Practical unix security - Securing IBM's AIX

Simon Taylor
IBM South Africa
70 Rivonia Road
Johannesburg
South Africa
simonta@za.ibm.com

Key words : AIX, unix, system security.

Abstract : This paper describes the process involved in securing a computer running AIX - IBM's version of unix. AIX is derived from the BSD and System V variants of unix, with the addition of a system and device configuration database - the ODM[1]. This has led to the creation of a large number of system-specific configuration and management commands which should be used instead of the default unix commands. For this reason, effectively securing an AIX system requires a system-specific approach. In any case, all versions of unix are sufficiently different to warrant a version specific implementation of general unix security guidelines.

1. BACKGROUND

One of the events that aroused my interest in computer security occurred during a service visit to a client. I had run a health check / information gathering script and noticed an unexpected .tar file in the root directory. When I asked the customer why the file was there, they mentioned that they had been security audited and that the auditors had reported that theirs was one of the better systems they had seen. I looked at the .tar file which had global read access and found that it contained a generic security checking script for unix as well as the script output including the shadow password file (again with global read access). In addition, root on an adjacent system had rsh access to the audited system.

This paper is not meant as an attack against generalized security standards, rather it seeks to demonstrate that standards (guidelines would be better) should not be slavishly followed. Those responsible for system security can, with moderate knowledge and the application of common sense create systems that are sufficiently secure to resist all but a deliberate, focused attack.

To illustrate the application of common sense to system security I would like to draw parallels between domestic security in Johannesburg and computer system security:

Everyone understands basic household security.

Home/Business	Generic Description	Computer System
Locked Door	Perimeter Security	Firewall. Router. Port closure.
Alarm System	Intrusion Detection	Port Monitor. Service Wrapper. File integrity checking. Log monitoring.
Security Company	Intrusion Response	Software or configuration update.

Most households and all businesses subscribe to an armed response security company. How many businesses implement the equivalent computer security activities?

1.1 Aix vs. Other unix variants

While AIX has roots based in both BSD and System V (witness the recent login buffer overflow)[3], it has sufficient unique aspects from an administration point of view to warrant a specific approach to implementing and maintaining system security. Many configuration parameters are stored in the ODM configuration database and are manipulated using commands that differ from other unix variants. Nevertheless, there are a number of security related configuration areas (mainly network parameters) that are not held in the ODM and are manipulated using AIX specific commands[4] (no - network options and nfso - nfs options). While most configuration parameters are stored in ASCII files, these files are not meant to be directly edited. For example:

/etc/inittab	mkitab, chitab, lsitab, rmitab
/var/spool/cron/crontab	crontab (-e|-l|-r|-d)
/etc/inetd.conf	chsubserver
/etc./services	chservices
/etc/passwd	mkuser, chuser, lsuser, rmuser
/etc/security/passwd	chsec, lssec, usrck, pwdck, grpck

In many cases, the commands shown act on more than one file and make synchronised changes. While it is possible to edit these files directly, doing so is likely to create discrepancies between files as well as causing changes to permissions and ownership's.

AIX also has a distinct approach to the management of system software installation and upgrades. System software is packaged to be manipulated by the "installp" program. This program interrogates the ODM to determine what software and hardware exists on the system and will apply or commit updates in an appropriate fashion. In most cases, copying system software between systems is guaranteed to cause corruption.

1.2 Common unix security vulnerabilities

SANS[2]/FBI top twenty vulnerabilities.

The SANS (System Administration, Networking, and Security) Institute and the FBI recently expanded the previously released SANS security vulnerability list:

<u>Vulnerabilities affecting all systems.</u>

1 - Default installs of operating systems and applications. Most installs are designed for ease of use rather than security.

2 - Accounts with No Passwords or Weak Passwords. Any password that is found in or derived from a dictionary word is trivial - almost any password cracking program will reveal it.

3 - Nonexistent or Incomplete Backups. If a system cannot be easily and quickly restored to a working state, it is seriously at risk.

4 - Large number of open ports. Many services exist for historical reasons and are not commonly used.

