Implementing and Maintaining AIX Security Policies

Andre Derek Protas

Introduction

This paper is meant to serve as an introductory guide to the basic security and server hardening functions present in AIX. Many of the features and functions shown throughout this guide are applicable to AIX 4.3 and above, but are more directed toward AIX 5.2. Since security is and will always remain a major issue in server environments, it is crucial that system administrators have a strong working knowledge of security policy implementation and hardening features. This knowledge can be applied to new systems, or to bring older systems up to date.

All administrators should have a thorough understanding of what is presently installed and running on their system. But, with the wide range of server applications, administration specialization is often necessary. Therefore, it is imperative that at least one primary and one secondary administrator per team maintain a strong working knowledge of security. By staffing administrators with security emphasis, the system will be maintained with the newest updates, programs, and patches that deal with security or server hardening issues.

Keep in mind that security is defined on a server-by-server basis. Administrators should not implement any of these security features without personal research as some may cause software conflicts. Each feature must be fully understood and the system checked to ensure that the server will properly handle the security change. All tests should be made on a *Proof of Concept* box prior to production, as well as making sure all changes have gone through Change Management prior to implementation. Also, a backup of important files with a well-documented backout plan should always be utilized, especially when dealing with larger installs of security features on production servers.

Network security is very sturdy but should not be relied upon to the point of ignoring stand-alone security or server hardening features. **Do not depend only on network security to safeguard the servers**. This is the last line of defense, not the first. Many times networking can be bypassed internally within a company, or externally by accessing one vulnerable machine present on the network and running telnet/rsh to another server. One vulnerable node is very likely to be able to take down an entire network. Network security is very powerful, but should be used as a *supplement*, not a *crutch*.

Although much of the responsibility for security tends to lie with a dedicated security team, each administrator should have a basic knowledge of security practices and exploits for their own production systems. "Ethical Hacking" is strongly recommended on servers prior to production rollout. By performing "attacks" on one's own system, the same information that intruders discover will be revealed to the administrators. If the amount of information available is minimized by filling holes identified with these tools, the intruder will have nearly no information with which to launch an attack and will be "pinging in the dark". Of course, these measures *must* be granted with written authorization via change management and must not affect production servers.

Security, as many people mislabel it, is not a game of cat and mouse. The intruders are very intelligent and deserve respect for their knowledge of security and general computing. They should be respected as equal, not inferior. A security-minded administrator would use the appended list of computer exploit websites created by these people to have a better understanding the resources available to parties who may exercise these vulnerabilities.

“If ignorant both of your enemy and yourself, you are certain to be in peril”
-Sun Tzu

Environment Analysis

There are certain considerations that must be taken into account before any implementation can begin upon an AIX server. What version of operating system exists on the system? Where will the server be located? What programs will be running on this server? These are but three of the many questions that need to be answered to direct the path of a security policy. Depending on the answer to many of these questions, the level of security can be determined. For instance, if a server is simply running audits for other servers and is not a production machine, it would seem to not require very much security. On the other hand, if it is running these audits, it is most likely to have the same subnet as the other servers. Because of the possibility of jumping from node to node, high security precautions must be implemented on this example system to ensure that the production servers are not exploited.

Here are a few sample questions that need to be answered prior to the start of building a security framework.

Is the server in an open environment accessible by more than system administrators? --If this is true, the server may be physically vulnerable to Knoppix Live CD's or other threats.
Is there proprietary information stored on the server dealing with a sensitive material? --Data needs to be secured and encrypted, as well as server security protecting these files from possible copying and removal.
Is the server on the same subnet as production servers? --One node can bring down an entire network.
If a production server, does the client want high security? --If a client knows that the information on the server is sensitive to their business, they may request higher security.
Does the Client interface with the server using their own system? --Client computers should not be trusted to remain secure and should be ready to be physically disconnected if there are signs of vulnerability. Work with the client to help them secure their own systems by advising them of security updates/patches.
Do more than administrators log into the server? --When more than administrators are permitted to log in, a server may be vulnerable to employees with access.
Are there people accessing the server that are not familiar with AIX/UNIX? --Untrained users may accidentally damage a system without knowledge of how to recover.

If the answer yes to more than half of these questions, the higher echelon of security features outlined should be implemented. If the unit does not seem to require high security, implement the standard features and review the enhanced security features and evaluate if they would be useful for certain clients.

Security Policy

A server has many types of security features implemented. There are two main approaches for stopping intruders and exploits: security policy to maintain the integrity of the system and server hardening features to avoid malicious penetrations and exploits. Both are equally important but separate fields. Security policies (like all features of security) must be universally applied to the entire subnet, if not the entire network. By ensuring that all possible nodes are secure, the likelihood of intrusion is substantially decreased.

Hardware Passwords

- *Summary*:
    - Booting from CD is a very powerful tool, especially when diagnosing problems with a system. But, there has been new technology with boot CD's that allow root access to any machine that is booted from it. These CD's commonly run a version of Linux that is loaded on a ram-drive. Many times rootkits (hacking tools) come precompiled.
- *Reference*
    - www.knoppix.net
- *Default Security*
    - None
- *Standard Security*
    - BIOS Password: Configuring the BIOS should only be done by administration. Ensure that boot from CD is disabled.
- *Enhanced Security*
    - Power-On Password: With the threat of Live CD's (Knoppix), systems should not be turned on unless proper authorization is given. These live CD's load into a RAM drive and boot off of that drive. They leave no trace, and immediately grant root access to any of the mounted drives. If a person had physical access to a computer, all it takes is a Live CD in the system when it is booting and that person now has root access no matter what other security features may have been implemented.

Login Control

- *Summary*:
    - Numerous setting are supplied and can be tailored to the desired security level. The following are the most important values to edit:
        - sak_enabled: Setting this value to true will require that a certain key sequence (CTRL+X then CTRL+R) is required prior to opening up secure communication. This is similar to the windows "CTRL+ALT+DEL" sequence necessary to login.
        - -logintimes: Specifies certain login times (throughout the day) that a user can login.
        - -logindisable: Set to the number of times a bad password is received before that login is rejected.
        - -logininterval: Specifies the amount of time given for login disable. For instance, if set to a minute, and logindisable is set to 4. If, within 1 minute, there are 4 bad logins, the account will be disabled.
        - -loginreenable: Time until the user login is reenabled.

