX can add and remove SCSI devices on a running system. The `cfgmgr` command will scan the system for devices added since the last bootup or `cfgmgr` command. The `rmdev` command removes SCSI devices if they are not holding active volumes. eg `rmdev -l hdisk0 -d.`

rootvg Configuration

AIX has a Logical Volume Manager (LVM) to manage disks, volumes, partitions, filesystems, and volume groups. A quick overview: Disks are divided into partitions. A logical grouping of partions makes a logical volume which typically holds a filesystem. Filesystems can grow on demand as can volumes. A group of volume is, of course, a volume group. The volume group with the boot logical volume, the volume holding /

(hd4), /usr (hd2), /var (hd9var), /tmp (hd3) , and perhaps others is called the rootvg. For more information consult the AIX documentation set.

Larger disks on the replacement system allows logical volumes and filesystems like /tmp and /var to be increased to prevent their filling as a denial of service attack. Larger memory requires that the paging space be increased or evaluated for an increase. If the system has enough memory to not page, no need for a page file. Normal rule of thumb is 1.5 times real memory for the paging space. Likewise the sysdump volume needs to be big enough to hold a memory image.

Note: From here on in the Step by Step Guide section, the configuration changes are what would be done with a new system install. The only difference is
instead of "check to ensure that"..... it would be "change to ensure that".....

sendmail

As mentioned in the system description, a smtp email gateway to Microsoft Exchange is provided by this machine. AIX 4.3.3 provides sendmail V 8.9.3 so we have a fairly recent sendmail version that is usually regarded as stable. IBM has addressed recent vulnerabilities and the sendmail fileset usually gets updated with a recommended maintenance level. The fileset for sendmail is `bos.net.tcp.client;` and for the sendmail m4 macros and such `bos.net.tcp.adt.`

An unsettled debate as to sendmail's security stance against other MTAs is ongoing. We feel comfortable with the risks running sendmail as the MTA. It is vendor maintained, though as shipped it is configured to be an open relay. We do a few things with the supplied .mc file to enhance security, close the open relay and use a mailer table. The modified `aix433.mc` file is given in Appendix A. We then run sendmail under SRC with this command and qpi defined as 5 minutes.

```
/usr/lib/sendmail "$src_running" "-bd -q${qpi}"
```

Why 5 minutes? The other AIX nodes have sendmail configured with the same `/etc/sendmail.cf` file but do not run sendmail under SRC. Instead the systems use the local mailer for email intra-system and send all inter-system smtp mail to the mail gateway on this machine. The other machines run sendmail as a process under cron with a cron entry similar to:

```
25 2,4,6,8,10,12,14,16,18,20,22  * * * (startsrc -s sendmail -a "-bd -q1m"; sleep 300; stopsrc -s sendmail)
> /dev/null 2>&1
```

This cleans out the mail queue and allows receipt of mail queued for the local machine on the mail gateway. This means the sendmail is running as a daemon for 5 minutes every other hour, which is our balance between security and utility. Thus in addition to being a gateway, it is a smart relay. Remember there may be inter-system mail, system email to root, cron errors when stderr is not redirected, applications like to send mail to the user, and on bootup the system sends email to users with editor journal files.

This system has an alias (mailhost) in DNS and NIS so we can move the smtp gateway to another system by changing the alias.

DNS/BIND

Securing BIND/DNS starts with blocking UDP and TCP inbound packets to port 53 with border router ingress and egress filtering and blocking of source routed packets. On the DNS servers syslog is turned up for the named daemon.

For the /etc/named.conf file we restrict zone transfers with the allow-transfer statement, and prevent glue fetching with the fetch-glue no; statement. Dynamic updates are off by default. A portion of /etc/named.conf:

```
options {
        allow-transfer {aa.bb.c.dd};
        fetch-glue no;
};
```

NIS

Not much we can do to secure NIS. Make plans to replace it for authentication, we do have plans as we transition the NT backend to Windows 2000.
We do choose our NIS domain name carefully, keep current with patches and monitor security vulnerabilities, use /var/yp/securenets, restrict access to the NIS master node, and do not allow the use of ypsetme. The above mentioned border router ingres/egress filtering, tcpwrappers, and vigilance are our efforts till we can get NIS replaced.

Since we are using NIS, we do use a lot of aspects of NIS. Automount maps are served by NIS as are the services list, the ethers file, the networks file, etc.
We also use netgroups to limit access to resources. An example is a netgroup admins in the `/etc/passwd` file to restrict access to this machine:

```
root:!:0:0::/:/bin/ksh
<SNIP>
+@admins::0:0:::
+::0:0::/nologin:/bin/nologin
```

This allows anyone in the netgroup admins in with their UID, GID, $HOME, shell, etc. specified in the admins netgroup NIS map, but others fall through to the next line which prevents logins since the specified shell is not in `/etc/shells`. As the name indicates, the admins NIS netgroup contains the usernames of the UNIX administrators.

AIX also supports NIS+ so if all your systems also support NIS+, that option can be considered. Be aware that this is AIX's first attempt at NIS+ and its use on AIX is not wide spread.

What facilities use NIS and the other options like local files is given in the /etc/netsvc.conv file. Be sure to edit this file to reflect your security policy. The options for the AIX implementation are extensive allowing different options for IPv4 and IPv6. We edit ours to be similar to:

```
hosts=local,nis,bind
```

This means hosts lookup is to the local /etc/hosts file, then NIS, then DNS.

NFS

Once the patch area is completely moved to the SP CWS (Control Work Station) we will turn off the NFS server on this machine. In the interim we only export the patch area when and while we roll around the patches.

It is an interesting effort to trim the number of NFS daemons down with SMIT or /usr/sbin/chnfs. You might think that specifying the number of nfsd daemons to run as 0 would have the desired affect. SMIT checks and does not allow 0 as a value. Removing /etc/exports is the correct way to turn off nfsd. Processes needed by an NFS server include rcp.mountd, nfsd, rpc.lockd, and rpc.statd. Processes needed by a NFS client include biod, rpc.lockd, and rpc.statd.

AIX has the nfso command that sets attributes of the NFS subsystem. We turn on the following:

```
         portcheck= 1
portcheck Checks whether an NFS request originated from a privileged port. The
default value of 0 disables port checking by the NFS server. A value of 1
directs the NFS server to do port checking on the incoming NFS requests.
         udpchecksum= 1
udpchecksum Performs the checksum of NFS UDP packets. The default value of 1
directs the NFS server or client to build UDP checksums for the packets that it
sends to the NFS clients or servers. A value of 0 disables the checksum on UDP
packets from the NFS server or client. Use udpchecksum to check data integrity.
         nfs_use_reserved_ports= 1
nfs_use_reserve_ports Forces the client to use reserved ports for all
communication. The default is not to force that use. A value of 1 turns the
```

```
nfs_use_reserve_ports option on, while a value of 0 turns it off.
```

To use nfso to change a parameter

```
nfso -o portcheck=1
```

To use nfso to show all the parameters

```
nfso -a
```

If the system does need to export a filesystem, limit the export to just the nodes needing the filesystem or use NIS netgroups. As mentioned elsewhere in this paper it is best to do this in the system supplied /etc/rc.nfs to get the parameter set just before the nfs daemons start. An upgrade may replace or modify /etc/rc.nfs and other IBM supplied rc files so be sure to check these after upgrades, PTFs, etc.