5 – Not filtering packets for correct incoming and outgoing addresses. Assume that all network traffic is benign.

6 - Nonexistent or incomplete logging. If no log of normal behavior exists, how can abnormal behavior be detected?

7 - Vulnerable CGI Programs. CGI programs are designed for external invocation so are more sensitive than other programs.

Unix specific vulnerabilities

1 - Buffer Overflows in RPC Services. RPC - Remote Procedure Calls are designed for external invocation.

2 - Sendmail Vulnerabilities. Widely used and interprets externally generated input.

3 - Bind Weaknesses. Widely used and accepts external commands.

4 - R Commands. Remote access commands.

5 - LPD (remote print protocol daemon). Widely used and accepts external commands.

6 – sadmind and mountd. Sadmind is Solaris' remote administration server and mountd accepts external commands.

7 - Default SNMP Strings. Simple Network Management Protocol - accepts external commands sometimes without access control.

It can be seen that any program or service which accepts external input is a popular target.

Windows specific vulnerabilities have been ignored. Quote from the SANS Web Page http://www.sans.org/top20.htm. These few software vulnerabilities account for the majority of successful attacks, simply because attackers are opportunistic – taking the easiest and most convenient route. They exploit the best-known flaws with the most effective and widely available attack tools. They count on organizations not fixing the problems, and they often attack indiscriminately, scanning the Internet for any vulnerable systems.

2. OBJECTIVE

2.1 Improve the security of an average system so that it is no longer a soft target.

Organizations should (at minimum) harden systems so that they do not fall victim to random or opportunistic attacks. Sufficient monitoring should be performed so that an attack or unwelcome attention can be detected before actual intrusion takes place. See network monitoring.

2.2 Reasons for increasing system security.

2.2.1 Is a firewall defence enough?5

A common reaction when the question of system security is raised is, "We already have a firewall". My response is, "the Titanic had watertight bulkheads, but the lifeboats were still needed."

While I agree that a properly configured firewall is an important element of a secure environment, it must not be the only form of defense. Firewalls exist to provide controlled access to computers. They are normally configured to permit access through a small number of communication ports, typically providing for mail, name resolution and html traffic. They do not normally differentiate between normal and hostile traffic over these ports.

2.2.2 Corporate audit requirements.

The requirements of corporate auditing departments must be addressed, but it must be remembered that few hackers have respect for corporations.

2.2.3 Automated scans/attacks.

Many of the recent high profile security problems - Nimda, Code Red, Ramen[6], etc. have scanning code built into them. The scan code and payload typically do not contain logic to avoid non-vulnerable operating systems.

At least one security analyst[7] has predicted that a Nimda style attack will be launched using the system V login buffer overflow vulnerability.

3. SECURING A SYSTEM

3.1 Basic steps

Hiding your money under the mattress. It may not be very secure, but it's better than leaving it in plain sight.

3.1.1 Analyse the system.

System analysis should provide sufficient information to asses the potential vulnerability of the system as well as being a starting point for documentation. Analysis should cater for both external and internal access. Example: Which network ranges can access the host, and in the case of logged in users, are members of one group able to access data belonging to other groups? In addition to generic analysis tools such as nmap, nessus, sara and saint[8], the following AIX specific checks should be performed:

password and group files should be checked with usrck, pwdck and grpck; these are standard AIX utilities to check and maintain synchronisation between the password and group files and their shadow counterparts.

Network configuration settings should be checked; AIX uses the command 'no' to set and display network configuration parameters.

NFS configuration settings should be checked; AIX uses the command 'nfso' to set and display network file system configuration parameters.

3.1.2 Disable unnecessary / insecure services.

Common services

AIX, in common with many unix variants enables common services which are considered to be insecure. These services include echo, chargen, daytime, statd, rusersd, rwalld, sprayd, pcnfsd, discard and time. These services are started by the inetd daemon and can be disabled by modifying the /etc/inetd.conf file and restarting the daemon.

NFS
The use of NFS is not desirable in a secure environment[9], but if NFS must be used, at minimum, privileged port checking should be enabled.