        - -logindelay: Great way to combat DoS or brute force attacks. If set to 5, after the first login, the terminal will wait 5 seconds until the next request. If another failed login, the terminal will wait 5*2 seconds. If a third failed login, 5*3 seconds. Each time the user on the other end is having to wait longer and longer, which will tend to frustrate programs such as L0phtcrack or John the Ripper.
- *Location of File*
    - /etc/security/login.conf
- *Reference*:
    - AIX Security Manual (page 17)
- *Default Security*
    - sak_enabled: false
    - logintimes: none
    - logindisable: 0
    - logininterval:0
    - loginreenable: 0
    - logindelay: 0
- *Standard Security*
    - sak_enabled: false
    - logintimes: none
    - logindisable: 4
    - logininterval:60
    - loginreenable: 30
    - logindelay: 5
- *Enhanced Security*
    - sak_enabled: true
    - logintimes: specified values
    - logindisable: 3
    - logininterval: 300
    - loginreenable: 360
    - logindelay: 10

System Resource Control

- *Summary*
    - Poorly written code (or perhaps well written code if there is malicious intent) many times will use a process that drains the CPU and paging resources. By calling this process over and over, the CPU and paging are increased so high that connections start to become lost with other processes, possibly resulting in DoS. System resource control is used to restrict the CPU and paging resources from becoming too heavily burdened. There is a hard limit and a soft limit. The user has control of reducing both limits, but only increasing the soft limit to the hard limit threshold. The hard limit may only be increased by the root user.
- *Location of File*
    - /etc/security/limits.
- *Reference*
    - AIX Security Supplement (page 146).

- *Default Security*
  - The AIX Environment already has decent sized hard limits implemented.
- *Standard Security*
  - Change the system resource limits so that users are only limited to the resources they need, as well as ensuring that services do not exceed their expected limits of resources.

Global .profile Files

- *Summary*
  - A standard .profile file can be used to describe a $PATH other than what is valid. This can be especially dangerous for users with higher privelages. Because of this, system administrators are urged to create protected .profile files for their users. These files should only be writeable by root to protect them from being tampered with.
- *Location of File*
  - /etc/security/.profile (for root)
  - $HOME/.profile (for each user)
- *Reference*
  - AIX Security Manual (page 18)
- *Default Security*
  - User configurable .profile.
- *Standard Security*
  - Global .profile files for users based upon their groups to ensure that $PATH stays valid.
- *Enhanced Security*
  - Enforce automatic logoff in .profile security file. By ensuring automatic logoff, idle sessions will be terminated. Suggestion: 10 minute automatic logoff period.
  - Append: TMOUT = 600; TIMEOUT = 600; export readonly TMOUT TIMOUT.

Password Strengthening

- *Summary*
  - A week password can be broken via brute-force attacked very quickly. The proper precautions are necessary to ensure that all passwords created are strong enough to last through a brute-force attack until the terminal is logged off. The following variables may be configured:
    - dictionlist: Verifies passwords do not include certain strings.
    - maxrepeats: Maximum number of characters that can be repeated in passwords.
    - maxexpired: Maximum number of weeks beyond maxage that an expired password can be changed by the user.
    - maxage: Maximum number of weeks beyond maxage that an expired password can be changed by the user.
    - histsize: Number of password iterations allowed.
    - histexpire: Number of weeks before password can be reused.
- *Location of File*
  - /etc/security/user
- *Reference*
  - AIX Security Manual (page 39)

- *Default Security*
    - dictionlist: NA
    - maxrepeats: 8
    - maxexpired: -1
    - maxage: 0
    - histsize: 0
    - histexpire: 0
- *Standard Security*
    - dictionlist: NA
    - maxrepeats: 4
    - maxexpired: 4
    - maxage: 16
    - histsize: 20
    - histexpire: 26
- *Enhanced Security*
    - dictionlist: /usr/share/dict/words
    - maxrepeats: 2
    - maxexpired: 2
    - maxage: 4
    - histsize: 20
    - histexpire: 52

No null Passwords

- *Summary*
    - Passwords have the ability to be null if the user decides not to use one. This is a very dangerous practice and can easily lead to exploitation. Also, some of the standard users (pre-installed) do not have passwords. These should either be removed (refer to “Unnecessary Accounts”) or should be forced to have an associated password.
- *Location of File*
    - /etc/passwd
- *Reference*
    - Is Your AIX Environment Secure?, Shiv Dutta
- *Default Security*
    - None. The system does not force the use of a password.
- *Standard Security*
    - Use a script to check for null passwords in the password file.
        - **awk -F: '{if ($2 == "") print $1}' /etc/passwd**
- *Enhanced Security*
    - Enforce the previous command in a crontab to be implemented daily.
    - Daily review the logs of the script from the crontab to ensure that there are no null passwords.

Physical Security

- *Summary*
  - Typical physical security measures must be maintained. Servers should always be locked up and users should only have access to the servers they must have access to. Having monstrous server rooms with an insecure door is a deathtrap. Also, magnetic-badge locks must be implemented on every possible access to a server room. There should also be human-monitoring systems on all entrances and the server room itself (either camera or person).
- *Reference*
  - *The Art of Deception*, Kevin Mitnick
- *Standard Security*
  - Badge readers, camera installations, human surveillance.
- *Enhanced Security*
  - Biometrics.
  - www.thumbdrive.com/productinfo.htm

---

wall

- *Summary*
  - The **wall** command can be used to broadcast text to all users that are logged in. This could also possibly used as a social engineering tool for those who are unfamiliar with the wall command.
- *Location of File*
  - /bin/wall
- *Reference*
  - AIX Security Supplement (p 135)
- *Default Security*
  - None. The system does not secure the wall command.
- *Standard Security*
  - Ensure the wall command is only executable by the administration group.

Trusted Computing Base

- *Summary*
  - By using the TCB, the root user is able to designate use of the trusted communication path to ensure that only verified users are using those programs in the TCB. The TCB must be installed during the original configuration (cannot be added after install).
- *Location of File*
  - /etc/security/sysck.cfg
- *Reference*
  - AIX Security Manual (page 3)
  - http://www.unet.univie.ac.at/aix/aixbman/admnconc/tcb.htm

- *Default Security*
    - The TCB will not be installed automatically when installing the AIX operating system.
- *Standard Security*
    - Configure the TCB (with all options) on the initial Base Operating System (BOS) installation.
- *Enhanced Security*
    - Thoroughly audit all of the programs to be put into the TCB. Ensure that sensitive programs are placed here, and any exploitable programs are not. Run Tripwire on the /etc/security/sysck.cfg to ensure the file is not tampered with.

User Control Admin

- *Summary*
    - There should be two administrators (plus root) that should be allowed to change users. By allowing two administrators besides root (redundancy), root access is no longer necessary to change/make users. The less often root is used, the better, as it has full control and can be very detrimental to a machine if used inproperly.
- *Location of File*s
    - /etc/security/user
    - /etc/security/limits
    - /etc/security/audit/config
    - /etc/security/lastlog
- *Reference*
    - AIX Security Manual (page 27)
- *Default Security*
    - Root access is required each time user updating is necessary.
- *Standard Security*
    - Create two administrative users that have control over the users. These administrators should not have access to any other root permissions, just the user control.

Distributed Authority

- *Summary*
    - Many times root access is overused. Root, the all powerful superuser, should only be used in times where multipermission access is necessary. It should not have to be used every time a CD needs to be mounted, or a common script to be run. The more people that use and know the password to root, the higher the possibility of unauthorized activities. Root should be kept as secure as possible by a senior administrator.
    - Separate users should share the power of root. For instance, one user could be the user for making other users. Another user could be the authority when it comes to changing file permissions. By having these jobs segregated, a typical user, if granted access to one of these other accounts for necessity, will not gain root access.
- *Location of File*
    - /etc/security/user

- *Default Security*
  - root access or nothing.
- *Standard Security*
  - Three separate users to break root authority into three sections. One for user control, one for file-system maintenance, and another for other typical commands (such as **mount**). These accounts should be dispersed to different administrators to maintain checks and balances.
- *Enhanced Security*
  - Create more accounts with certain root privileges. The more accounts with root-type permissions for certain aspects of the system, the better. That ensures that not one account has full permissions (unless root is necessary).

