Issue Date: IBM edition for UNIX
October 2003, Posted On: 10/1/2003

Security for All

Jaqui Lynch

With the increase of e-commerce and on demand computing, security has never been more critical to business and non-business communities. It only takes one well-publicized break-in where key private data such as credit card numbers are stolen for a company to find its customers moving elsewhere. With the increase in privacy laws and HIPAA compliance requirements, protecting customer data and the systems where this data resides has never been more important.

Out of the box, a UNIX* system does have some vulnerabilities, ranging from services left activated by default—such as bug-ridden mail servers that start automatically—to poorly configured default permissions on files. By default on AIX*, logging isn't initially configured, which makes it difficult to debug a problem when it occurs.

The key elements to dealing with security issues include developing a response plan for attempted or successful break-ins and a security plan with standards for locking down systems.

Response Plan

The response plan shouldn't only be a list of technical people who need to be contacted in problem situations. It should also include copies of policies, evidentiary gathering rules, contact information for management, lawyers and law enforcement, and any other related information.

A clearly written acceptable use policy (AUP) is a must, since it not only protects the company but also makes the rules clear to everyone. All employees and users should be required to read, sign and accept the AUP's agreement.

Protecting Your AIX Investment

People and technology are two important areas to consider when locking down a system. Common sense involves ensuring that all personnel understand that company data, especially passwords, aren't to be shared or written down. The password schema, however, must not be so complex that no one can remember it. Giving someone a password like GH!@YhkO9AS forces them to write it down. Likewise, 123456 is just as unwise because it's easily cracked. Password rules need to be reasonable and comprehensible.

The technical side of security involves understanding the latest hacker methods and taking proactive steps to prevent the attacks—or at least making your system harder to break into so the hacker gets frustrated and goes elsewhere. This involves layering security, including protection from those external and internal to the firewall. It should also include plans for protection against those with legitimate access to the system.

Logging

A key component in dealing with attackers is to implement a good log strategy. syslogd runs by default on AIX, but the log configuration file isn't set up to actually log. It's best to set up a separate filesystem for logs (e.g., /usr/local/logs) rather than /var/spool. If /var fills up, the system will crash; if your separate filesystem fills up, it will just stop logging. Logs can be written to a file, sent to the console, logged to a central host across the network (be wary of this as the traffic can be substantial), e-mailed to an administrator or sent to all logged-in users. The most commonly used method is writing to a file in a filesystem. Once the filesystem is set up, the next step is to code a /etc/syslog.conf file. An example file would be:

```
*.emerg       /usr/local/logs/syslog
*.alert       /usr/local/logs/syslog
*.err         /usr/local/logs/syslog
auth.notice   /usr/local/logs/authlog
mail.debug    /usr/local/logs/mailog
daemon.info   /usr/local/logs/infolog
*.emerg       /dev/console
*.crit        /dev/console
```

In the file above, all emergency, alert and error messages are written to the /usr/local/logs/syslog file. Critical and emergency messages are also written to the console. Authentication messages should go to a separate log (authlog) and mail messages to a separate mailog. Daemon messages are set to go to infolog—daemon.info is where certain daemons, including TCP Wrappers, are configured to log to. This separation of the logs makes it easier to search for patterns and problems. Once the syslog.conf file is coded, the log files are created using the touch command:

```
touch /usr/local/logs/infolog
```

Next, the syslog daemon should be stopped and started using:

```
stopsrc -s syslogd
startsrc -s syslogd
```

In order to analyze the logs, it's important that the timestamps on all affected systems are correct and consistent. Some type of time-synchronization mechanism, such as network time protocol (NTP), should be implemented. Typically, one server is set up as the timeserver and all other servers point to it to synchronize their time. In large networks, it may be necessary to set up a time hierarchy so one server isn't overwhelmed by requests.

Logs need to be monitored regularly or post-processed to look for patterns or problems, as should performance statistics. For example, if network bandwidth utilization to the box jumps to 100 percent, it's possible you're being used as a warez drop site, or someone is moving big files to and from this system.

The Root of the Problem

Obtaining root access is nirvana for a hacker. Root access should be limited to only those systems administrators who absolutely need it. Administrators should log in as themselves and then use su to become root. However, su provides minimal logging, so a better option is to implement sudo, a freeware tool that provides excellent logging and allows a user to get enhanced access. sudo also provides methods for limiting certain users to specific commands, access from certain systems, and so on. It uses a configuration file called /etc/sudoers, which has a great deal of granularity. (Note: For more sudo coding information, visit www.courtesan.com/sudo.)