We have most of our data on Network Appliance NFS servers. This gives us the ability to NFS serve at very high speed, share data to both NFS and CIFS (Common Internet File System), do both UNIX and NT security models, and enforce quotas. User's $HOME directories are quota limited to a very small amount and we have a project structure for almost all of the diskspace on the NetApps. What does this have to do with security? We feel that not having any company data in user's $HOME and catalogued in a structure corresponding to a project model makes permission management a manageable task. This also limits the users from scanning the NFS space looking for data items.

Limit services, daemons, information leaks, etc.

Each `rc` file in `/etc` should be read and reviewed to remove any service, daemon or information leak that is possible without disabling a needed service, daemon or information item.

Review /etc/inittab to limit services, daemons, and information leaks to those necessary for the system.

Review /etc/inetd.conf to limit services to those necessary to the running of the system and applications. We remove the small services from inetd.conf, the finger service, uucp, bootp, rusers, etc. For security reasons we will not list all the services we do leave and your security policy will what you leave enabled..

Do be aware that patches, updates, upgrades, and similar can edit or replace these files so be sure to review them after such activity and on a periodic basis. A frequent network and host based vulnerability scanner run also helps keep the unneeded services, daemons and information leaks from inadvertent reactivation.

You can add information gathering to some of the daemons and services in /etc/inetd.conf like adding a `-l` to the `ftpd` line to log more information.

In our case the remaining lines in /etc/inetd.conf

```
# grep -v ^# /etc/inetd.conf
ftp     stream  tcp      nowait  root    /usr/local/bin/tcpd    ftpd -l -u 077
telnet  stream  tcp      nowait  root    /usr/local/bin/tcpd    telnetd -a
shell   stream  tcp      nowait  root    /usr/local/bin/tcpd    rshd
login   stream  tcp      nowait  root    /usr/local/bin/tcpd    rlogind
bpcd    stream  tcp     nowait  root    /usr/openv/netbackup/bin/bpcd bpcd
kshell  stream  tcp     nowait  root    /usr/sbin/krshd krshd
klogin  stream  tcp     nowait  root    /usr/sbin/krlogind krlogind
```

NOTE: When removing items from /etc/inittab the comment character is the colon NOT the pound sign!!
When you attempt to remove an item from inittab with a pound sign comment, you have instead enabled a item with a pound sign as the first character

Remove /etc/hosts.equiv and monitor to ensure it does not reappear. The file would contain hosts trusted by this system for the r<commands> like rsh. This is a system level trust not a user level trust.

Public Domain Security Tools and Utilities

We install tcpwrappers V7.6, lsof v 3.68, OpenSSH V3.1p1. We also install other public domain applications but those listed have security aspects.

Several issues may arise with installing public domain applications. Some sites have policy statements against non-commercial applications for support reasons. Some commercial companies exist for that reason. We use PowerBroker instead of sudo as an example of that aspect.

Some sites do not have compilers to build the applications since IBM does not supply a c compiler as part of AIX now. There are several sites that have the major public domain applications available as binaries for those without compilers, but this method has the drawback of not reviewing the source and the make process to make sure no trojan code is injected or other malware is enabled. Monitoring the system for behavior patterns helps to counter this issue somewhat since it takes a lot of experience to find a well-injected malware component. Using one of these sites to install such public domain applications does have the advantage of using `installp` to install the package so `lslpp -h` can show the installation history. The BULL site also has methods of building your own installation package so the ideal would be to get the source from the site that actually distributes the public domain application, review the source, configure, build, build the installation image, then install the application.

Compilers can be a liability as well. Intruders can and do bring their applications with then in source code form, compiling on the target machine to fit the environment.

In the case of tcpwrappers and OpenSSH there are enough changes we make to require a build at our site.

Tcpwrappers by default logs to the mail facility in syslog. Configure /etc/hosts.allow and /etc/hosts.deny to match your security policy. As seen from the above /etc/inetd.conf listing, we use the tcpwrapper method of changing the inetd instead of replacing the wrapped applicaiton. We think this gives us more control and is easier to maintain. Our policy is a subnet can do everything or can not do anything so the /etc/hosts.allow is similar to:

```
ALL : <IP Subnet> : ALLOW
```

while our /etc/hosts.deny is:

```
ALL : ALL
```

OpenSSH is changed to use tcpwrappers, Xauth, and the random number device/application.

For OpenSSH we need to move the host keys to the new machine. For OpenSSH configure the client (ssh_config) and server (sshd_config) files to reflect your security policy. Ours are listed in Appendix A.

Building the application packages is usually site specific enough to be left as an exercise for the reader.

rc Files

AIX uses the rc files startup method. As shipped there is no `rc.local` nor a `rc.local` entry in `inittab`. Thus you will need to create the `/etc/rc.local` file, protect it via permissions, make it executable, then make an entry for in the `/etc/inittab` file. An example of the `inittab` entry:

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

The rc files need to be rwx for root, no access for world, we leave r-x for group since it is system as a just in case back door.

```
ls -l /etc/rc.local
 -rwxr-x---   1 root     system       268 Sep 12 2001  /etc/rc.local
```

Our rc.local file sends email at reboots, starts OpenSSH daemons, etc. Most of the site specific changes to the rc files should be done to the rc.local instance as the ones supplied by IBM are subject to change by updates, upgrades, PTFs, etc. Exceptions would be changes to the IP parameters with the `no` command and to the NFS parameters with the `nfso` command which should be part of the `rc.tcpip` and `rc.net` files to provide as small a window to the setting of these parameters as possible. The integrity checker should catch changes to these files with upgrades, etc.

vi Wrapper

As this is an administration machine with several administrators and several infrastructure configuration files, the potential exists for more than one administrator to edit one of these configuration files and place it into production thus clobbering the work of the other. We wrap the vi binary in a script that creates a lock file and notifies the user of the lock status of the vi edit. The changed name for the vi executable and the vi wrapper are inserted into the TCB.
The vi wrapper is in Appendix A. Administrators use vi to make configuration changes by policy.

Accounting and sar

Accounting is not turned on as part of AIX as shipped. Accounting and associated utilities like sar (System Activity Recorder) can find system anomalies that could indicate malware or other security problems. We run accounting under the adm account instead of root. Most of the commands to enable accounting are provided but commented out.

To enable accounting:

As root invoke /usr/lib/acct/nulladm wtmp pacct to zero the data files and set permissions
Edit /usr/lib/acct/holidays to reflect the site's holiday schedule
add a line to /etc/rc to start accounting early in the bootup process

```
#/usr/bin/su - adm -c /usr/lib/acct/startup
```

Create subdirectories to hold the `runacct` results, owned by `adm` `/usr/adm/acct/nite` `/usr/adm/acct/fiscal` `/usr/adm/sum`

Edit the adm crontab file to be similar to:

```
0 8-17 * * 1-5 /usr/lib/sa/sa1 1200 3 &
0 * * * 0,6 /usr/lib/sa/sa1 &
0 18-7 * * 1-5 /usr/lib/sa/sa1 &
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 3600 -ubcwyaqvm &
5 * * * * /usr/lib/acct/ckpacct &
0 4 * * * /usr/lib/acct/runacct 2>/usr/adm/acct/nite/accterr &
0 7 1 * * /usr/lib/acct/monacct &
```

We also modify the `runacct` script to not `nulladm` the `wtmp` file so that the last command is not reset every time runnacct runs.
To do so just comment the line from /usr/lib/acct/runacct

```
#nulladm ${_wtmp}
```

Root kits usually deal with accounting as well so this is not the perfect tool for detecting security problems but with enough tools turned on the chances of detects is higher.

syslog

We use a central server for syslog entries as well as the local machine. As mentioned in a later section we also have a console log server. The syslog service is part of the `bos.net.tcp.client` fileset. The syslog service has facilities -- the predefined portion of the system using the syslog service; levels -- the predefined relative severity of the message; and destination – user defined result of the trigger of the facility and level. The service is provided by the syslogd daemon process started and controlled by SRC in `/etc/rc.tcpip`. The configuration file to specify which facility, level, and destination is given in the /etc/syslog.conf file.