Hardening Features

Server hardening is used to prevent intrusion and exploitation. These features are essential, especially for systems that require high network access. Keep in mind what level of security is needed for the server based upon the data and applications running. There is a standard security framework to be implemented on all servers, but higher security needs call for stronger features. If there are any questions or concerns about any of the features, refer to the documented source.

.netrc/.rhosts/hosts.equiv Files

- *Summary*
    - These files are meant to help automate certain functions. These files can offer some "shortcuts", but can also be very large security risks. They should be deleted as they are not encrypted but hold usernames and passwords. If the system requires that these files remain intact, they should be confirmed to have 600 permission.
- *Location of File*
    - $HOME/.netrc
- *Reference*
    - Strengthening AIX Security
- *Default Security*
    - Files are present and unencrypted.
- *Standard Security*
    - Permissions = 600.
    - Continually check for new files of these types (crontab) and ensure they are not being created.
- *Enhanced Security*
    - Delete all .netrc/.rhosts/hosts.equiv files. Automation of functions should not be necessary for higher security machines.

Login Message

- *Summary*
    - Ensure that the login screen to an AIX system does not tip off an intruder to the version or possibly the patches applied. By knowing what system and what patches are installed, an intruder knows what exploits could be used on the system. This will save them time and will concentrate their efforts on documented exploits for that system.
- *Location of File*
    - /etc/security/login.cfg
- *Reference*
    - AIX Security Manual (page 17)
- *Default Security*
    - The default login screen for an AIX machine will show the name of the system along with the AIX version and possibly even the maintenance level.
- *Standard Security*
    - "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\nWelcome to: <server name>\n\rUnauthorized Access is Prohibited\n\rlogin:"
- *Enhanced Security*

    - "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\nUnauthorized Access is Prohibited\r\nlogin:

X11 Interception

- *Summary*
    - The X11 server has the vulnerability of allow keystroke logging via the **xwd** and **xwud** commands. These commands may be necessary for certain programs or accesses to the system. Confirm by reviewing access dates of the commands to ensure that they are not being used.
- *Location of File*
    - xwd and xwud commands.
- *Reference*
    - AIX Security Manual (page 20)
- *Default Security*
    - xwd and xwud are enabled and active.
- *Standard Security*
    - xwd and xwud should be disabled and OpenSSH should be installed to ensure that server interaction is not done over clear-text transfer.

GUI Disabling

- *Summary*
    - Security issues are common when dealing with desktop environments. These environments can be exploited by intruders and can cause numerous threats. Typically the GUI is not utilized in higher security servers and causes unnecessary vulnerabilites.
- *Location of File*
    - /etc/rc.dt
- *Reference*
    - AIX Security Manual (page 20)
- *Default Security*
    - Desktop environments (KDE, GNOME, and CDE) are installed.
- *Standard Security*
    - CDE only.
- *Enhanced Security*
    - No desktop environment.

Direct Root Login

- *Summary*
    - Administrators must **su –** in order to gain root access. This will ensure that the only people accessing / are administrators. This servers two purposes: two layers of passwords to gain root access and logging of which administrator does gain root access (sulog).
- Command
    - /etc/securetty
        - Only entry should be "console"

        - Permissions should be set to 400.
- *Reference*
    - AIX Security Manual (page 21)
- *Default Security*
    - Direct root login is allowed.
- *Standard Security*
    - Disable direct root login for remote.
- *Enhanced Security*
    - Disable direct root login for console (as well as CDE, which should be disable anyway)

Unnecessary Accounts

- *Summary*
    - Be leaving default user accounts, the accounts may be used as a stepping stone for an intruder to gain access to other accounts. These accounts, unless being used, should be disabled.
- Command
    - smit rmuser
- *Reference*
    - AIX Security Manual (page 33)
- *Default Security*
    - uucp and nuucp : The owner of the hidden files for the uucp protocol. This protocol is used to communicate with other UNIX machines other than AIX and has been overtaken by other protocols (ftp, etc).
    - ipd : Owner of files used by printing subsystem.
    - imnadm : IMN search engine used for Documentation Library Search.
    - guest : Allows access to user who do not have access to accounts.
- *Standard Security*
    - guest and imnadm accounts must be disabled.
- *Enhanced Security*
    - Ensure all files that are owned by bin and migrated to be owned by root. The bin accounts should remain, but should not own any files.
    - Have lockout times for non-administrator accounts. By enforcing availability to regular users only to work hours, non-work hour attacks will be minimized to only administrator accounts.
    - Disable all mentioned accounts. Ensure that there are no dependencies upon those owners that may cause problems with the removal of the accounts.

Unnecessary Groups

- *Summary*
    - There are three groups that may not be needed after the standard install of AIX 5.2.
        - uucp : Group to which uucp and nuucp users belong.
        - printq : Group to which the lpd user belongs.
        - imnadm : Group to which imnadm user belongs.
- *Location of File*

    - /etc/group
- *Reference*
    - AIX Security Manual (page 33)
- *Default Security*
    - All three are groups are enabled.
- *Standard Security*
    - If none of the groups are necessary on the server, disable all three accounts.

Unnecessary Services

- *Summary*
    - Many services are automatically configured and run on startup. Most of these servers have specific tasks and do not require all of the preconfigured services to be running. Some of these unnecessary services also hold security risks and should therefore be disabled, especially those which are listening on ports.
- *Location of File*
    - /etc/inetd.conf
    - /etc/inittab
    - /etc/rc.nfs
    - /etc/rc.tcpip
    - /var/adm/inetd.sec
- *Reference*
    - AIX Security Manual (page 219 - Appendix C)
- *Default Security*
    - Common services are installed and typical run when server is up.
- *Standard Security*
    - Ensure that /etc/inetd.conf and /var/adm/inetd.sec are owned by root and have permissions of 600.
    - Refer to the documentation listed from IBM and disable services based upon the needs of the server. Any extra services that are not needed by that server must be disabled.
- *Enhanced Security*
    - Refer to Chapter 8 of the AIX Security Supplement to review tightening up services that must be maintained on the server.

FTP Exploitation

- *Summary*
    - The ftp command must not have the sticky bit for root. If the sticky bit is set, there is a certain command sequence that will force the server to run arbitrary code from another remote server. This code could easily be malicious, and this exploit should be avoided.
- *Location of File*
    - /bin/ftp (check permissions)
    - ~ftp/bin - 555
    - ~ftp/etc - 111
- *Reference*
    - www.insecure.org/sploits_AIX.html (ftp mget vulnerability)

- *Default Security*
    - Setuid bit is set for ftp
- *Standard Security*
    - Setuid bit must be disabled.
    - To ensure other vulnerability security, always make sure that the ftp version present is the highest available (maintenance levels will include updates).
    - For other security concerns, ensure the ftp directory does not have: .rhosts, .forward, .netrc files.