Sendmail
The default sendmail configuration allows sendmail to display both the operating system and sendmail program versions as well as permitting user verification and mail list expansion. If sendmail service is required, the configuration should be altered to prevent this information leak.[10]

FTP and Telnet
Both ftp and telnet transfer user data in clear text on the network. They should never be used over an external network and should only be used on an internal network when it can be guaranteed that no traffic monitoring can be done.

Modify insecure configuration values.

Default path:
The default path for AIX is defined in /etc./security/.profile and includes the 'current directory' as the last entry. While this may be convenient for users familiar with DOS systems, it is insecure.

The root user's default path should contain only system directories containing commands required for the administration of the system.

Remote access:
The /etc/hosts.equiv file should either be empty or removed and no users should have a .netrc file in their home directory.

Network options:
arpt_killc - arp buffer time-out; default value is 20 minutes. To avoid arp buffer poisoning attacks, this value should be reduced to between 1 and 5 minutes.

Bcastping - response to a broadcast ping message; default is to ignore.

Directed_broadcast - permit directed broadcast; default is to allow. To prevent undesirable system use, this should be disallowed.

Clean_partial_conns - disconnect connection attempts which have not been successful; default is not to disconnect failed connections. To avoid SYN flood attacks, this should be disallowed. Ipforwarding - forward packets destined for a remote network; default is to deny. This should only be enabled if the system is to act as a router.

Ip6srcrouteforward - forward IP version 6 source routed packets; default is to allow. This should be disabled to prevent undesirable system use.

Ipsrcrouteforward - forward source routed packets; default is to allow. This should be disabled to prevent undesirable system use.

Ipsrcrouterecv - accept source routed packets; default is to disallow. This should not be enabled.

Ipsrcroutesend- transmit source routed packets; default is to allow. Source routed packets should not be transmitted.

Ipignoreredirects - process redirected packets; default is to allow. Redirected packets may be received as the result of hostile action and should not be processed.

Ipsendredirects - send redirected signals; default is to allow. This should only be enabled if the system is to act as a router.

Rfc1122addrchk - perform RFC1122 address validation; default is to allow. This should be disabled to block incoming & outgoing SYN packets aimed at loopback and multicast addresses.

Tcp_pmtu_discover - perform TCP MTU discovery; default is enabled. This should be disabled to avoid a potential DOS attack.

Udp_pmtu_discover - perform UDP MTU discovery; default is enabled. This should be disabled to avoid a potential DOS attack.

Nonlocsrcroute - control strict source routing outside the local network; default is disabled. This should only be enabled if IBM PSSP topology service support is required.

NFS:
As mentioned before, this should not be used, but if it is required, it should be configured as securely as possible; AIX permits the use of secure ports (those lower than 1024) as well as allowing encrypted NFS traffic (between two AIX systems).

Sendmail:
AIX ships with sendmail enabled. The default configuration should be altered to prevent the display of operating system and sendmail version in the connection banner. User verification and list expansion should also be disabled.

Name resolution:
If named is to be run, care should be taken to ensure that zone transfer information is severely restricted and that version information is not displayed.

System logging:
AIX ships with the syslog daemon enabled, but with an empty configuration file. Log files should be enabled and logs examined on a daily basis for evidence of illegal or hostile activity.

Information leaks:
Any program or service which displays information to casual access should be identified and leaks stopped. An example is the welcome message displays at initial login - normally the operating system version is displayed. This should be replaced with a restricted use banner. This is often a recommendation of auditing firms[11].

3.1.3 User and password management.

AIX provides a number of utility programs for the management of users and groups, these include mkuser, chuser, lsuser, rmuser, chsec, lssec, usrck, pwdck, grpck; these programs permit normal and shadow files to be kept synchronised. Additional programs and scripts should be used to ensure that users meet security standards, for example to check whether users have logged in within a specified time period, or whether they have changed their passwords within the defined interval.

User removal needs special attention - rmuser removes a user id, and optionally the home directory, but a search should be run for files owned by a user anywhere on the system before that user is removed.