Another way to keep hackers at a safe distance is to make sure your secured system is patched. This means keeping up to date with the latest alerts from organizations such as The CERT Coordination Center (CERT/CC), a center of Internet security expertise operated by Carnegie Mellon University. Once an alert has been located, go to the Fix Delivery Center at IBM and install the associated patches or maintenance levels.

Also, it's important that regular checks are done to help ensure that all accounts on a system are valid and have valid user IDs. Accounts, such as those labeled guest, should be disabled, and there should be no group (shared) accounts or accounts without passwords. Accounts that are needed to own services but aren't logged into should be set to a dummy shell such as /bin/false or /dev/null. (Make sure these are in /etc/shells first.) A policy should also be established to determine the umask default for all accounts. I usually use 077, which sets permissions to rwx for files. However, make sure you don't set system accounts, such as root, to this policy.

Hackers will try to get their hands on anything they can—so don't give away unnecessary information. By default, the login prompt at the console and the message of the day (motd) at login state the flavor and version of the OS. This makes it much easier for the hacker, as they now know the OS they're targeting. The login prompt for an AIX system can be modified by editing the /etc/security.cfg file and coding a herald line.

You should also regularly scan the system for files that open up the system for insecure access. These files include .netrc, .rhosts, .shosts and .exrc files. Check the .forward files to ensure they aren't executables, and regularly scrutinize the permissions on .login and .profile files. Also check the /etc/hosts.equiv file (and shosts.equiv) to make sure no one has added remote access without password requirements. I usually delete all the lines in this file so that it's empty. The same applies to the /etc/hosts.lpd file, unless this is a print server. Both hosts.equiv and hosts.lpd should have group-write access removed in their permissions.

By default, many services in /etc/inetd.conf are still enabled out of the box. This includes many that have major security holes, such as ttdbsrvr and cmsd. Previous recommendation suggested the administrator comment out the unused services. However, when maintenance is run on the system, it isn't uncommon for items that were commented out to get added back in. The current recommendation is to copy the /etc/inetd.conf file somewhere (e.g., to /etc/inetd.conf-orig-date) and delete everything in the /etc/inetd.conf, except the specific services that should be enabled.

Another alternative is to leave the services there but set them all to /bin/false so they would never execute successfully. One reason for this approach is so log entries (when TCP Wrappers are installed) for attempted accesses are generated. Maintain a list of the services that are disabled on a system so these are once again disabled during upgrades.

/etc/rc.tcpip should also be edited to disable services that aren’t needed. For example, if you aren’t using snmp, comment it out and also comment out dpid2 in rc.tcpip. This is also where you would comment out the startup of sendmail, which, by default, starts at boot time. Once everything is commented out or deleted, reboot if possible. If a reboot can’t be scheduled, stop the services such as sendmail and dpid2, and refresh the inetd daemon so it rereads its configuration file.

It’s also useful to secure disallowed services in case someone accidentally enables them later. An example of this would be tftp. If you code a /etc/tftpaccess.ctl file, any successful tftp login can only access the directories or files listed. An empty tftpaccess.ctl file means there’s no access. If you’re in a clustered or SP environment, there are specific directories that must be enabled in the access file.

Useful Third-Party Software
Although many companies may have policies against downloading third-party software, this has now become a critical component in securing your AIX system. In fact, now it’s virtually impossible to secure a system well without some of these tools. Section 5 of IBM’s Redbook, “Additional AIX Security Tools on IBM eServer pSeries*, IBM RS/6000* and SP/Cluster” (SG24-5971), has an informative section on some of these tools. There are several tools that no system should be without, including SSH, SUDO, TCP Wrappers and LSOF.

SSH—This tool was written to replace insecure protocols such as rlogin, rcp, ftp and telnet with more secure alternatives. SSH provides authentication, encryption and data integrity across the Internet. Unlike protocols such as telnet, SSH encrypts all authentication traffic, helping ensure that usernames and passwords don’t travel in the clear. SSH also allows the option of standard UNIX passwords or the use of a public/private key pair for authentication. SSH interfaces with TCP Wrappers for logging and access control and has its own built-in access control. The tool includes several features—such as the ability to do secure backups, tunneling and X11 forwarding—all within a secure environment. SSH also comes with a secure ftp server, sftp.

TCP Wrappers—This program is called by inetd before calling a service. The wrapper checks two rules files (/etc/hosts.deny and /etc/hosts.allow), logs the attempt, authorizes or denies the attempt and builds an audit trail. It only does this for services that have been told to take advantage of the wrapper. In Figure 1, both ftp and telnet are using the wrapper. You can see that the program tcpd has been inserted to execute before the ftpd or telnetd programs. Another option is to actually replace telnetd or ftpd with the wrapper. However, this causes problems when maintenance is run, so the best method is to install the wrapper as a separate program, as Figure 1 shows.