An example of `/etc/syslog.conf`

```
*.debug;daemon.info;mail.none                    /dev/console
*.debug;daemon.info;mail.none                    /usr/adm/messages
*.debug;daemon.info;mail.none                    @hostlog

local4.warning                                     /dev/console
local4.warning                                     /usr/adm/l4.log
local4.warning                                     @hostlog

*.alert;kern.err;daemon.err;user.none            operator
*.alert;user.none                                root

*.emerg;user.none                                *
```

The local4 facility is what we use for tcpwrappers. The syslog daemon can be turned all the way up by specifying the facility and level pair as *.debug. We use this during problems and suspected attacks. Just add the line to the /etc/syslog.conf, the refresh the syslogd service with the `refresh -s syslogd` command. Be sure to revert after the information gathering to save syslog space and running the daemon at the elevated level

User Maintenance and Configuration

AIX provides some facilities to enhance user maintenance and configuration. We will list some here, but there are more. Be aware that some of these enhanced user security features are possible in a heterogeneous environment, but only on the local system of course. Most are defined as a `default` template and can be then specified for specific users not meeting that default template. The default is usually at the beginning of files like /etc/security/limits so these values are applied unless over ridden by user specific stanzas later in the file.

User and Group check

The `/usr/bin/usrck`, `/usr/bin/grpchk`, and `/usr/bin/pwdck` (part of the `bos.rte.security` fileset) checks for consistency and common configuration errors in the files associated with user, group, and associated files defining users and groups. It is good security practice to run these utilities on a periodic basis and investigate any issues the utilities find. You would usually run these commands with the `-n` switch to report but not fix problems. Then run with the `-t` switch to report errors and ask if they should be fixed. With NIS a lot of anomalies will be detected. A check for users with UID 0 should be done more often. As an example:

`/usr/bin/usrck -n ALL`

which checks in report mode all users.

User configuration files

AIX uses files to limit user's consumption of resources by stanzas in `/etc/security/limits`. Other files to specify users are also in the `/etc/security` subdirectory.
These files specify the user's environment (`/etc/security/environ`), the user's attributes (`/etc/security/usr`), user's roles (`/etc/security/roles, /etc/security/user.roles`) and similar files. Prototype files to be invoked as part of user logins like `/etc/profile` are changed to implement site security policy. Other prototype files that are copied to the users $HOME are created to also reflect site security policy. An example is `/etc/security/.profile` which is the user's `$HOME/.profile`. The user is defined by`: /usr/lib/security/mkuser.default, /etc/security/user, /etc/security/limits, /etc/security/login.cfg, /usr/lib/security/mkuser.sys.`
A recommended change is the default group in `/usr/lib/security/mkuser.default` to a group added at your site.

Note: Be careful when editing these files. Format is important and can cause problems. Use SMIT to make changes if possible.

Heralds and banners

The `/etc/security/login.cfg` has the ability to put out a herald for telnet and other access. Computer lore specifies the herald should not specify welcome and should specify the aspects of the site security policy the user should know before attempting access and authentication. If you run ftp then the appropriate heralds should be modified to meet site security policy. Some sites modify `/etc/motd` to have a security tip or a security warning. We have opted to having the user check the two lines just after the `motd` display giving the time and source of the user's last login and last unsuccessful login.

User restrictions

The files that restrict access based on username like `/etc/ftpusers, /etc/cron.deny, /etc/cron.allow, /etc/at.deny, /etc/at.deny` should be changed to implement site security policy. Usual practice is to add usernames like `root, sys, bin`, etc. to `/etc/ftpusers`. Against logic the ftpusers file specifies users who are not permitted to use ftp – but this is the AIX supplied ftpd so if you supply another ftp daemon it should be modified accordingly. Our /etc/ftpusers:

```
root
daemon
bin
sys
adm
uucp
guest
nobody
lpd
nuucp
sbnet
```

Remove users

Do not forget to remove users who leave the company. Remember to remove the users that have left from all instances including each machine, each machine type, database logins, etc. We script this remove to keep a record of the user's assigned UID, GID, $HOME, and other information along with who did the remove, on which machines, on what applications like databases, and when.
Also do not forget to remove users supplied by AIX that are not needed like uucp and guest.

Special considerations for root username

Root should not have . in its $PATH. Reason being a commonly used command name on a script in the current working directory (CWD) will be executed instead of that commonly used command if . (dot – representing the CWD) is in the $PATH environmental variable before the actual path. Check often by `echo $PATH` and looking for dot (.).

Root should not have a /.rhost file. If there was a /.rhost than someone can spoof the IP address or name of an entry in that /.rhost and gain root access.

Root's .profile and .kshrc should be minimal. The root user should only be used to do privileged tasks so an enhanced environment should not be necessary. Also this encourages administrators to use their own account/username for most of their work, using su to do tasks only root can do then returning to their username.

The root user should not be able to login from the network. This is accomplished by setting `rlogin = false` for the root stanza in `/etc/security/users`.

If possible the root user should not have a shell history. (`export HISTFILE=/dev/null` in /.profile)

Usernames and account settings

We use usernames that correspond to the NT username. These usernames are built with an algorithm based on the human name. Usernames can be deduced from an employee list, but they cannot be used to generate the next username in a sequence if one username is found if the usernames were serialized.

The list of parameters for each username in AIX is extensive, a list in given in Appendix A. Use the appropriate set of parameters to implement your security policy. Items to consider: password policy settings, can another user su to this user, secondary authentication, lock account for failed logins, etc. While in SMIT in the Change User screen, at each field quite extensive help is available with the F1 key.

Filesystem Maintenance and Configuration

core Files

Most users do not need core dumps from applications. If the vendor or the applications support groups need a core file to work an application problem the core dump facility can be turned on or the application run on a system with core dump facility enabled. Core files can and often do contain text that might contain passwords, other user input, or other information giving more information than necessary. Limit core file sizes to 0 with the stanzas in `/etc/security/limits`.

```
core      - soft core file size in blocks
core_hard - hard core file size in blocks
```

suid/sgid Files

If possible the NFS mounts of external filesystems to this system should have suid ability turned off. This is done with the `nosuid` option to the mount command. A list of suid files should be obtained with a command similar to:

```
find /<filesystem> -xdev -perm -4000 -exec ls -alp {} \;
```

Change the 4000 to 2000 to find SGID files.

Once this list is generated for each local filesystem, check each entry to make sure suid is a needed attribute. Save this list and run the command again at periodic intervals to find any changes. A similar scan is part of our integrity checker.