3.2 Advanced Steps.

Keeping your money in the bank. A bank vault is designed to resist attack for a fixed time period. It is also monitored to detect attempted attacks.

3.2.1 Securing network access/Implementing openssh or openssl.

As far as network access is concerned, a system connected to any network should be considered vulnerable. A secure approach is first to disable all access and thereafter to grant access only as required. While this may sound totalitarian, the alternative is to grant access to anyone on any port and then to attempt to apply restrictions. [12]

A valuable tool for the management of ftp and telnet access is Wietse Venema's tcp wrapper program. This open source program is started by inetd in place of the standard telnet and ftp programs which it invokes after checking that the calling address is allowed access. It is simple to compile, install and configure on an AIX system. It has the added benefit of recording successful and failed connection attempts for all services it protects. [13]

So far, so good. Unfortunately, both ftp and telnet transmit all traffic across the network "in the clear", which means that anyone with a network connection and a traffic monitor can read your user id and password as well as your transaction information. Ftp makes this particularly simple as the password is transmitted in a single packet unlike telnet which tends to send each character of the password in a separate packet. This may sound far fetched - why would someone sit and monitor network traffic just to find your password. Well, the simple answer is that they don't, unless they are unaware of network password sniffers - these tools will log hostnames, protocols, userids and passwords to a file for subsequent use. [14]

The solution to this problem is openSSH [15] - an open source program which can be used to replace both ftp and telnet and encrypts all traffic (not just passwords). OpenSSH relies on openSSL [16] - a secure socket layer (network traffic is encrypted). OpenSSL and openSSH compile for AIX, but openSSL requires a source of genuine randomness to ensure that the encryption used for a session cannot feasibly be broken. Other operating systems have /dev/random as a source for truly random numbers. AIX requires egd - entropy gathering daemon to serve as a source of true randomness. Again this program is open source and easy to compile and install.

Note that commercial versions of this software are available.

3.2.2 Network monitoring.

How do we know if our system is under attack? In most cases, we only find out if the attack was successful and if the attacker was foolish enough to leave traces. This is obviously unsatisfactory. A far better approach is to test for "attack signatures". A simple example of this would be to examine the /etc./security/failedlogin file. If there have been a hundred failed login attempts logged for the root id since yesterday, it is reasonable to assume that our system is under attack.

What about other access methods? Once again, the open source community as well as the commercial sector have solutions. Network intrusion detection software can be installed on a host system and configured to detect known attacks. Unless an attacker is very sophisticated or very stupid, an attack attempt would be preceded by some form of port scan [17] (either targeted at specific ports or at the system in general). Traffic which could be viewed as a prelude to an attack (or at least unwelcome) would be a scan aimed at port 53 (DNS). If this traffic takes the form of a connect or partial connect, followed by a DNS version query or a request for a zone transfer it is time to batten down the hatches. This kind of traffic will pass straight through your firewall (assuming that DNS traffic is permitted - which it usually is) and may not even raise alerts there! The normal reason for a DNS version query is to check whether your version of the name daemon is remotely exploitable. Again if this seems paranoid, we have logged precisely this type of traffic on systems in Johannesburg where the queries initiated from the Netherlands, the USA and Korea (a handful of .edu or .ac addresses in Seoul)! I have also seen systems successfully hacked using name daemon vulnerabilities.

A host based Network Intrusion Detection system that I have used successfully on AIX is Snort[18]. This open source program uses rule/signature based detection mechanisms. It provides individual host or subnet monitoring. The rules are easy to create and manage and are regularly updated. It is easy to configure to avoid false alerts. It is reasonably easy to compile, but relies on the portable packet capture library libpcap which is more difficult to compile for AIX. Fortunately, AIX 5 includes libpcap.

3.2.3 File monitoring.

What do we do if our system was successfully attacked? First we have to recover from the attack - restore damaged/trojaned/deleted files. Then we have to guard against a repeat visit. How can we tell which files have been affected and whether trojaned programs or back doors exist?