Remote Root

- *Summary*
    - Remote root access is very dangerous, especially with telnet where the password is transferred over clear-text. Although this is a huge security concern, many times remote root access is necessary for server administration.
- *Location of File*
    - /etc/security/user
        - "rlogin = <boolean>" for root user.
- *Reference*
    - AIX Security Supplement (page 159)
- *Default Security*
    - Remote login is activated.
- *Standard Security*
    - Use OpenSSH so that the root password is transferred via encryption, not clear-text.
- *Enhanced Security*
    - Disable remote login. By distributing root authority (as discussed earlier), root access can be avoided by using different administration accounts for different permissions. OpenSSH should still be implemented.

Authentication

- *Summary*
    - Authentication is used primarily to ensure that users and services are who they say they are. Authentication uses encryption as well as ticket management systems to ensure that security is implemented on all services and that users are not "spoofing" other users by using their IP addresses or passwords.
- *Location of File*
    - cas.server and cas.client
- *Reference*
    - AIX Security Manual (page 77)
    - http://web.mit.edu/kerberos/www/dialogue.html
- *Default Security*
    - No Authentication is implemented on install.
- *Standard Security*
    - Use OpenSSH to offer minimal authentication for used of remote login.
- *Enhanced Security*

    - Implement and enforce use of PKI (AIX Security Manual p. 77), Kerberos, PAM, and LDAP. Be careful as the configuration of many system files may be dependent on the previous settings before installation. On new systems, implement authentication in the installation to ensure dependency stability. On running systems, do research into each authentication module suggested and investigate dependency problems that may arise.
    - The use of biometrics/secureID is suggested.

IPSec

- *Summary*
    - IPSec holds many secure network tools such as vpn that should be implemented, especially when remotely logging into the server. These files can be found on the install media. This package is required for all IPv6 connections, and is suggested for IPv4 connections as well.
- Check if IPSec has been installed
    - lslpp -L '*ipsec*'
        - bos.msg.en_US.net.ipsec
        - bos.net.ipsec.keymgt
        - bos.net.ipsec.rte
        - bos.net.ipsec.websm
- *Reference*
    - AIX Security Supplement (page 19)
- *Default Security*
    - IPec may be present, but may not be active after install.
- *Standard Security*
    - Activate IPSec and log all traffic.
- *Enhanced Security*
    - Configure IPSec as a packet filter and define filtering rules. Also, by using smitty installp in the IPSec directory, vpn tunnels can be configured.

SecureTCPIP

- *Summary*
    - For systems that require higher security, many commands should not be allowed execution. For instance, rlogin, rlogind, rcp, rsh, rshd, tftp, tftpd, trpt.
- Command
    - To block execution of these non-secure commands, only one command is necessary: **securetcpip**.
- *Reference*
    - AIX Security Manual (page 121)
- *Default Security*
    - All of the previously mentioned commands are activate and executable.
- *Standard Security*
    - Because some of these commands may be necessary to certain processes or programs, standard security practice would be to disable those that aren't.
- *Enhanced Security*

  - Run securetcpip each time TCP/IP is installed and ensure that these commands are not used.

Port Knocking

- *Summary*
  - Requires a “password” to open a specific port. For instance, port 28 cannot be opened until port 1256, 1257, and 1259 were pinged within 3 seconds. This sequence can change each time it is used to ensure non-predictability.
- *Location of File*
  - There must be a client script to go along with the server script. The client script should be kept secure (encryption/removable media).
- *Reference*
  - www.portknocking.org offers many useful scripts.
- *Default Security*
  - Firewall support from networking services.
- *Enhanced Security*
  - If certain non-secure protocols must be used, keep the ports closed until the passcode has been pinged. This will keep these protocols to be only used by authorized personnel.

System/Data Integrity

Even though a sever may be very secure and hardened, there is always the likelihood of intrusion. This can be either from inside the network or outside. The main goal of an intruder is to usually open holes in the system to enter back in later. These could be Trojan Horses, new user with root UID levels, etc. Integrity systems are used to watch system-critical files and create alerts when they are tampered with. Also, integrity systems can be used to monitor network traffic to the server. There are integrity programs within standard AIX, but many times the commercial or open-source products are more robust.

UID Control

- *Summary*
  - User ID values signify the identification of the user. These must be unique values and non-repeating. Most importantly, there should only be one UID of 0 (root). A common entry point of a hacker is to create a new user with a UID of 0, which opens a backdoor to root access.
- *Location of File*
  - /etc/passwd
- *Reference*
  - *Is Your AIX Environment Secure?*, Shiv Dutta
- *Default Security*
  - Only root has a UID of 0, but this must be maintained.
- *Standard Security*
  - crontab the following command to run daily
    - **awk -F: '{if($3 == 0) print $1}' /etc/passwd**
    - This script will check for other UID that have the value of 0. There should only be one: root.

Audit Control

- *Summary*
  - The audit subsystem can be used by a security administrator to step through potential or actual exploits of security policy. The audits allow for security holes to be revealed. Also, the incredible logging features allow the administrator to track the path of the system in the case of an exploit/breech of security. Some of the logging features supported are: kernal logging, change configuration, change host ID, change route, conection, create socket, export object, etc.
- *Location of File*
  - /etc/security/audit
- *Reference*
  - AIX Security Manual (page 47)
  - IBM System Management Guide: Communication and Networks

- *Default Security*
    - Audit subsystem is not enabled.
- *Standard Security*
    - Use audit command without enabling full subsystem via watch command.
        - Example: watch -eFILE_Open -o /tmp/vi.watch vi /etc/hotsts
        - This command will show all of the events for file opening when vi /etc/hosts is run.
- *Enhanced Security*
    - Implement/launch audit subsystem. More directions/documentation can be found in IBM Security Manual (page 53).

Remote Syslog Exploit

- *Summary*
    - Syslog is used to document common system errors. Syslog is set up remotely in case another system sees a log on its own system with an @hostname for the action field. The system will send the other system a syslog for notification. The problem lies in the fact that no authentication is used for this system, and the format for a remote syslog is well documented (www.insecure.org/AIX_Sploits). This exploit allows for fake syslogs to be created. It may also be used as a DoS attack if enough syslogs were created to cause a memory overrun.
- Location of Problem
    - **netstat -a | grep udp**
        - This command will list commands that are open for the udp user (which is typicaly where the syslog network daemon is present). If the udp user has been disabled, sort through the **netstat -a | pg** list.
- *Reference*
    - www.insecure.org/sploits_AIX.html
- *Default Security*
    - Syslog remote daemon is listening and is vulnerable to syslog spoofs and possibly DoS attacks.
- *Standard Security*
    - Allow syslogs to come in, but put a resource limit on the amount of resource syslogs can take. Also, ensure that no syslogs can trigger a script to be run.
- *Enhanced Security*
    - Disable syslog remote daemon will keep remote syslogs from coming in.