Also note that rshd and rlogind are configured to take advantage of the wrapper logging. In these cases the wrapper will log attempts to use these services, but the services have been set to execute /bin/false rather than the real daemon. Because the wrapper won’t log for services that don’t call it, this is a way to get log entries for attempted break-ins without allowing the service to actually run.

TCP Wrappers uses two files to control access to the system. /etc/hosts.deny controls the denying of access, and /etc/hosts.allow controls the allowing of access. In order to keep things simple, I put all:all in the /etc/hosts.deny file. This means that all access to the listed services is denied unless it’s explicitly allowed in the hosts.allow file. The hosts.allow file can be configured to post a banner for each service, whether the service is granted or not. This is one way to help ensure that users see the AUP. It can also be used to execute a command when a connection is denied, or to issue an ident lookup when someone attempts a connection. Rules can be coded using IP addresses, domain or system names, or hostmask/IP combinations. There are two special keywords: all, which means anyone; and LOCAL, which means the system itself. If you want someone on the system to telnet to it, you must include the LOCAL option, IP or name of the server in the allowed list:

Sample /etc/hosts.allow

```
portmap: 123.123.  255.255.255.
ftpd : .abc.com,123.123.123.4
```

```
sshd : all
telnetd : 123.123.123.0/255.255.255.0
xmservd : .abc.com,123.123.123.4
rexecd : LOCAL,.abc.com,123.123.123.4
```

LSOF—This program lists open files on a system. Since all network connections are open files, it's possible to get information about who is connected to a port. Commands of use are:

```
lsof  |  grep TCP |  more
lsof  |  grep UDP |  more
```

Portmap—Many of the current vendors still ship a version of portmap that allows anyone to read or modify its tables, and will forward any request so it appears to come from the local system. Wietse Venema, who now works at IBM's Watson Research Laboratory, rewrote portmap to incorporate access control. The replacement product can be used to lock down access to services such as NIS, NFS and other RPC-based services. It's similar to the TCP Wrappers package in style and provides security libraries and access control lists to some of the services.

Stunnel—This tool provides a wrapper utility for encrypting TCP sessions using SSL. It requires users to download and install Openssl and can be used to secure additional daemons such as imapd, pop and ldap. It has built-in TCP Wrappers support and uses the /etc.hosts.allow file for security rules.

Mail Servers

By default, the sendmail server with AIX is enabled and allows blind relaying. This is a major security problem and needs to be addressed by either turning off sendmail, configuring it correctly or replacing it with Postfix, which can be obtained from Venema's Web site, porcupine.org. Regardless of whether sendmail is enabled or not, an aliases file (usually /etc/aliases) should be created so root and postmaster mail has somewhere to go. Once this file is updated, it's necessary to run the newaliases or "sendmail -bi" command to prompt the mail server to use the new set of aliases.

Rootkits

No discussion of security is complete without mentioning rootkits. Hackers are trying to obtain root access on the system; once this access is acquired, hackers usually install a rootkit that contains all the tools they need. This includes tools that replace standard UNIX commands such as ls, ps, ifconfig, netstat, etc. They replace ls and ps with commands that act normally, but don't show the hackers' files or processes. Telnet may be replaced by a version that acts normally but saves the information (user ID and password) to a file or e-mails it to the hacker. Rootkits get hidden in directories that may not be spotted easily, such as ".. " (.. followed by two spaces) or in /dev as fake pty files. If your system has been "rooted" the only way to secure it is to rebuild from scratch using trusted installation CDs. Anything that gets reinstalled from the hacked system needs to be manually inspected for signs of these fake directories and replaced files. Some tips for looking for rootkits include:

```
ps -no-headers -ef | wc
ls -d /proc /[0-9]* | wc
```

(The above two commands should show the same number.)

```
file /dev/* | grep text
```

(Look for items like /dev/ptyw showing up as ASCII text.)

```
Find / -name ".*"
```

(Look for anomalies such as ".." in directories. Since all directories have a "." entry and a ".." entry, this will be a long list.)

Summary

Security is everyone's business and everyone's responsibility—from the user who has a user name and password to the administrator who sets up the system to the management who determines the access policies. It’s essential to have policies and procedures in place to protect the systems in a layered manner. This helps ensure that security is in place at every level—from the firewall right down to the internals of the system.