This is a difficult task, but it can be made easier if we have some sort of reference. This is where file monitoring can be useful. A file monitoring system will keep a record of all the important attributes of a file or directory including permissions, size, modification time and checksum. Obviously not all files can or should be monitored as some will change (at least in size) many times in the course of a day. Nevertheless, selected files can provide clear evidence of an intrusion. On an AIX system, most (if not all) of /usr, /etc. and /var should be monitored. If maintenance has not been applied to the system and the login, ps, who, netstat and ls programs changed at the same time you can be reasonably confident that a rootkit has been installed. Note that decent rootkits will attempt to keep the same time stamps on all of these files, good ones will not only succeed, but may maintain the same simple checksums and file sizes. They are highly unlikely to manage to maintain the md5 checksum though and the selection of a file monitor should take this into account. The use of a simple script to check files may not be adequate. Again file monitors are available in both open source and commercial versions. Tripwire[19] is one of the better known commercial versions and Aide[20] is a good open source program. Aide is easy to compile for AIX and is reasonable straightforward to configure and manage. A useful bonus of using a file monitoring program is that backups can be checked to ensure that all modified files are in fact part of your backup plan.

3.2.4 Log Management.

It will be evident that the increased level of monitoring and logging will create problems of its own - particularly where large or multiple systems are concerned. One of the great things about the unix operating system is that there are already solutions to most problems and log file management is no exception. Swatch and Logsurfer[21] are both open source (again) programs that can be used to wade through your log files and to alert you when predefined conditions occur. They can monitor logs in real time or as batch processes and are valuable tools.

4. MAINTAINING SECURITY

As long as hackers search for ways to attack systems, we will have to continue to maintain and improve security. It seems likely that hackers will continue attacks, so maintaining security is something we will have to live with. The challenge is to integrate security maintenance and management into normal operations.

4.1 Regular Audits.

The effective security of a system cannot be ensured without regular auditing. Audit components range from the ongoing tracking of network and file monitoring logs to running simulated attacks against the system. Another recommended audit tool is to attempt to crack user passwords - to ensure that sensible passwords are chosen. There are a number of tools available for attack testing and password cracking including

Nessus	external attack
Saint	external attack
Sara	external attack
Whisker	http server probe
Crack	password cracker
John the ripper	password cracker

Again, this may seem destructive or insecure, but we have found that crack (for example) can guess a weak password within one or two seconds.[22]

Part of the task of maintaining a secure system is in keeping up with attack techniques and attack trends. There are a number of organisations which publish a daily or weekly newsletter on security related issues and attack trends. These include CERT, SANS, ISS and Neohapsis.[23]

4.2 Keeping up to date with maintenance releases.

All software vendors produce official maintenance releases at fairly frequent intervals and IBM is no exception, AIX is currently available in two versions; 4.3.3 and 5.1.0. Maintenance release 9 is presently available for 4.3.3 and maintenance release 1 for AIX 5.1.0. It is common practice to upgrade to a maintenance release within three months of its becoming available. In the case of a system where security is considered important, maintenance releases should be closely tracked. In the case of AIX, software maintenance can be applied and the previous version saved. This allows rollback if that should become necessary.

4.3 Security related software patches (APARs in IBM-speak).

IBM releases individual software patches as and when it becomes necessary. The majority of these patches are to provide improvements in function or to resolve problems. Some are to resolve security related problems. Individual patches are incorporated in the next subsequent maintenance release. IBM provides a subscription service that will send an e-mail notification when patches become available.[24]

4.4 Security emergency fixes

From time to time IBM releases emergency fixes in response to known threats. These threats are normally CERT documented. The difference between efixes and patches is that efixes involve manual replacement of existing files or programs. An efix should be installed on a system if that system is potentially vulnerable to the threat. An efix was released for the System V login vulnerability at the same time as the threat was made public. Once again, IBM provides an e-mail subscription service for efixes.

5. SUMMARY/CONCLUSION

My personal feeling is that it is (relatively) easy to secure a system as a "once-off" project. This is because a large quantity of information is available on general and specific vulnerabilities as well as advice on their resolution.