Unowned Files

A file created by an authorized user will be owned by that user's UID. When that user leaves and that account and UID are removed the `ls -l` listing will show the numeric UID since the entry in the /etc/passwd file is removed. These are normal. What is not normal is users restoring files from an archive like tar, users getting UID assigned when files are created via commands like ftp, and the files created by remote root being mapped to the UID nobody. A scan for such files with a command similar to:

```
find /<filesystem> -xdev -fstype jfs -nouser -exec ls -alp {} \;
```

Then review the output and take action as necessary

World Writeable Directories and Files

Having permissions to allow world writeable files and directories should be the rare exception. Filesystems like /tmp and /usr/tmp are examples of those exceptions but those should have the sticky bit set such that only the user who created the file can remove it.

```
ls -ld /tmp
drwxrwxrwt  25 bin      bin         1024 Mar 22 12:19 /tmp
```

Normal permissions for system directories are rwx-r-xr-x

umask

Typically set as part of /etc/profile or /etc/security/user. This should be changed to reflect the site security policy. We use a umask of 027. This value used since most of our users are department oriented along the lines of our UNIX group file. This gives newly created files the permissions of rwxr-x---. For ftp set the umask via the –u <umask> to the ftpd stanza in /etc/inetd.conf. (See the /etc/inetd.conf listing previous) Users can specify the unmask on the ftp command line.

ACLs

AIX provides support for ACLs (Access Control Lists) on files. This gives finer granularity that standard UNIX permissions on files, devices, and directories. Thus ACLs can enhance the security of files/devices that require more access control than permissions can provide. Be aware that backups need to be able to store and recover ACLs.

The `aclput` command sets ACLs, `acledit` sets/changes ACLs, and `aclget` shows ACLs. An example

```
aclget smit.log
attributes:
base permissions
    owner(user):     rw-
    group(techsup):  r--
    others:  r--
extended permissions
    enabled
    permit   rwx     u:guest
```

Which lets the username guest read, write, and execute smit.log even thought world permissions do not allow that access.

Devices

Devices you might want to alter standard or supplied permissions include the floppy drive, the tape drives, the CD-ROM, the Optical drives, etc. Once the owner and permissions are set as you want, be sure to update the device entries in the TCB. Also ensure you get all instances of the device like /dev/rmt0.1, /dev/rmt0.2, ... /dev/rmt0.7 for /dev/rmt0

```
ls -lap /dev/rmt0
crw-rw----   1 root     system    21,  0 Mar 08 17:36 /dev/rmt0
```

symlinks

Symbolic links can be used to cause an overwrite or other security problems with files. A scan of the filesystems can find problems or issues with symlinks. A find command similar to the above to find symlinks

```
find /<filesystem> -xdev -type l -exec ls -lap {} \;
```

A similar command combined with a status check on each symlink found can find links which are stale, i.e. not pointing to an existing file. An example of this type of scan is in the integrity checker (currently commented out) in Appendix B

Network

IPv6

AIX provides support for the IPv6 IP stack in addition to the IPv4 stack. This means there are two entries in /etc/inetd.conf for telnet and the like. It also means you need to get a version of tcpwrappers that understands IPv6. This is, of course, if you have enabled IPv6 as part of the system build.

no Command

Most of the parameters of the TCP/IP stacks are tuned or modified with the no command. While most of the no command parameters deal with tcp/ip stack tuning, the resources should be given to the network to withstand intrusion attempts. You can go too far with some of the parameters so review the man page for the no command and seek advice from IBM and the AIX newsgroups. This list is a partial list of parameters with security aspects:

```
#Syn flood prevention
/usr/sbin/no -o clean_partial_conns = 1
#ICMP redirects
/usr/sbin/ro -o ipignoreredirects = 1
#disallow source routing,etc.
/usr/sbin/ro -o ipsendredirects = 0
/usr/sbin/ro -o ipsrcrooutesend = 0
/usr/sbin/ro -o ipsourcerouteforward = 0
/usr/sbin/ro -o ip6sourcerouteforward = 0
/usr/sbin/ro -o tcp_pmtu_discover = 0
/usr/sbin/ro -o udp_pmtu_discover = 0
```

This list is from the IBM website and should be checked for problems in your environment before implementing. The MTU path discover options can cause problems.

ntp

The system like the others on the LAN uses ntp to keep the system time of day clock synchronized to the correct time. We have a few ntp servers in a DMZ connected to the Internet. A few ntp servers on the LAN

with access through the firewall to those DMZ ntp servers, then all the others synch against those LAN ntp servers. We configure to broadcast instead of polling for the lower strata clients on the LAN. AIX now supports ntp, they did not in previous AIX versions, The /etc/ntp.conf file:

```
broadcastclient
driftfile /etc/ntp.drift
```

X Windows

As insecure as X Windows is, the `xhost +` command can add even more insecurity. Most users do the `xhost +` and think they can live with other users accidentally throwing displays on their workstation. The convenience of allowing all systems they might visit and need an X display back from enabled with a single command is tempting. What they may not realize is other machines can now read keystrokes from the keyboard and capture screen images. Instead use `xauth`. If your $HOME is NFS served then you do not even have to use `xhost +.`

The best method of using X is to tunnel through OpenSSH. Thus `ssh -X <hostname> <xapplication>`

Ongoing Maintenance

Patches

For AIX V4 IBM defines "Full Maintenance Level" as a combination of these levels: Version, Release, Maintenance, Fix, and Recommended Maintenance.
The `oslevel` command gives the Version Release Maintenance Fix (VRMF) level. In our current machine this results in `4.3.3.0`. To determine the recommended maintenance level use the `instfix` command with the maintenance level as a parameter to the `-ik` (installed and keyword) switch. The most recent recommended maintenance level for AIX 4.3.3 at the time of this writing is 09 so to see if this has been applied on the current machine:

```
instfix -ik 4330-09_AIX_ML
  Not all filesets for 4330-09_AIX_ML were found.
```

which shows the latest recommended maintenance level has been applied. The return from this command can be misleading; this return indicates not all of the filesets that make up that recommended maintenance level were found on the system. This will occur when not all filesets for the AIX 4.3.3 version were installed on the machine thus not all the updates could be applied. To find which filesets are not up to date use the command:

```
instfix -ciqk 4330-09_AIX_ML | grep ":-:"
```

Patches for AIX, called PTFs (Program Temporary Fix), are shown with the `lslpp` command. This is different from the fix level given by the `oslevel` command above. Then there are efixes (emergency fixes) which are usually for security type issues/patches/fixes. These efixes are usually turned into PTFs after some QA from IBM. PTFs can be applied to the system, meaning the patch/fix is on the system. The PTF can also be committed, meaning the ability to
regress or remove the patch/fix is removed thus implying the PTF can also be removed. The `lslpp` command provides information on these operations on PTFs. The `installp` command installs PTFs as well as version, release, maintenance and fix updates. This all seems like a lot to comprehend for a patch process, but it is a lot easier now then in AIX V3 and provides a lot of features like the ability to apply the patch to a running system -- often the executable is NFS mounted so new runs of the executable run the patched version while the old copies continue to run. Most other UNIX platforms have something similar.
So, the method we use to patch AIX. As we are self-maintenance for software, we have to check on available recommended maintenance level availability on a periodic basis. We usually wait a week or so and monitor newsgroups and email lists to see if others find significant problems. Then the maintenance level is applied starting with the workstations progressing to the servers then the critical servers. We monitor newsgroups and email lists for problems and fixes via PTFs as well. We also pay for subscriptions to security alert services, use the vendor's notification methods and monitor CERT, SANS, and similar for security and important subsystem issues/problems. Once the efix or PTF for these are available we apply in the reverse order – critical servers, servers, then workstations. We do commit these security PTFs but make and keep a sysback image of the system to roll back if needed. We usually exhaust any other method before doing a roll back for security PTFs. Our application of a PTF involves a run of our system integrity checker just before the install of the PTF(s)
(this to find any changes between the daily cron run of the integrity checker and the install of the PTF). Then after the PTF(s) we rerun the integrity checker to find all the changes done as a result of the PTF which are compared against the PTF control header. PTFs have co- and pre-requisites so this can get quite involved.

