Securing AIX Network Services

Skill Level: Intermediate

Sandor W. Sklar (ssklar@stanford.edu)
Systems Administrator
Consultant

24 Dec 2001

Better understand the network services in AIX and the impact each one has on system security. Administrators responsible for RS/6000s connected in some way to a public network can use the information in this tutorial to achieve the necessary balance between functionality and security.

Section 1. Before you start

About this tutorial

This tutorial is for AIX systems administrators who want to better understand the network services in AIX and the impact each one has on system security. Administrators responsible for RS/6000s connected in some way to a public network can use the information in this tutorial to achieve the necessary balance between functionality and security.

No third-party tools are used in this presentation; only the components available to all AIX systems are explored and addressed. While a true security model takes much more than just turning off services and modifying configuration files, this tutorial provides a solid foundation to build upon to reach the goal of complete system integrity.

About the examples in this tutorial

The examples in this tutorial were run on an IBM RS/6000 that had a complete installation of AIX 4.3.3, Maintenance Level 08. No software beyond that available on the AIX installation media was placed on the host. All practices and configurations discussed in this tutorial are equally applicable (and have been

confirmed) on a production system running AIX 5L (5.1 ML 01).

It is important to understand the potential impact of any change made to a system's configuration; this is especially true when dealing with security-related concerns. Before making any modifications to a production system, be sure that the changes have first been tested in a suitable development environment. Always back up systems, wear a seat belt, and close the cover when striking.

Section 2. Understanding security

What is the problem?

A normal installation of AIX (or almost any operating system) includes a dizzying array of services. Some of these services are critical: without a `telnetd` daemon active, for example, there would be no way to remotely log in to the system. Many of these services, though, were developed at a time when the Internet was much smaller, and the perceived danger from crackers and other people trying to gain unauthorized access was considered less than the benefits of easy remote access and simple authentication methods.

Today's computing environment is a much more dangerous place: the number of hosts connected to the Internet has grown exponentially, and with that growth, attempts to access and subvert computers have become commonplace events. Insufficiently secure systems can be quickly compromised; once an attacker has gained access to a system, information stored there is no longer private, and its contents cannot be trusted.

How does it happen?

Any system compromised by remote attackers has had, by definition, its own network connection used against it. Some of these attacks take advantage of bugs in a particular version of a network daemon. Other attacks are successful because they exploit a known weakness in a particular protocol, common to all systems running that service.

It is important to evaluate all active network services for not only their usefulness in fulfilling necessary tasks, but also for their shortcomings or vulnerabilities. Balancing the good and bad in each service can be difficult, but it is a critical part of keeping a system safe.

Barbarians at the sockets

A default AIX installation offers numerous services, each responsible for listening on at least one port. The table below details each open port, the system daemon that is bound to the port, and the configuration file that starts up the daemon.