Maintaining a secure system is more challenging. Changes to system and application configurations may introduce vulnerabilities. New attack methods will continue to arise. It is difficult for the average organisation to devote significant time to security (to be fair, this is not the primary function of most computer systems).

What can be done?
Security can be outsourced.

This option should be exercised with care. As mentioned earlier, a generic approach to security could result in system or environment specific loopholes remaining unplugged. Additionally, the "at arm's length" approach to managing system security is not ideal. If the entire computer operation is outsourced, the ability of the outsourcing agent to provide effective security should form part of the selection process.

I believe that maintaining a secure system should be an active process. An intrusion detection system should be deployed and its logs should be used to assess the actual threat that systems are exposed to. Further security measures can be deployed (or not) based on real rather than theoretical risk.[25] This does not mean that regular audits and tests are unnecessary.

I would recommend that companies form a security relationship with vendors or specialists who support their operating system and application environments. These relationships should result in the regular flow of information and advice on specific security issues. The practical benefit is that the vendor or specialist can provide information on issues which are likely to affect specific environments.

6. REFERENCES / LINKS

1 ODM (Object Data Manager)
http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixgen/topnav/topnav.htm

2 SANS/FBI Top Twenty Vulnerabilities http://www.sans.org/top20.htm

3 CERT Advisory CA-2001-34 Buffer Overflow in System V Derived Login http://www.cert.org

4 AIX-specific commands
http://www.rs6000.ibm.com/doc_link/en_US/a_doc_lib/aixgen/topnav/topnav.htm

5 Marcus J. Ranum "Thinking About Firewalls" http://www.ranum.com/pubs/think/index.html

6 Nimda - "The worm searches for Web servers using randomly generated IP addresses."
http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html

Code Red - "attempt to exploit more computers by targeting random IP addresses."
http://www.symantec.com/avcenter/venc/data/codered.worm.html

Ramen - "The first network activity caused by the worm is the Synscan probe of the random class b address space."
http://www.whitehats.com/library/worms/ramen/

Note: While none of the above worms actively attack AIX, they do not avoid it and may negatively affect services on ports that are scanned or targeted.

7 John Pescatore "Special to CNET News.com" December 17 2001

8 nmap - documentation and download from http://www.insecure.org
Nessus - documentation and download from http://www.nessus.org
Sara - documentation and download from http://www-arc.com/sara/
Saint - documentation and download from http://www.dsi.com/saint

9 NFS http://csrc.nist.gov/publications/nistpubs/800-7/node148.html

10 Sendmail http://www.sendmail.org/security.html

11 and 12 National Security Agency "The 60 Minute Network Security Guide" http://www.nsa.gov

13 Tcp wrapper - documentation and download from ftp://ftp.porcupine.org/pub/security/

14 Password Sniffers http://www.faqs.org/faqs/computer-security/sniffers/

15 OpenSSH http://www.openssh.com

16 OpenSSL http://www.openssl.org

17 Port Scanning - John Wack, Miles Tracey "DRAFT Guideline on Network Security Testing" *Recommendations of the National Institute of Standards and Technology*

18 Paul Innella and Oba McMillan "An Introduction to Intrusion Detection Systems"
Snort - documentation and download from http://www.snort.org/

19 Tripwire http://www.tripwire.com/

20 Aide http://www.cs.tut.fi/~rammer/aide.html

21 Swatch http://www.stanford.edu/~atkins/swatch/
Logsurfer http://www.cert.dfn.de/eng/logsurf/

22 Whisker http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=2
John The Ripper http://www.openwall.com/john/
Crack / Libcrack http://www.users.dircon.co.uk/~crypto/

23 CERT http://www.cert.org
SANS http://www.sans.org/newlook/home.htm

ISS http://http://www.iss.net/index.php
neohapsis http://www.neohapsis.com/

24 IBM e-mail subscription http://techsupport.services.ibm.com/server/listserv
AIX Software Fixes http://service.software.ibm.com/rs6k/fixes.html