The integrity check run just after the PTF saves change shock at the cron run and informs us of all the changes made to the system to help diagnose problems from the PTF install.

Backups

We do `sysback` backups on a monthly basis. The command for `sysback` is given in the Making the clone section. The sysback tapes are written to a system in the computer room and the sysback tapes are kept in the computer room. Most systems are also part of Veritas NetBackup or Legato NetWorker backups to backup servers. These are done even against workstations with no local data to capture log files, accounting information, the errlog data, local machine patch history, etc. The backup servers are GigE on fiber connected for speed, but also to help with security on the network path the backup data transverses (it is harder to tap or snoop fiber). The backup server's tape vaults are in the secured computer room which is manned 24x7. Tapes are shipped to the offsite storage location in locked cases and the agent for the offsite storage service must pickup those cases from the computer room. The next aspect we are looking to do is encryption of the backups to the tape pool that is shipped offsite. We do not want to do all the work in collecting the company's computer data into one place and make it available to anyone who can steal, borrow, or bribe and has an industry standard tape transport.

Syslog and Console logging

All of our UNIX type systems do multiple syslog levels. The console is used for syslog where we also capture console messages like bootup messages before the operating system is running. These are all then captured by a system that scans for text indicating a problem, correlates all the console output to a single collection point, and stores this data for analysis. A document on the first generation of this system is available here.

Syslog also goes to local files. While it is recognized that an intruder will erase or alter locally stored syslog output if possible, we don't think they can get to the logs stored on the console capture system or the central syslog server. Then if they do alter the local syslog file we can determine what records were altered or removed.

As mentioned we also log to a central syslog server which is GigE fiber connected and hardened to an extra degree. The telecom gear also logs to this central syslog server.

The central syslog server and the console capture system also provide a common timestamp for the records just in case our ntp subsystem allows some clock drift. Both the central syslog server and the console capture system scan for key text strings that could indicate a problem. If such a string is found near real-time notification of support staff is sent and both systems have the capability to take other actions. Some examples of these strings are su to root failed, NFS server not responding, reboots, etc.

The local4 facility is what we use for tcpwrappers. The multiple syslog levels are to console (which is also captured by our console server), the standard path and name for the messages file, and to hostlog which is an alias so we can move the actual machine the syslog messages is sent to. The actual /etc/syslog.conf file is given in the Step by Step section.

Failed Login Monitoring

On a periodic basis monitor the failed logins. The command to do this: `/usr/bin/who -s /etc/security/failedlogin`

This file is in the format of lastlog and needs the who command to process the binary records. The output will be similar to:

```
UNKNOWN_    pts/8       Feb 25 09:04    (host1)
junkijf     dtlogin/_0  Mar 12 14:56
UNKNOWN_    pts/9       Mar 18 19:32    (host32)
```

This log file also needs to be cleaned out which is done with tidysys which is discussed later in this section.

sulog Monitoring

Once a day we get an email of the last day's entries in /var/adm/sulog. There should be no unexpected entries as we get syslog notices of su events via the daemon.info entry. Is is often a benign event when we see a failed su to root, the user could just have hit return after the su command by accident, the user could have forgotten the syntax of the command, or other circumstances not attempting to gain root privilege OR it could be exactly what they were attempting to do. It is hard to tell the difference. A one off occurance is usually tagged as a user mistake.

The sulog is also cleaned out by tidysys.

Vulnerability Scans

We use SARA (Security Auditor's Research Assistant) on a periodic basis against the UNIX and telecom subnets. The runs are compared against the last scan and differences are noted. We have plans to network scan for vulnerabilities with Nessus as well to do more defense in depth. We also plan on adding host based scans with RSA Keon UNIX Platform Security once they add SGI IRIX to the supported platform list.

The SARA scans are done every two weeks with scanning level heavy. The first scan level when starting to use SARA was extreme. We choose to do the scans on Thursday starting at 18:00 local time. This gives us a lazy Friday to address problems, impacts users less (our opinion), gives us an approaching weekend to address problems, and a few hours to monitor the runs before we go home.

Integrity Checker

AIX has the Trusted Computing Base and utilities like `tcbck` and `chtcb` to maintain and query that base. Due to the requirement that the base be built from a fresh install the TCB can only be built at an install or upgrade. If the system has not been built with TCB and you are at the latest AIX version, you can still get TCB built by preservation upgrading to the current version from the current version. Once you have TCB a check of the TCB will show any alterations to the key system files with an entry in the TCB. This is like tripwire with key attributes and checksum of key system files. If you ever get root kitted or gain a trojan you can probably determine the changes with TCB.

When AIX V3.2 came out IBM supplied a utility on a floppy to document the existing V3.1 system before the upgrade. This utility produced a text report that could help in finding problems with the upgrade.

We have taken that upgrade utility and added checks more with items like the TCB check, output from `nfso, no, netstat` and other system utilities all captured in a file that we can then compare the next run of the integrity checker against. The output from some commands needs to be filtered to remove fields that change but don't indicate a problem like packet counts from `netstat -i`. For key system configuration files the contents of the files are saved and compared at each run so you can not only determine that a file like `inetd.conf` has changed, but *what* changed. The output of these integrity check runs are stored on a central fileserver as well as to local files. As this was built on the IBM supplied utility and contains a copyright notice I cannot include that whole script in this paper, but building one of your own is trivial, will be more suited to your environment, and well worth the effort. This type utility documents changes to systems by upgrades, finds changes made by intruders, and gives possible early notice of problems BUT it also helps to find misconfigurations done by administrators. If you think of each change

you do to the system and ask yourself "Is this something significant enough to track changes to?" If so, add the output of the utility that lists characteristics of that aspect of the system to your integrity checker. Examples to consider: `lsfs` to list all filesystems, `lsvg` to list all volume groups, `lsps` to list all paging spaces, etc. The output of the integrity checker is emailed to the system administrator who checks the output and the number of email messages to ensure each machine has made its run and successfully sent that email. The portions we have added to the IBM supplied script is given in Appendix B.

errorlog

AIX has a hardware and software error logging facility. The `errpt` command generates the report of such errors like disk media, SCSI timeouts, and core dumps and the `errclear` command allows old errors to be cleared from the system. Periodic runs of `errpt` can alert you to potential problems before they become real issues. We run an analysis run every day and email the non-zero length results to the system administrator. There can be a lot of noise in such a daily errpt report so errpt utility has the option to remove and/or add events to the report. You will need to determine the events to add and/or remove from your report. We choose to get the full report and only exclude events we are sure we do not want to see.

At one time IBM supplied a `dsense` utility to expand hardware errorlog entries via templates. I can not find the status of that utility now. Although the crontab entry shown here to run an errlog report uses that utility to expand the supplied information, taking the `dsense` filter off the command until you can ask your IBM person for a copy will still provide information valuable for monitoring your system.