TCPD (aka "TCP Wrapper")

- *Summary*
    - This program enhances logging and access control to network services. It is stable, and the logs are incredibly robust. TCPD does basic auditing of IP address from inetd and will then call upon services. Can be used as pure auditing, or can be used as an intrusion stopperd. This is best utilized with PortSentry.
- *Location of File*
    - /etc/inetd.conf should execute tcpd.

- *Reference*
  - ftp://ftp.purcupine.org/pub/security/index.html
  - www.cert.org
  - *Is Your AIX Environment Secure?*, Shiv Dutta
- *Default Security*
  - None
- *Standard Security*
  - Implement tcpd to do auditing on the TCP connections.
- *Enhanced Security*
  - Implement tcpd in PARANOID mode.
  - Deny all hosts (hosts.deny = all:all) and manually list hosts that are acceptable in the /etc/hosts.allow file.
  - Use tcpd to block intruders from running certain daemons such as telnet, rlogin, and finger.
  - Launch TCPD with PortSentry to allow for dynamic port scan blocking.

PortSentry

- *Summary*
  - PortSentry is a scan detector. When programs such as nmap or SAINT (as described in the programs section) are being run against a server with Port Sentry, PortSentry will automatically add the IP address of the scanning source to the list of denied IP addessses in the TCP Wrapper file (/etc/hosts.deny). Bear in mind that a paranoid program (nMap) may delay pings for hours, which may not be detected by PortSentry.
- *Reference*
  - www.psionic.com/abacus/portsentry
  - AIX Security Supplement (page 103)
- *Default Security*
  - None
- *Standard Security*
  - Log port scans
- *Enhanced Security*
  - Implement tcpd and PortSentry together to block port scanning attempts for produtcion servers that may be vulnerable to internet attacks.

Tripwire

- *Summary*
  - Tripwire constantly monitors files (chosen by the administrator) for validity. Many different methods are used to check the integrity of the files: md5, permission logging, etc. This is very important for securing a system against trojan horses. Many times a user can change a small part of a script or file that is run commonly and turns a benign script into a malicious one. Tripwire (hopefully before the script is run) will notice the change and will notify the administrators. Tripwire may even be configured to launch defensive tactics to check the validity of other files and possibly even shut down vulnerable services before exploits occur.

- *Reference*
    - www.tripwire.org / www.tripwire.com
- *Default Security*
    - The audit subsystem can be used as a similar system, but will not be nearly as robust or powerful in its detection.
- *Standard Security*
    - crontab files
        - To ensure that invalid entries or tampering has not been done to the cron scripts.
    - /etc/passwd
        - Very important file and very vulnerable to exploit. Tripwire would be best for this as it also reviews all permissions (instead of just checksums).
    - /etc/profile
        - Especially if this is a global profile setup. This also contains the $PATH variable which can be exploited to spoof other users.
    - /etc/hosts.equiv and /etc/ftpusers
        - These are very important as they allow remote computers to run ftp commands as well as remote commands – if they are in this list.
    - /etc/security
        - This folder carries with it many of the security policies run on this computer; ensure that it is always valid and clean.
    - /etc/syslog.conf
        - This should be kept under audit to ensure that logging functions are not changed in order to mask future attacks.
    - /etc/publickey and /etc/.rootkey and /etc/keystore
        - These store the public keys for all of the users over a network – ensure this file is never tampered with and only edited using chkey or newkey commands.
    - /etc/shadow
        - This file holds the encrypted passwords and should only accessible as root.
    - Permissions/owners that must be checked/maintained
        - /etc/filesystem : -rw-rw-r-- root system
        - /etc/inittab : -rw------- root system
        - /etc/passwd: -rw-r--r-- root system
        - /etc/group : -rw-r--r-- root system
        - /etc/vfs : -rw-r--r-- root system
        - /etc/security/failedlogin : -rw-r--r-- root system
        - /etc/security/audit/hosts : -rw-rw---- root audit
- *Enhanced Security*
    - Keep the tripwire binaries and configuration files offline so that they will not be tampered with. If these files become vulnerable to attack, an intruder could possibly change the previous logs of important files to match that of the trojan horse they possibly planted.
    - /etc/*
        - The entire folder holds many secure files that should be monitored in higher security environments.

Vulnerability Assessment

What better way to test the security of a server using the same tools that intruders use? Many times, by using the most common tools that hackers use to check for vulnerabilities, an administrator is able to find holes in their system (possibly even before being rolled onto the network). There are certain elements that intruders try to find in order to exploit a server; know them and use them before the intruders do. By ensuring that common intrusion tools/rootkits are useless against a server, the amount of possible intruders has substantially decreased as many do not know much beyond the use of tools. Redundancy is not a waste of time. By cross-checking systems with multiple tools, an administrator can be certain that most rootkits will not be able to offer much information to an intruder.

Computer Oracle and Password System (COPS)
COPS is used to check a system for possible vulnerabilities based upon recent CERT advisories. COPS does not stay thoroughly up to date, so it should not be used as the end all of vulnerability assessment, but it is a good start towards ensuring a system is secure.
COPS: ftp:/info.cert.org/pub/tools/cops

NMAP
nmap is referred to as one of the strongest tools in a "rootkit" (hacking toolkit to achieve root status). System administrators should run nmap on their system and fix any vulnerabilities found. This will repeal most attacks, as many of the intruders don't have much more of a toolkit for vulnerability assessment besides nmap. Nmap offers many different variations, one of which is the "paranoid" mode. This will send the ping requests at 5 minute inteverals (if not more), which will usually not be detected by intrusion detection systems. Because of these variations that can bypass intrusion detection, it is of utmost importance that a system adminstrator run nmap before the server is rolled into production.
nMap: www.insecure.org/nmap/

Crack/John the Ripper
Crack and John the Ripper are very popular tools for cracking passwords. Like any other vulnerability assessment program, if the server's passwords can pass a brute force attack from these two programs, chances are that it will hold up to an intruder who is most likely using the same programs. both programs require that the password files be merged into one file. Since AIX uses a shadow password file, a merge script is needed and can be found the in Appendix of the AIX Security Supplement.
Crack: ftp://coast.cs.purdue.edu/pub/tools/unix/pwdutils/
John the Ripper: www.openwall.com/john/

Network Security Auditor (NSA)
Fileset (from IBM) : nsauditor.base. This program essentially works as a port scanner, but will give a good understanding (as many of the other programs listed) of a hacker's perception of the system.