Port Number	*Protocol*	*Well-Known Name*	*Daemon/Application*	*Started From*
7	tcp	echo	/usr/sbin/inetd	/etc/inetd.conf
7	udp	echo	/usr/sbin/inetd	/etc/inetd.conf
9	tcp	discard	/usr/sbin/inetd	/etc/inetd.conf
9	udp	discard	/usr/sbin/inetd	/etc/inetd.conf
13	tcp	daytime	/usr/sbin/inetd	/etc/inetd.conf
13	udp	daytime	/usr/sbin/inetd	/etc/inetd.conf
19	tcp	chargen	/usr/sbin/inetd	/etc/inetd.conf
19	udp	chargen	/usr/sbin/inetd	/etc/inetd.conf
21	tcp	ftp	/usr/sbin/ftpd	/etc/inetd.conf
23	tcp	telnet	/usr/sbin/telnetd	/etc/inetd.conf
25	tcp	smtp	/usr/sbin/sendmail	/etc/rc.tcpip
37	tcp	time	/usr/sbin/inetd	/etc/inetd.conf
37	udp	time	/usr/sbin/inetd	/etc/inetd.conf
67	udp	bootps	/usr/sbin/bootpd	/etc/inetd.conf
111	tcp	sunrpc	/usr/sbin/portmap	/etc/rc.tcpip
111	udp	sunrpc	/usr/sbin/portmap	/etc/rc.tcpip
161	udp	snmp	/usr/sbin/snmpd	/etc/rc.tcpip
177	udp	xdmcp	/usr/dt/bin/dtlogin (spawned by /etc/rc.dt)	/etc/inittab
199	tcp	smux	/usr/sbin/dpid2	/etc/rc.tcpip
512	tcp	exec	/usr/sbin/rexecd	/etc/inetd.conf
513	tcp	login	/usr/sbin/rlogind	/etc/inetd.conf
514	tcp	shell	/usr/sbin/rshd	/etc/inetd.conf
514	udp	syslog	/usr/sbin/syslogd	/etc/rc.tcpip
518	udp	ntalk	/usr/sbin/talkd	/etc/inetd.conf
543	tcp	klogin	/usr/sbin/krlogind	/etc/inetd.conf
544	tcp	kshell	/usr/sbin/krshd	/etc/inetd.conf
1001	tcp	rpc.statd	/usr/sbin/rpc.statd	/etc/rc.nfs
1001	udp	rpc.statd	/usr/sbin/rpc.statd	/etc/rc.nfs
1002	tcp	rpc.statd	/usr/sbin/rpc.statd	/etc/rc.nfs
1002	udp	rpc.statd	/usr/sbin/rpc.statd	/etc/rc.nfs
1234	tcp	instsrv	/home/netinstall/bin/instsrv	/etc/inetd.conf
2401	tcp	writesrv	/usr/sbin/writesrv	/etc/inittab

6000	tcp	X11	/usr/lpp/X11/bin/X /etc/inittab (spawned by /etc/rc.dt)
6112	tcp	dtspc	/usr/dt/bin/dtspcd /etc/inetd.conf
32768	tcp	dtlogin	/usr/dt/bin/dtlogin /etc/inittab (spawned by /etc/rc.dt)
32769	tcp	rpc.ttdbserver	/usr/dt/bin/rpc.ttdbserver /etc/inetd.conf
32772	tcp	dpid2	/usr/sbin/dpid2 /etc/rc.tcpip
32785	udp	cmsd	/usr/dt/bin/rpc.cmsd /etc/inetd.conf
49213	tcp	httpd	/usr/IMNSearch/httpdlite/bin/httpdlite /etc/inittab

What can be done?

There are two basic steps in hardening the networking services of an AIX system:

1. Disable unnecessary services.
   Should a system whose purpose is to serve Web sites also have a mail server running on it? If the ability to boot a system from a network source will not be used in a data center, should the daemons that provide that service be active? The answer to both of those questions is, obviously, no. If a network service is running, but there is no use for it, then it should be turned off.

2. Configure remaining services for secure operation.
   Of course, it is not possible to turn off every service. (If it is possible, then an organization needs to rethink its use of information technologies.) It is possible, however, to mitigate potential risks by configuring those active services to operate in a secure fashion. In a perfect world, default configurations would also be secure configurations. Sadly, the world is not perfect.

Section 3. Disabling unnecessary services

Where are network services started?

On an AIX system, network services are typically started in one of four ways:

1. An entry in the `/etc/inittab file`. These services are activated at system startup, and depending on the method used, may automatically respawn if killed.

2. An entry in the `/etc/rc.tcpip` file. This shell script is executed during system startup via an entry in `/etc/inittab`, and is responsible for starting the bulk of the standard network daemons.

3. An entry in the `/etc/inetd.conf` file. This configuration file for the `inetd` daemon (the super server) contains numerous entries for services that are useful and possibly necessary, but also many of dubious value in a secured environment.

4. Manually, by the invocation of a command by a user. Programs that bind to a port number above 1024 and accept connections from the network do not have to be run by the root user. Any user who has the ability to log in to a system also has the ability to run their own network services.

Each of the above methods must be checked for insecure or unnecessary services if the system is to be protected from network-originating attacks.