The crontab entry to run an errorlog report via `errpt`:

```
#!/bin/ksh
export SENSE_PATH=/Remote/user/dsense_dir/dsdir
day2ck=$(date +%m%d0000%y)
[ "$(/usr/bin/errpt -a -k 7239ac3d,bf06fa0d -dH -s $day2ck)" ] && \
  /usr/bin/errpt -a -k 7239ac3d,bf06fa0d -dH -s $day2ck \
  | /Remote/user/dsense                                 \
  | mail -s "errpt `hostname`" admin@aa
echo `date` > /dev/console
```

Thus setup `dsense`, run a report to see if there will be output, then run again to pipe though `dsense` and mail the result to an administrator at a central machine. The events killed with the `-k` switch relate to core dumps and such and can be added to to suit your needs.

tidysys

This script is available on the Group Bull site. Written by Terry Murray the script keeps logs and the system tidy. It was written as a replacement for skulker which is supplied by IBM for AIX, but commented out of the root crontab. We run both daily since we have some applications that have special requirements for the files in /tmp. In a similar vain we trim all user's mailboxes in the system directories to 30 days every 30 days. The security aspect of this is to keep the log files and directories clean to make finding items of interest in the log files easier and to help survive an attempt to fill the log files for a denial of service or hide entries.

crack

On a periodic basis run a password cracker utility like John the Ripper with a dictionary sized appropriate to the task and your security policy. Take action on those usernames with passwords found according to your security policy. Our policy is not to send a note via email. We usually call and counsel on good passwords, how to change, and why good passwords are important.

Our integrity checker checks for users with null/empty passwords. When found the user is disabled and must contact the helpdesk to have their password reset.

netstat –a and lsof –i

On a periodic basis run the `netstat -a` and `lsof -i` commands to get a list of network ports open. Reconcile this list against the ports that should be open.

Also run `netstat -p tcp` and look for counts out of the ordinary indicating failed connects. A daily run of `netstat -r` is part of the integrity checker to find routes added by ICMP redirects, administrators, or intruders. These changes in route tables allow us to quickly find routing problems.

User Education

Users are typically a large part of the security problem. In addition to the majority of security problems being the result of disgruntled insiders, security problems are also dependent on users being unaware of security issues at all levels at all times. The user who opens an email attachment typically does not do so with harmful intent in mind. With well planned user security awareness not only will the users be less prone to do security mistakes, they will become a large part of the solution to the security problem. Users are the ones on the system that should and could notice permission changes on a file, an indication that they were accessing the system in some manner while they were in fact out of the office, that Joe is logged in while it is known to them Joe is out on vacation, etc. are all things that users have knowledge of that can help in the security effort.

Security policy publication and understanding are key. Resistance to social engineering is an ongoing education process. A security tip of the day/week helps to keep security awareness sharp. Changes to the infrastructure to enhance security should be fully explained to the users.

User Monitoring

Monitoring, controlling and policing users is very site specific. Users can be rocket scientist or convected felons or anything in between. We rely on user education more than we should.

We do scan monthly for users having .rhosts in their $HOME. If found the user is contacted to see if we can provide a better mechanism to accomplish the user's task. Other sites just remove the file or create an empty file owned by root with --------- permissions to prevent its creation by users.

A similar scan for .netrc files is done weekly with stronger wording on the advice to users.

We do not restrict user's access to `cron` or `at`. A few users do need these facilities. We are going to add the listing of all crontabs to the integrity checker.

Our monthly user scan also looks for setting the Input Field Separator (IFS) environmental variable. This started when users came to our site from other locations and having the IFS set caused problems. Setting the IFS is a method of attack. Be sure to check .kshrc and other shell startup scripts as well. Use of IFS for exploits is given in the "Protecting Against Programmed Threats" chapter of Practical Unix & Internet Security book by Garfinkel and Spafford.

root password changes and proxy

Changing the root password is very important as you would expect. We use peer review instead of the AIX supplied password policy settings, our review being stricter. Not only should it be changed frequently (30 – 60 days seems to be the norm), but it should also be changed when an administrator leaves or when an incident is suspected. If you do not have such a strict peer review you might consider adding some of the password policy requirements to the stanza for root in /etc/security/users. Such password policies vary but often involve settings similar to:

maxage=8
maxrepeat=2
minalpha=5
mindiff=3
maxrepeats-3

Where:

maxage
```
 Defines the maximum age (in weeks) for the user's password. When the
 password reaches this age, the system requires it to be changed
 before the user can login again. The value is a decimal integer
 string. If 0 is specified, this feature is disabled.
```
maxrepeat
```
 Defines the maximum number of times a character can be repeated
 within the user's password. The value is a decimal integer string. If
 8 is specified, any number of characters can be repeated.
```
minaplha
```
 Specifies the minimum number of alphabetic characters that must be in
 the user's password. The value is a decimal integer string. If 0 is
 specified, no minimum number of alphabetic characters is required.
```
mindiff
```
 Specifies the minimum number of characters required in the user's new
 password that were not in the old password. The value is a decimal
 integer string. If 0 is specified, no minimum number of different
 characters is required.
```
maxrepeats
```
 Defines the maximum number of times a character can be repeated
 within the user's password. The value is a decimal integer string. If
 8 is specified, any number of characters can be repeated.
```
All this from the SMIT Help screen for each parameter

With the root account you have more involvement than setting general user password policy as it only the administrators that need to be consulted and agree.

Even though there are several people with the root password, we place the new password in a sealed envelope, sign and date that envelope, then place it in a company safe. All the administrators do go to functions together and if we get hit by a bus the group security chief or someone he designates can obtain the current root password from the safe.

Check of Configuration

smtp Gateway

To ensure the vrfy and expn commands are disabled we telnet to the smtp port on the mailhost smtp gateway and verify.

```
ksh$ telnet xxxx 25
Trying...
Connected to xxxx.
Escape character is '^]'.
220 xxxx ESMTP Sendmail AIX4.3/8.9.3/8.9.3; Fri, 8 Mar 2002 16: 15:40 -0900
vrfy
252 Cannot VRFY user; try RCPT to attempt delivery (or try finger)
expn
502 Sorry, we do not allow this operation
debug
500 The command "debug" is not known
wiz
500 The command "wiz" is not known
```

To check that the open relay is closed:

```
ksh$ telnet xxxx 25
Trying...
Connected to xxxx.
Escape character is '^]'.
220 xxxx ESMTP Sendmail AIX4.3/8.9.3/8.9.3; Fri, 8 Mar 2002 16: 20:39 -0900
HELO a.b.com
250 xxxx Hello yyyy [aa.bb.cc.dd], pleased to meet you
MAIL FROM: j@xxxx.com
250 j@xxxx.com... Sender is valid.
RCPT TO: fred@hotmail.com
550 fred@hotmail.com... Relaying denied
```

DNS/BIND

Users tell us real quick if DNS is not working. To check for zone transfer blocking use nslookup ls –d command or dig. Try to telnet to port 53 on the system.

Before such attempts turn up debug level on the named daemon with USR1 signals sent to the named daemon's PID. After checking turn off debugging with a USR2 signal.

The nslookup attempt

```
$ nslookup
Default Server:  xxxx.aa.com
Address:  aa.bb.cc.dd

> ls aacom
[xxxx.aa.com]
*** Can't list domain aa.com:Unspecified error
```

The result in syslog of that attempt

```
Mar  8 17:32:05 xxxx named[4928]: unapproved AXFR from [aa.bb.cc.dd].40752 for aa.com
```

NFS

SARA scans will show machines with NFS daemons running as well as determine filesystems exported by the machine. The SARA report should be checked against the list of machines serving NFS.