It is always important to ensure that multiple assessment tools are used to offer the full picture of the server security status; this tool offers a great angle.
NSA : www-4.ibm.com/software/security/firewall/about

FPing
Fping is a network sweep utility that is used to map the network topology. This should be used for testing a network versus the documentation of the network. This can also be used to check for certain ports responses. On higher security servers, there should be very few (if any) responses the fping. Ensure that the broadcasts are also check (port 0 and 255). This is a common tool in a hackers toolbox, by ensuring that a system is not responding to the fping requests, the intruder has less information with which to launch an attack.
FPING : www-frec.bull.fr

SNORT
A very powerful network sniffer as well as filter to be used to enforce rules upon certain types of network traffic.
SNORT: www.snort.org

Nessus
Remote security scanner with over 1200 checks that will also offer suggestions for filling holes found in the audit.
Nessus: www.nessus.org

AIDE
Free AIX Tripwire replacement.
AIDE: www.cs.tut.fi/~rammer/aide.html

Security Administrator's Integrated Network Tool (SAINT)
SAINT is a very powerful tool that will scan for system vulnerabities (much like NSA and nmap). SAINT is constantly being updated, so to ensure that the newest audits are running, download the latest version of SAINT each time SAINT is to be run. A mailing list is available that will notify an administrator when a new version of SAINT is released.
SAINT: www.wwdsi.com/saint/index

This is a source code version and must be compiled before use. It is strongly suggested that a C compiler is kept off of production servers, so compile it on a proff of concept box and copy it over to the production PC.

SARA
Similar to SAINT but open source and not supported.
SARA: www-arc.com/sara

Maintenance

The maintenance of servers is important. Security flaws are constantly being discovered, and it is important that an administrator keep up-to-date. If just one server is neglected from security patches, and an intruder finds that one node, he could feasibly take out a network of patched/up-to-date systems. Implement the suggested features on all servers on a subnet.

Maintenance Level

Maintenance Levels are released by IBM and tend to be very stable and secure. There is thorough documentation on IBM's AIX website to deal with Maintenance Levels.

The Maintenance Level on a machine should always be the highest available. The only exception is if the system cannot go to a higher level because of dependency problems. If this is the case, security patches galore must be installed as these levels hold many security patches inside of them. IBM offers the compare_report command to help with automation. By downloading the compare report file of the present patches, this is then compared to the local server. Following this, the administrator then updates the output file to IBM, and a custom service pack with all of the necessary patches to bring the AIX system up to date is created. Depending on how often the system is patched, this can be a very large service pack.

After a maintenance level has been installed, run a security audit of the system to ensure the state of security has not been altered.

Crontabs

Crontabs are highly effective at automatic security auditing. Once strong crontabs are setup, as long as they are not changed or altered unknowingly (use Tripwire or a similar product to supervise any changes to the files), an administrator will watch the logs/flags for suspicious activity on those crontab logs. By automating many of these security audits, administrators are allotted more time to work towards offering stronger customer solutions.

Sample of Suggested Crontabs

Daily

- fin / -nouser –ls: This command will list all of the files that do not have a user; these files may or may not be needed, but should be set to a different user if they are necessary.
- find / -perm -4000 –user 0 –ls and fin / -perm -2000 –user 0 –ls: These commands will list all sticky bit files that can be potentially exploited for root permissions.
- awk –F: ‘{ if ($2 == “”) print $1 }’ /etc/passwd : This will give a listing of all of the users that are registered without passwords
- awk –F: ‘ {if ($3 == 0) print $1}’ /etc/passwd : This will show all users that have a UID of 0 (root user) – there should only be one unless there is other testing being done with another user.
- tcbck : This command ensures that the TCB contains no files that clearly violate system security and updates/adds/deleted trusted files.

- find all files that are writeable by world
- check all files executable by root to ensure that they are owned by root and their parent directory is owned by root
- check for .netrc files
- check for .rhosts files.
- check for .netrc files
- check for hosts.equiv files.
- COPS (more information in Programs section)
- Tripwire: possibly hourly depending on the severity of the files being monitored (multiple threads can be instantiated so that one may run daily, while another runs hourly) (more information in Programs section)

Monthly

- skulker –p: This command will remove all obsolete files from the following categories: /tmp directory; executable a.out files; ed.hup files. These files can be exploited from time to time and should be kept clear.
- compare_report : This command is used in concordance with IBM to get a listing of all of the filesets and compare them against present day filesets from an IBM website. The results can then be uploaded (automatically) and the updates can then be downloaded and reviewed prior to installation. –Refer to AIX compare_report command.

Security Patches

- Once at the highest maintenance level, ensure that all delta security patches (patches that are released after the most recent maintenance level) have been installed.
- Review emergency patches for IBM to ensure system is up to date.
- After install, run a quick check of major security functions to ensure that the patch has not compromised a different security feature.

E-mail/Websites

- Review security and system hardening related websites (including IBM) for new headlines or vulnerabilities.
- Review security forums and mailing lists. These get rather lengthy so it may be smart to start a generic hotmail account (company accounts would not be advised as to tip off others on the mailing list of occupation/location).
- Review attached resources as often as possible.

Log/Report Review

- Review the sulog for suspicious activity.
- Review the audit logs.
- Review the wtmp file for login/logout activity.
- Review the syslogd to review security issues.

- Ensure all crontab scripts are working correctly (refer to audit/tripwire checksum reports of chrontabs).
- Review all tripwire reports to ensure that all files being supervised remain authentic.
- All logs should be saved for 13 months in the case of an incident.

Live CD's/USB's

Live CD's are an intruder main rootkit now. These tools can be used on nearly any Intel-compatible system and gain them immediate root access. If launched on a benign computer, it leaves no trace and after an attack, is nearly impossible to trace.

Because of this, a security administrator is strongly urged to keep up to date with Linux Live CD's. These come out daily, and should be downloaded and run to see what tools are coming packaged. Tools are changing daily, and these are great barometers for the status of rootkits for intruders.

These CD’s can now be launched from USB drives. This offers the option of data extraction (as USB thumbdrives are write-able). Ensure that BIOS does not allow “Boot from USB”.

Resources

Mailing Lists (Create a new account for these as they will flood an inbox)
-Unix-security: security@cpd.com
-Security-misc: comp.security.misc
-Virus-list: comp.virus
-ACM risks: com.risks
-Cert-tools: cert-tools-request@cert.org
-Cert-advisory: cert-advisory-request@cert.org
-Security Announcements: comp.security.announce

Books
Practical Unix & Internet Security by Simson Garfinkel and Gene Spafford. O'Reilly & Associates, Inc., ISBN: 1-56592-148-8
Unix System Administration Handbook by Evi Nemeth et al. Prentice Hall. ISBN: 0-13-151051-7.