An overview of /etc/inittab

The following listing is an excerpt from the default `/etc/inittab`.

```
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
powerfail::powerfail:/etc/rc.powerfail 2>&1 | alog -tboot > /dev/console # Power Failure
Detection
mkatmpvc:2:once:/usr/sbin/mkatmpvc >/dev/console 2>&1
atmsvcd:2:once:/usr/sbin/atmsvcd >/dev/console 2>&1
load64bit:2:wait:/etc/methods/cfg64 >/dev/console 2>&1 # Enable 64-bit execs
rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
fbcheck:2:wait:/usr/sbin/fbcheck 2>&1 | alog -tboot > /dev/console # run /etc/firstboot
srcmstr:2:respawn:/usr/sbin/srcmstr # System Resource Controller
rcnetw:2:wait:/etc/rc.netware #start Netware
cnsview:2:wait:/usr/bin/cnsview -c "daemon start" >/dev/console 2>&1 # Start cnsview daemon
rctcpip:2:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
rcnfs:2:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
cron:2:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pio/etc/pioinit >/dev/null 2>&1  # pb cleanup
qdaemon:2:wait:/usr/bin/startsrc -sqdaemon
writesrv:2:wait:/usr/bin/startsrc -swritesrv
uprintfd:2:respawn:/usr/sbin/uprintfd
logsymp:2:once:/usr/lib/ras/logsymptom # for system dumps
httpdlite:2:once:/usr/IMNSearch/httpdlite/httpdlite -r
/etc/IMNSearch/httpdlite/httpdlite.conf \
                & >/dev/console 2>&1
diagd:2:once:/usr/lpp/diagnostics/bin/diagd >/dev/console 2>&1
imnss:2:once:/usr/IMNSearch/bin/imnss -start imnhelp >/dev/console 2>&1
imqss:2:once:/usr/IMNSearch/bin/imq_start >/dev/console 2>&1
pmd:2:wait:/usr/bin/pmd > /dev/console 2>&1 # Start PM daemon
dt:2:wait:/etc/rc.dt
cons:0123456789:respawn:/usr/sbin/getty /dev/console
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
```

```
l6:6:wait:/etc/rc.d/rc 6
l7:7:wait:/etc/rc.d/rc 7
l8:8:wait:/etc/rc.d/rc 8
l9:9:wait:/etc/rc.d/rc 9
```

Unlike most configuration files in AIX, entries in `/etc/inittab` *cannot* be disabled by placing a pound sign (#) at the beginning of the line. To disable an `inittab` entry, it must either be deleted from the file, have a colon (:) as the first character of the line, or have the value of the action field set to Off. Changes to `/etc/inittab` may also be made via the `chitab` command; entries may be deleted with the `rmitab` command.

Before making changes to this file, ensure that there is a good backup copy available, as problems with the `inittab` file can prevent a system from starting.

Editing /etc/inittab entries

- `rcnetw:2:wait:/etc/rc.netware #start Netware`
  The `rcnetw` entry executes the `/etc/rc.netware` script, which in turn, loads drivers for the Novell NetWare networking protocols. If NetWare will not be used with this system, it is safe to disable or delete this entry.

  `rctcpip:2:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons`
  The majority of the stand-alone network daemons are started from within the `/etc/rc.tcpip` shell script, which is executed at startup via this `inittab` entry. The contents of this script and the services started by it will be discussed in detail further in this tutorial.

- `rcnfs:2:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons`
  NIS/YP and NFS services are started by the `/etc/rc.nfs` script. However, the lines that start up NIS are commented out by default, and unless NIS is in use , they should remain disabled.

  The `biod` daemon is required on systems that are either mounting (as a client) or exporting (as a server) filesystems via NFS.

  `rpc.statd` and `rpc.lockd` provide locking and crash recovery used by some applications. Both of these daemons have had security problems resulting in the compromise of a system. Though the versions of `rpc.statd` and `rpc.lockd` that come with the latest releases of AIX are not known to be vulnerable, these daemons are often unnecessary, and can usually be disabled with no ill effect on the use of NFS-mounted filesystems.