The integrity checker list the output from the nfso command to compare against the last run.

tcpwrappers

Part of the tcpwrappers distribution is a utility to check the tcpwrapper configuration. `tcpdck.tcpdchk -v` option gives the rules which we do not want to publish. The utility with no switch will report any problems with the rules, and the utility with the –d switch checks hosts.allow and hosts.deny.
On our replacement system these produce:

```
tcpdchk
tcpdchk –d
```

The `tcpdmatch` utility checks access via tcpwrappers, `tcpdmatch <process/service> <IP addres>`
On our replacement system this produces:

```
./tcpdmatch telnet 200.200.200.200
warning: telnet: no such process name in /etc/inetd.conf
client:   address  200.200.200.200
server:   process  telnet
matched:  /etc/hosts.deny line 1
access:   denied
```

core files

With no limit to core file size in /etc/security/limits force a core file by sending a IOT or QUIT signal to a running process like a `ls –laR | more`

```
default:
          fsize = 2097151
          core = 2048
          cpu = -1
          data = 262144
          rss = 65536
          stack = 65536
$ ls -l core
ls: 0653-341 The file core does not exist.
$ ls -l core
 -rw-r--r--   1 user  group    12547 Mar 13 09:37 core
```

Change the /etc/security/limits entry for core

```
default:
          fsize = 2097151
          core = 0
          cpu = -1
          data = 262144
          rss = 65536
          stack = 65536
$ ls core
ls: 0653-341 The file core does not exist.
$ ls -l core
 -rw-r--r--   1 user  group    0 Mar 13 09:42 core
```

SARA Scan results

```
Results – bbb.aa.com


                        General host information:
```

```
                    Host type: unknown type
                    FTP server
                    NFS server
                    NIS server
                    SSH server
                    X Windows server
                    XDM (X login) server
                    NIS client
                    Subnet aa.bb.cc
                    1 Trusted host(s)
                    Scanning level: heavy
                    Last scan: Fri Feb 8 2:55:15 2002

          Vulnerability information:

                    rpc.statd is enabled and may be vulnerable
                    kerberos authentication may be vulnerable
                    Check login banner for telnet connect
                    DNS may be vulnerable
                    R Series: rlogin could be vulnerable
                    rshd should not be on the Internet
                    exports /inst.images to SARA host (possibly
others)

          Actions:

                    Scan this host
```

In the General host information section are the services found on the scanned machine, just as we would expect from our build and configuration.
In the Vulnerability information section are some potential problems with services. Each of these need to be verified if listed as may be vulnerable, etc.
The check login banner for telnet connection is there because we do not have the "unauthorized" keyword SARA scans for. In the actual HTML formatted report each of the lines has a link to more information.

Appendix A

/usr/samples/tcpip/sendmail/cf/aix433.mc

```
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos43N src/bos/usr/samples/tcpip/sendmail/cf/aix433.mc 1.1
#
# Licensed Materials - Property of IBM
#
# Restricted Materials of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1999
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
divert(0)dnl
OSTYPE(aix433)dnl
FEATURE(accept_unqualified_senders)dnl
DOMAIN(generic)dnl
MAILER(local)dnl
MAILER(smtp)dnl
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,noreceipts')dnl
define(`SMART_HOST', `smtp:[mailhost]')dnl
```

vi Wrapper

```
#!/bin/ksh

VI_PATH="/usr/bin/vi.exe"
LOCK_FILE_PREFIX=/tmp/

if [ $# -lt 1 ]; then
   echo "ERR: Syntax: $0 file [file...]"
   exit 1
fi


function handle_trap {
   echo "WRN: Entered signal trap - aborting"
   if [ -f ${lock_file} ]; then
      rm ${lock_file}
   fi
   exit 5
}

trap handle_trap 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

while [ ! -z "${1}" ]; do
   full_filename=`(cd \`dirname ${1}\`; pwd)`
   full_filename=${full_filename}/`basename ${1}`
   lock_file=`echo $full_filename | sed -e 's/\//\./g'`
   lock_file=${LOCK_FILE_PREFIX}${lock_file}

   if [ -f ${lock_file} ]; then
```

```
      echo "WRN: ${full_filename} is being edited by `cat ${lock_file}`"
      read a?"Hit ENTER to continue..."
   else
      echo "INF: $lock_file is lock file"
      echo "`who -m` LOGGED IN AS `/usr/bin/whoami` AT `date`" > ${lock_file}
      chmod ugo+r ${lock_file}
      read a?"Hit ENTER to continue..."
      $VI_PATH ${1}
      rm ${lock_file}
   fi
   shift
done
```

ssh_config

```
Protocol 2,1

# Be paranoid by default
Host *
        ForwardAgent no
        ForwardX11 no
        FallBackToRsh no
```

sshd_config

```
#       $OpenBSD: sshd_config,v 1.42 2001/09/20 20:57:51 mouring Exp $

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# This is the sshd server system-wide configuration file.  See sshd(8)
# for more information.

Port 22
Protocol 2,1
ListenAddress 0.0.0.0
#ListenAddress ::

# HostKey for protocol version 1
HostKey /usr/local/etc/ssh_host_key
# HostKeys for protocol version 2
HostKey /usr/local/etc/ssh_host_rsa_key
HostKey /usr/local/etc/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 768

# Logging
SyslogFacility AUTH
LogLevel INFO
#obsoletes QuietMode and FascistLogging

# Authentication:

LoginGraceTime 600
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# rhosts authentication should not be used
```

```
RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
PermitEmptyPasswords no

# Uncomment to disable s/key passwords
#ChallengeResponseAuthentication no

# Uncomment to enable PAM keyboard-interactive authentication
# Warning: enabling this may bypass the setting of 'PasswordAuthentication'
#PAMAuthenticationViaKbdInt yes

# To change Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no

# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes

X11Forwarding yes
X11DisplayOffset 10
PrintMotd yes
#PrintLastLog no
KeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net
#ReverseMappingCheck yes

Subsystem       sftp    /usr/libexec/sftp-server
XAuthLocation  /usr/X11/bin/xauth
```

User parameters

```
                                  Add a User

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

[TOP]                                                   [Entry Fields]
* User NAME                                            []
  User ID                                              []                          #
  ADMINISTRATIVE USER?                                  false                      +
  Primary GROUP                                        []                          +
  Group SET                                            []                          +
  ADMINISTRATIVE GROUPS                                []                          +
  ROLES                                                []                          +
  Another user can SU TO USER?                          true                       +
  SU GROUPS                                            [ALL]                       +
  HOME directory                                       []
  Initial PROGRAM                                      []
  User INFORMATION                                     []
```

```
EXPIRATION date (MMDDhhmmyy)                          [0]
Is this user ACCOUNT LOCKED?                           false                  +
User can LOGIN?                                        true                   +
User can LOGIN REMOTELY?                               true                   +
Allowed LOGIN TIMES                                   []
Number of FAILED LOGINS before                        [0]                     #
     user account is locked
Login AUTHENTICATION GRAMMAR                          [compat]
Valid TTYs                                            [ALL]
Days to WARN USER before password expires             [0]                     #
Password CHECK METHODS                                []
Password DICTIONARY FILES                             []
NUMBER OF PASSWORDS before reuse                      [0]                     #
WEEKS before password reuse                           [0]                     #
Weeks between password EXPIRATION and LOCKOUT         [-1]
Password MAX. AGE                                     [0]                     #
Password MIN. AGE                                     [0]                     #
Password MIN. LENGTH                                  [0]                     #
Password MIN. ALPHA characters                        [0]                     #
Password MIN. OTHER characters                        [0]                     #
Password MAX. REPEATED characters                     [8]                     #
Password MIN. DIFFERENT characters                    [0]                     #
Password REGISTRY                                     []
Soft FILE size                                        [2097151]               #
Soft CPU time                                         [-1]
Soft DATA segment                                     [262144]                #
Soft STACK size                                       [65536]                 #
Soft CORE file size                                   [0]                     #
Hard FILE size                                        []                      #
Hard CPU time                                         []
Hard DATA segment                                     []                      #
Hard STACK size                                       []                      #
Hard CORE file size                                   []                      #
File creation UMASK                                   [022]
AUDIT classes                                         []                     +
TRUSTED PATH?                                          nosak                 +
PRIMARY authentication method                         [SYSTEM]
SECONDARY authentication method                       [NONE]
```