Redbooks
AIX Security
AIX Security Supplement (2000 and 2003 versions)
AIX Security Tools
AIX Administration

Tutorials/Whitepapers/Articles
Deploying Tripwire on AIX
Using IPSec as a packet filer on AIX
Deploying OpenSSH on AIX
Securing AIX Network Services on AIX
Strengthening AIX Security: A System Hardening Approach
Find out ways to protect your system from intruders
FAQ: Network Intrusion Detection Systems

Security Related Websites

Many of the best places to learn about those who we are defending from are from the exploiters themselves. This list offers many computer exploit websites (as well as various tutorials) that can offer great assistance into hardening servers.

ciac.llnl.gov	www.linuxsecurity.com/
www.cd.purdue.edu/coast/	www.microsoft.com/technet/security/
www.pclinuxonline.com/	www.net-security.org
Dir.yahoo.com/Computers_and_Internet/security_and_encryption/	www.niscc.gov.uk/
hack4u.owns.us/	www.packetstormsecurity.com/
kloosterboy.nl/	fux0r.phathookups.com/
localareasecurity.com	www.proxomitron.info/
Publib16.bouler.ibm.com/pseries/en_US/infocenter/base	www.radium.ncsc.mil/tpep/library/rainbox
Security-protocols.com	www.radium.ncsc.mil/tpep/process/faq-sect3
www.cerias.purdue.edu	www.rootcompromise.org/
www.cert.org/	www.roothack.org/
www.cymru.com/	www.rootprompt.org/
www.dc214.org/	www.sans.org/
www.defcon.org/	www.securityfocus.com/
www.eeye.com/	www.spidynamics.com/
www.faqs.org/faqs/computer-security	www.theregister.co.uk/
www.faqs.org/faqs/cryptography-faq/	www.trusecure.com/
www.first.org	www.uniras.gov.uk/vuls/2004/236929/index
www.freewebz.com/lexsdomain/index.htm	www.vnunet.com/Home
www.gocsi.com/	www.worldwidewardrive.org/
www.hackinthebox.org/	www-1.ibm.com/servers/security/planner
www.hackthissite.org	www-3.ibm.com/security/index.shtml
www.hackwire.com/	www.infosyssec.com/
www.iana.org/assignments/port-numbers	www.insecure.org/sploits_AIX.html
www.robertgraham.com/	hxdef.czweb.org/
www.l0pht.com/	www.tcim.com
www.governmnetsecurity.org	www.foundstone.com
www.ietf.org/	www.auscert.org.au/
ftp://ftp.purcupine.org/pub/security/index.html	http://www.intelligententerprise.com/info_centers
http://www.owasp.org/index	www.2600.com
www.alw.nih.gov/security/prog-full.html	www.ebcvg.com
www.securitywizardry.com	www.hackerslab.org
www.infosecuritywriters.com	www.hdcwargame.com
www.ossim.net	www.try2hack.nl
wargames.unix.se	www.mod-x.co.uk
www.arcanum.co.nz	lightning.prohosting.com/thegame
www.ralf-mengwasser.de	Digitalparadox.org
www.cyberarmy.com	www.learntohack.org
hackme.elderson.net	x-avier.com
www.slyfx.com	m4tr1x.wsn.at
vortex.labs.pulltheplug.com	www.hackerplayground.com
quiz.ngsec.biz:8080	

Conclusion

By reading through this paper and utilizing the checklist that accompanies it, an AIX administrator now has a base knowledge of security, server hardening, intrusion detection, auditing, and security tools. This knowledge can be directly applied to their servers and many vulnerable holes will now be filled. Bear in mind that many holes that exist have yet to be discovered. Therefore, it is critical that every AIX security-minded administrator maintains their knowledge of security by researching and referring to the Internet resources that have been attached. If there is ever a question about implementation of any of the suggested features, refer to the AIX Security manuals that were designated with the specified feature (all features have been documented).

Help other administrators by documenting all changes and vulnerabilities found. Many times one administrator will find a hole and fix it, while many other servers that may be on the same network are left vulnerable. Communication and documentation is essential to keep AIX servers secure.

By utilizing this paper and checklist, the overall security of all AIX systems will be dramatically improved.

“If you know the enemy and you know yourself, you need not fear the result of a hundred battles”
-Sun Tzu

Questions may be directed to randori82 [at] hotmail [dot] com.

Standard Security Measures

This is a summary of standard security implementations that should be incorporated in every system. The only explanation for not implementing a feature is that it causes conflict on that machine (in which case, a similar security fix should be sought out).
This list may also be used as an audit checklist for regular security systems.

Description	Check	Fix	Page
Security Policy			
Hardware Passwords	If the machine boots up without prompting for a password, the system is vulnerable.	Refer to BIOS manual and implement a BIOS password.	
Login Control	/etc/security/login.conf	sak_enabled:false logintimes:<none> logindisable:4 logininterval:60 loginreenable:30 logindelay:5	
System Resource Control	/etc/security/limits	Limit services and users from using too many resources.	
Global .profile's	Check permissions for $HOME/.profile	User profiles should not be writeable by anyone but root. Administration should create the profile files.	
Password Strengthening	/etc/security/user	dictionlist:NA maxrepeats:4 maxexpired:4 maxage:16 histsize:20 histexpire:26	
Null Passwords	**awk -F: '{if($2 == "") print $1}' etc/passwd**	Ensure that no users are seen to have a null password. Disable the account immediately or create a temporary password.	
Physical Security	Badge Readers, Camera Installations, Human Surveillance should be incorporated.	Incorporate the previously described security checkpoints.	
Trusted Computing Base	/etc/security/sysck.cfg	If TCB is not installed, the system will have to be reinstalled with the TCB option activated.	
User Control Administrator	/etc/security/user	Create two accounts (other than root) that have permissions to	

		write to the user config file.	
Distributed Root Authority	/etc/security/user	Separate the power of root between three sets separate administrators. One set for user control (as described earlier), one for file-system maintenance, and one for other privelaged commands (such as **mount**).	
System Hardening			
.netrc Files	Check Permissions of :$HOME/.netrc	Ensure permissions are 600 and the files are owned by root.	
Login Message	/etc/security/login.conf Herald : <login message>	Herald:"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\rWelcome to Server: <server name> \n\rUnauthorized Access is Prohibited\n\rlogin:"	
X11 Interception Vulnerability	**xwd** and **xwud** should be removed from /bin	Implement Secure Shell	
GUI Disabling	Check GUI interface at terminal	Remove KDE and GNOME if they are installed. CDE should be the only GUI to be used as the others can offer security risks.	
Direct Root Login	Check permissions of the command: /etc/securetty Should be 400 and should only have one entry in the file : console	Ensure that root login is only directly available for the console (which requires physical access). If CDE is being used on the system, check /etc/dt/config/Xstartup to ensure that root console login via CDE.	
Unnecessary Accounts	/etc/security/user	Disable guest and imnadm accounts.	
Unnecessary Groups	/etc/group	Disable uucp, printq, imnadm accounts.	
Unnecessary Services	/etc/inetd.conf /etc/inittab /etc/rc.nfs /etc/rc.tcpip	Refer to AIX Service documentation (AIX Security Guide) and disable services that are unnecessary to the function of the server.	
FTP Remote Code Execution Vulnerability	Check Permission of: /bin/ftp	Ensure that the sticky bit for owner is disabled.	
Remote Root Login	/etc/security/user	User Secure Shell to login remotely as root.	
Authentication	N/A	Implement OpenSSH for minimal	

		authentication instead of telnet.	
IPSec	lslpp -L '*ipsec*' should reveal: bos.msg.en_US.net.ipsec bos.net.ipsec.keymgt bos.net.ipsec.rte bos.net.ipsec.websm	Activate IPSec so that it logs all IP traffic for future auditing.	
SecureTCPIP	Test rcp, rlogin, rsh, and tftp commands. If any are available, securetcpip is not enabled.	Disable previously mentioned commands as needed. If all are unnecessary, run **securetcpip.**	