  NFS server capabilities are started if two conditions are met:

1. The program `/usr/sbin/nfsd` exists and is executable.
2. The file `/etc/exports` exists.

If the system is not going to provide NFS services, ensure that at least one of the two conditions listed will not be met, by either removing the executable bit from the `nfsd` daemon or by changing its name. If the system will not be using NIS or NFS at all, it is safe to disable or delete the entry for `rcnfs` in `/etc/inittab`.

- `writesrv:2:wait:/usr/bin/startsrc -swritesrv`
  The `writesrv` entry enables the ability of users on remote machines to use the `write` command to send messages to users on the local system. This entry should be disabled or deleted on any system that is connected to a public network. Note that disabling the `writesrv` service does not prevent users from messaging one another on the local system.

- `httpdlite:2:once:/usr/IMNSearch/httpdlite/httpdlite -r /etc/IMNSearch/httpdlite/httpdlite.conf & >/dev/console 2>&1`
- `imnss:2:once:/usr/IMNSearch/bin/imnss -start imnhelp >/dev/console 2>&1`
- `imqss:2:once:/usr/IMNSearch/bin/imq_start >/dev/console 2>&1`
  The above three `inittab` entries comprise the AIX Documentation Search Service, providing a Web site interface for browsing and searching AIX documentation. It is not necessary to run this service on all the systems in a network, and the Web server that IBM provides with this server is fairly limited in its features. These entries should be disabled or deleted from `/etc/inittab`.

- `dt:2:wait:/etc/rc.dt`
  The `/etc/rc.dt` program starts the CDE graphical window server; aside from using up valuable resources, the Common Desktop Environment has been a source of many security vulnerabilities in the past. Though there are no known issues with the latest version, unless there is a specific need for CDE, this entry should be disabled or deleted.

- `l2:2:wait:/etc/rc.d/rc 2`
- `l3:3:wait:/etc/rc.d/rc 3`
- `l4:4:wait:/etc/rc.d/rc 4`
- `l5:5:wait:/etc/rc.d/rc 5`
- `l6:6:wait:/etc/rc.d/rc 6`

- `l7:7:wait:/etc/rc.d/rc 7`
- `l8:8:wait:/etc/rc.d/rc 8`
- `l9:9:wait:/etc/rc.d/rc 9`

The above `inittab` entries were added by IBM to AIX 4.3.3 in an attempt to make AIX more friendly to administrators of Linux and Solaris systems. They provide a set of directories that application startup scripts can be placed in and be invoked upon entering a given run level, a la System-V-based Unix systems. While there are no services started by these entries by default on AIX 4.3.3, system administrators must be aware that they exist and might be used accidentally by other people with root access on the system.

Unless there are plans to use this System-V adaptation to the AIX startup process, these entries should be disabled or deleted from `inittab` and the directory `/etc/rc.d` should be deleted or renamed to prevent misunderstandings.

An overview of /etc/rc.tcpip

The following excerpt from the `/etc/rc.tcpip` script indicates the services that are started or disabled in a default AIX installation.

```
echo "Starting tcpip daemons:"
trap 'echo "Finished starting tcpip daemons."' 0

# Start up dhcpcd daemon
#start /usr/sbin/dhcpcd "$src_running"

# Start up autoconf6 process
#start /usr/sbin/autoconf6 ""

# Start up ndpd-host daemon
#start /usr/sbin/ndpd-host "$src_running"

# Start up the ndpd-router daemon
#start /usr/sbin/ndpd-router "$src_running"

# Start up syslog daemon (for error and event logging)
start /usr/sbin/syslogd "$src_running"

# Start up print daemon
#start /usr/sbin/lpd "$src_running"

# Start up routing daemon (only start ONE)
#start /usr/sbin/routed "$src_running" -q
#start /usr/sbin/gated "$src_running"

# Start up the sendmail daemon.

qpi=30m  # 30 minute interval

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

# Start up Portmapper
start /usr/sbin/portmap "$src_running"

# Start up socket-based daemons
start /usr/sbin/inetd "$src_running"

# Start up Domain Name daemon
```

```
#start /usr/sbin/named "$src_running"

# Start up time daemon
#start /usr/sbin/timed "$src_running"

# Start up Network Time Protocol (NTP) daemon
#start /usr/sbin/xntpd "$src_running"

# Start up rwhod daemon (a time waster)
#start /usr/sbin/rwhod "$src_running"

# Start up the Simple Network Management Protocol (SNMP) daemon
start /usr/sbin/snmpd "$src_running"

# Start up the DHCP Server
#start /usr/sbin/dhcpsd "$src_running"

# Start up the DHCP Relay Agent
#start /usr/sbin/dhcprd "$src_running"

# Start up the DPID2 daemon
start /usr/sbin/dpid2 "$src_running"

# Start up the mrouted daemon
#start /usr/sbin/mrouted "$src_running"

# Start up the atm subagnet daemon muxatmd
#start /usr/sbin/muxatmd "$src_running"
/usr/lpp/x_st_mgr/bin/x_st_mgrd -b /usr/lpp/x_st_mgr/bin/x_st_mgrd.cf -s x_st_mgrd
```