The RS/6000 SP

The RS/6000 SP we will use to hold the accumulation of Patches for AIX is IBM's Scalable POWERparallel system. This system is a number of RS/6000s connected by a high-speed switch. Any number of the SP component RS/6000s can be used in processing large problems that can be parallelized. Each of the component system's OS can be loaded and built from the control workstation (CWS). Thus each of the AIX patches must be on the CWS for this function, so to reduce the space required for the patches we are moving all patches to the CWS. The SP (also known as Deep Blue for chess fans) has a Kerberos security realm, a private network for inter-node communication, and runs AIX 4.3.3.

Appendix B

Integrity Checker

This listing is trimmed of the IBM supplied portion, but should give a flavor of what can be done and serves as a starting point for your own checker.
__title__ is just a separation banner macro
This is a bit chopped up, but it gives the kind of things to check for after an upgrade, etc.

```
#!/bin/ksh

__title___ These are all the volume groups defined on this system:

lsvg

__title___ These are all the varied-on volume groups on this system:

lsvg -o

__title___ These are the physical volumes constituting each volme group:
__title___ These are all the defined paging-spaces:

lsps -a | cut -c 1-50,60-80

__title___ These are all the local defined users:

cat /etc/passwd

__title___ These are all the local defined groups:

cat /etc/group

__title___ This the /etc/security/limits file:

cat /etc/security/limits


__title___ This the setuid files


# ************************************************
# get locally mounted filesystems
# ************************************************
localfs()
{

df | grep "^/dev" | grep -v "^/dev/cd" | sort -k7 | awk '{ printf("%s ",$7) }'
return $?

}


LOCALFS=`localfs`

find $LOCALFS -xdev -type f -user root -ls 2>/dev/null |
        awk -v cnt=0 '{
```

```
                os = substr($3,4,1);
                gs = substr($3,7,1);

                if ( c == "s" || gs == "s") {
                        printf("%s %-8s %-8s %3s %2s %5s %s\n",$3,$5,$6,$8,$9,$1
0,$11);
                        cnt++;
                }
        }
        END { printf("\nTotal SUID/SGID Root Programs is %d\n",cnt); }'


__title___ This the TCB results;

tcbck -n ALL

#__title___ This the stale links;

#for link in `find $LOCALFS -type l -print`
#do
# if [ ! -f `ls -l $link | awk '{print $11}'` ]
# then
#  echo $link is stale
#  rm $link
# fi
#done


__title___ These are the tcpip parameters:
[ -f /etc/rc.tcpip ] && {


        lsdev -C -c  if -F "name"  | xargs -n1 ifconfig

#       route -f 2>/dev/null    # flush the routing tables
#       sh /etc/rc.net          # re-configure tcpip with the basic routes
        netstat -r | cut -c1-42         # list out the existing routes
        netstat -i | awk '{ print $1 ,  $2 ,  $3 }'
        /usr/bin/namerslv '-s' '-I' # display nameserver entries
        /usr/bin/hostent '-S'   #display address-mapping entries
#       /usr/bin/inetserv '-s' '-S' '-X' # show inetd.conf via odm
        cat /etc/inetd.conf
#       /etc/lsitab -a
        cat /etc/inittab

        [ -f /etc/resolv.conf ] &&
        lsnamsv -C
}

__title___ These the no parameters

        no -a
        nfso -a

__title___ These the DNS files

cat /etc/resolv.conf
cat /etc/named.boot

__title___ These the key config files
for file in `cat /var/docsys/docsys.files`

do
```

```
 echo $file
 ls -e $file
 cat $file
done


__title___ The qconfig file
cat /etc/qconfig


__title___ Subsystems\' Status

lssrc -a

__title___ This is the LPP installation/update history:

lslpp -h

if false; then                  # false
        __title___ These are all the tty definitions:
        mkdir /tmp/$$tty
        lsdev -C -t tty -F name  |
                xargs -i /bin/sh -c \
                        "echo {}: > /tmp/$$tty/{};
                         lsattr -E -F 'attribute value' -l {} >>/tmp/$$tty/{}"
        (cd /tmp/$$tty; pr * )
        rm -rf /tmp/$$tty

fi                              # false
__title___ These are the configured devices

lsdev -C

__title___ This the lscfg -v output:

lscfg -v


exit 0
```

Integrity Checker Filelist

This is a partial list of our filelist

```
/.rhosts

/etc/aliases
/etc/auto.*
/etc/bootptab
/etc/csh.cshrc
/etc/csh.login
/etc/environment
/etc/ethers
/etc/mailertable
/etc/netgroup
/etc/ntp.conf
/etc/profile
/etc/rc*
/etc/rpc
/etc/shells
/etc/snmp.conf
/etc/ftpusers
```

```
/etc/hosts.allow
/etc/hosts.deny
/etc/hosts.equiv
/etc/inetd.conf
/etc/inittab
/etc/named.boot
/etc/sendmail.cf
/etc/syslog.conf

/var/dns/namedb/hosts.db
```

References

IBM Sales Manual (US) Web Site
[IBM Public Information and Services (Frames)]
http://www2.ibmlink.ibm.com/cgi-bin/master?xh=QIwcIPWHNpGFrx1USenGnF9332&request=usa.salesmanual&parms=&xfr=F&xfr=N

SANS (System Administration, Networking and Security) Institute Web Site
The SANS Institute ~ System Administration, Networking and Security - Computer Security Education and Information Security Training
http://www.sans.org/newlook/home.php

SANS GIAC (Global Information Assurance Certification) Web Site
GIAC: Global Information Assurance Certification - Home Page
http://www.giac.org/

SANS GIAC GCIA (GIAC Certified Intrusion Analyst) Posted Practicals
GIAC: Global Information Assurance Certification - GIAC Certified Intrusion Analysts (GCIA)
http://www.giac.org/GCIA.php

Using VAX/VMS to Augment Security of a Large UNIX Environment
Using VAX/VMS to Augment Security - Unix
http://rr.sans.org/unix/VAX.php

Security Auditor's Research Assistant
SARA - Security Auditor's Research Assistant
http://www-arc.com/sara/

Nessus Remote Security Scanner
Nessus
http://www.nessus.org/

Tripwire
Tripwire.org - Home of the Tripwire Open Source Project
http://www.tripwire.org/

Group Bull Web site (A large library of AIX Public Domain utilities)
Download AIX Software
http://www.bull.de/pub/

Network Appliance
Network Appliance
http://www.netapp.com

Simson Garfinkel and Gene Spafford, Practical UNIX and Internet Security, O'Reilly & Associates, Inc;, 1996

Last Updated: June 24th, 2017

Upcoming Training

SANS London July 2017	**London, United Kingdom**	**Jul 03, 2017 - Jul 08, 2017**	**Live Event**
SANSFIRE 2017	**Washington, DC**	**Jul 22, 2017 - Jul 29, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**