System Integrity

UID Control	**awk -F: '{if($3 == 0) print $1}' /etc/passwd**	If any users other than root are listed, disable those users.	
Audit Control	/etc/security/audit	Use audit subsystem **watch** command to watch for events on specific files	
Remote Syslog Exploit	**netstat -a | grep udp** : check to see if the syslog daemon is listening if there is no udp user, check all **netstat -a** entries.	Put a resource limit on syslog storage space to ensure DoS attacks will be disabled for remote syslog vulnerabilities.	
TCPD aka "TCP Wrapper"	/etc/inetd.conf	Ensure that this file will execute tcpd for audit purposes.	
PortSentry	www.psionic.com/abacus/portsentry	Launch port sentry to log possible port scans on the system.	
Tripwire	www.tripwire.com	Implement Tripwire on the following /etc files: crontabs, passwd, profile, hosts.equiv, ftpusers, security (directory), syslog.conf, publickey, .rootkey, keystore, shadow, filesystem, inittab, group, vfs.	

Vulnerability Assessment

The following programs should be launched against a system prior to rollout to production. Once in production, extreme care must be taken when using these programs. Ensure written authorization prior to launch of any of these programs as well as pass through Change Control.

COPS	**John the Ripper**	**SNORT**	**SAINT**
NMAP	**NSA**	**Nessus**	**SARA**
Crack	**FPing**	**AIDE**	

Enhanced Security Measures

This is a summary of enhanced security implementations that should be incorporated in higher security-demanding systems. The only explanation for not implementing a feature is that it causes conflict on that machine (in which case, a similar security fix should be sought out).
This list may also be used as an audit checklist for enhanced security systems.

Description	Check	Fix	Page
Security Policy			
Hardware Passwords	If the machine boots up without prompting for a password, the system is vulnerable.	Bios password as well as a power-on password should be activated as well as be unique.	
Login Control	/etc/security/login.conf	sak_enabled:true logintimes:<specify time values> logindisable:3 logininterval:300 loginreenable:360 logindelay:10	
System Resource Control	/etc/security/limits	Limit services and users from using too many resources.	
Global .profile's	Check permissions for $HOME/.profile	User profiles should not be writeable by anyone but root. Administration should create the profile files. Enforce Automatic Logoff by appending the following line to the profile: *TMOUT = 600; TIMEOUT = 600; export readonly TMOUT TIMOUT*	
Password Strengthening	/etc/security/user	dictionlist:/usr/share/dict/words maxrepeats:2 maxexpired:2 maxage:4 histsize:20 histexpire:52	
Null Passwords	**awk -F: '{if($2 == "") print $1}' etc/passwd**	Ensure that no users are seen to have a null password. Disable the account immediately or create a temporary password. Run this daily via **crontab**.	
Physical Security	Badge Readers, Camera Installations, Human Surveillance should be	Incorporate the previously described security checkpoints as well as incorporate biometrics or	

	incorporated.	secure authentication devices such as mobile storage USB sticks with authentication tickets or biometric authentication built in.	
Trusted Computing Base	/etc/security/sysck.cfg	If TCB is not installed, the system will have to be reinstalled with the TCB option activated. Thoroughly audit all programs in the TCB. Add programs if they are necessary and proven to be fully secured.	
User Control Administrator	/etc/security/user	Create two accounts (other than root) that have permissions to write to the user config file.	
Distributed Root Authority	/etc/security/user	Separate root power into more than three different administrators. The more modular, the more secure.	
System Hardening			
.netrc Files	Check Permissions of :$HOME/.netrc	Remove .netrc files.	
Login Message	/etc/security/login.conf Herald : <login message>	Herald:"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\rUnauthorized Access is Prohibited\n\rlogin:"	
X11 Interception Vulnerability	**xwd** and **xwud** should be removed from /bin	Implement Secure Shell	
GUI Disabling	Check GUI interface at terminal	No desktop environment should be installed.	
Direct Root Login	Check permissions of the command: /etc/securetty Should be 400 and should only have one entry in the file : console	Ensure that root login is only directly available for the console (which requires physical access).	
Unnecessary Accounts	/etc/security/user	Disable guest, imnadm, ipd, uucp, nuucp.	
Unnecessary Groups	/etc/group	Disable uucp, printq, imnadm accounts.	
Unnecessary Services	/etc/inetd.conf /etc/inittab /etc/rc.nfs /etc/rc.tcpip	Refer to AIX Service documentation (AIX Security Guide) and disable services that are unnecessary to the function of the server.	
FTP Remote Code	Check Permission of: /bin/ftp	Ensure that the sticky bit for	

Execution Vulnerability		owner is disabled.	
Remote Root Login	/etc/security/user	Disable remote login. **rlogin = false** for root user	
Authentication	N/A	Implement OpenSSH, PKI, Kerberos, PAM, and LDAP to suit the needs of the server.	
IPSec	**lslpp -L '*ipsec*'** should reveal: bos.msg.en_US.net.ipsec bos.net.ipsec.keymgt bos.net.ipsec.rte bos.net.ipsec.websm	Activate IPSec so that it logs all IP traffic for future auditing. Configure IPSec as a packet filter and define filtering rules.	
SecureTCPIP	Test rcp, rlogin, rsh, and tftp commands. If any are available, securetcpip is not enabled.	Run **securetcpip** every time TCP/IP is installed.	
Port Knocking	Dependent on level of security: could be local, offline, or even on a thumbdrive.	Implement port knocking on insecure protocol ports (such as telnet) if these must be used for certain functions.	

System Integrity

UID Control	**awk -F: '{if($3 == 0) print $1}' /etc/passwd**	If any users other than root are listed, disable those users.	
Audit Control	/etc/security/audit	Launch audit subsystem in full.	
Remote Syslog Exploit	**netstat -a | grep udp** : check to see if the syslog daemon is listening if there is no udp user, check all **netstat -a** entries.	Put a resource limit on syslog storage space. Disable syslog remote daemon.	
TCPD aka "TCP Wrapper"	/etc/inetd.conf	Use tcpd for audit. Use tcpd to block certain protocols such as telnet, rlogin, and finger from certain hosts. Incorporate with PortSentry to block port scanning hosts.	
PortSentry	www.psionic.com/abacus/portsentry	Implement PortSentry with tcpd to act as a dynamic port scanning host blocker.	
Tripwire	www.tripwire.com	Implement Tripwire on all /etc/*. Keep tripwire binaries and configuration offline so that they cannot be tampered with.	

Vulnerability Assessment

In a high security environment, keep an image of the production servers up to date on proof of concept boxes. The following programs should be utilized (at minimum) once a month to try to break the proof of concept box. Following an intrusion, create a stopgap and implement it on the original image of the production server. If the patch holds up and does not cause any conflict, report it to www.cert.org and implement it on the production server. Be creative.

COPS
NMAP
Crack
John the Ripper
NSA
FPing
SNORT
Nessus
AIDE
SAINT
SARA