Since `/etc/rc.tcpip` is a Korn shell script, items may be disabled by placing a pound sign (#) at the beginning of the line, or by deleting the line entirely.

Configuring /etc/rc.tcpip

- `start /usr/sbin/syslogd "$src_running"`
  The `syslog` daemon provides a facility for handling messages and errors from various applications running on a system. `Syslogd` is started by default on AIX; this is a good thing. Unfortunately, the default configuration does not log any messages received by the daemon. Configuring the `syslog` daemon will be discussed later in this tutorial.

- `start /usr/lib/sendmail "$src_running" "-bd -q${qpi}"`
  Sendmail is started in daemon mode by the `/etc/rc.tcpip` script; in this default configuration, the AIX system provides SMTP service with no restriction or authentication. Unless the systems administrator rectifies this unfortunate fact, the system will be used as a relay, routing tremendous amounts of spam to users throughout the Internet and incurring the wrath of the recipients of said mail. Running sendmail in a secure fashion will be discussed later in this tutorial. Until the reader is ready to address the issue of proper configuration, sendmail should be disabled by commenting out this line in `rc.tcpip`.

- `start /usr/sbin/portmap "$src_running"`
  The `portmap` daemon is responsible for reporting the port numbers in use by all Remote Procedure Call (RPC) servers running on the system.

Some common RPC servers include those involved in NFS (both client and server), and a number of items started by the `inetd` daemon, including `rstatd`, `rexd`, and other items of dubious value and high risk for security vulnerabilities. If after reviewing the active services in use on the system, the administrator determines that there are none that use RPC, this daemon should be commented out or deleted from the `rc.tcpip` script.

- `start /usr/sbin/inetd "$src_running"`
  The `inetd` server is the super server, responsible for listening on multiple ports and responding to requests for network services by spawning the appropriate program, as specified in the configuration file `/etc/inetd.conf`. A large number of the services that `inetd` provides in its default configuration should not be run on a system connected to a public network because of either weaknesses in the authentication and authorization methods used by the service, or because a history of security problems in the application providing the service. Proper configuration of `inetd` is addressed later in this tutorial. If, after reviewing the services provided by `inetd`, the administrator determines that there are better replacements for the items that are required for the system to function, the `inetd` daemon can be disabled by commenting out or deleting this entry from the `rc.tcpip` script.

- `start /usr/sbin/snmpd "$src_running"`
- `start /usr/sbin/dpid2 "$src_running"`
  The `snmpd` and `dpid2` daemons both provide Simple Network Management Protocol (SNMP) services, which can be useful for monitoring the status of the system, but which were not designed with a good security model and have multiple known vulnerabilities. Unless the system is secured behind a firewall that restricts SNMP requests to the local network, both of these services should be disabled by commenting out or deleting their entries in `rc.tcpip`.

  The `dpid2` daemon, in particular, should always be disabled, as it is an archaic holdover from earlier SNMP developments, and no longer provides any useful service.

- `/usr/lpp/x_st_mgr/bin/x_st_mgrd -b /usr/lpp/x_st_mgr/bin/x_st_mgrd.cf -s x_st_mgrd`
  The above command is added to the `/etc/rc.tcpip` script by the installation of the fileset `X11.x_st_mgr.rte`, and it provides services to IBM Xstation graphical workstations. Unless the RS/6000 is providing these services to Xstations, this entry should be deleted from the `rc.tcpip` file, and the `X11.x_st_mgr.rte` fileset should be removed from the system.

An overview of /etc/inetd.conf

The following excerpt from `/etc/inetd.conf` lists all of the services that are active in the default installation of AIX:

```
ftp     stream  tcp6    nowait  root    /usr/sbin/ftpd         ftpd
telnet stream  tcp6    nowait  root    /usr/sbin/telnetd      telnetd -a
shell stream  tcp6    nowait  root    /usr/sbin/rshd         rshd
kshell stream  tcp     nowait  root    /usr/sbin/krshd        krshd
login stream  tcp6    nowait  root    /usr/sbin/rlogind      rlogind
klogin stream  tcp     nowait  root    /usr/sbin/krlogind     krlogind
exec stream  tcp6    nowait  root    /usr/sbin/rexecd       rexecd
bootps  dgram   udp     wait    root    /usr/sbin/bootpd       bootpd /etc/bootptab
tftp     dgram  udp6    SRC     nobody  /usr/sbin/tftpd         tftpd -n
ntalk   dgram   udp     wait    root    /usr/sbin/talkd         talkd
rstatd   sunrpc_udp     udp     wait    root    /usr/sbin/rpc.rstatd rstatd 100001 1-3
rusersd sunrpc_udp      udp     wait    root    /usr/lib/netsvc/rusers/rpc.rusersd rusersd
100002 1-2
rwalld   sunrpc_udp     udp     wait    root    /usr/lib/netsvc/rwall/rpc.rwalld rwalld
100008 1
sprayd   sunrpc_udp     udp     wait    root    /usr/lib/netsvc/spray/rpc.sprayd sprayd
100012 1
pcnfsd   sunrpc_udp     udp     wait    root    /usr/sbin/rpc.pcnfsd pcnfsd 150001 1-2
echo    stream  tcp     nowait  root    internal
discard stream  tcp     nowait  root    internal
chargen stream  tcp     nowait  root    internal
daytime stream  tcp     nowait  root    internal
time    stream  tcp     nowait  root    internal
echo    dgram   udp     wait    root    internal
discard dgram   udp     wait    root    internal
chargen dgram   udp     wait    root    internal
daytime dgram   udp     wait    root    internal
time    dgram   udp     wait    root    internal
ttdbserver      sunrpc_tcp      tcp     wait    root    /usr/dt/bin/rpc.ttdbserver
rpc.ttdbserver 100083 1
ssalld  sunrpc_tcp      tcp     wait    root    /usr/sbin/rpc.ssalld rpc.ssalld 300667 1
instsrv stream  tcp     nowait  netinst /home/netinst/bin/instsrv instsrv -r
      /tmp/netinstalllog /home/netinst/scripts
dtspc   stream  tcp     nowait  root    /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd
cmsd    sunrpc_udp      udp     wait    root    /usr/dt/bin/rpc.cmsd cmsd 100068 2-5
```

To disable a service listed in the `/etc/inetd.conf` file, delete the line or place a pound sign (#) as the first character of the line. Changes to `inetd.conf` will not take effect until the `inetd` daemon is restarted by either sending a HUP signal to the PID of the process or by executing the command **`refresh -s inetd`** as root.

Configuring /etc/inetd.conf

- `ftp stream tcp6 nowait root /usr/sbin/ftpd ftpd`
  This entry provides the FTP (File Transfer Protocol) service, enabling the uploading and downloading of files by an FTP client. In general, FTP is not considered a secure protocol as it transmits user IDs, passwords, and data in the clear, with no encryption. Secure alternatives, such as `sftp-server` included with OpenSSH, are available, though they require that different client programs be used.

  If a more secure alternative is acceptable, or if there is no need to provide

download and upload capabilities for the system, the FTP server should be disabled by deleting or commenting out its entry in the `/etc/inetd.conf` file. If FTP service is required, improving the security of the service is discussed later in this tutorial.

- `telnet stream tcp6 nowait root /usr/sbin/telnetd telnetd -a`
  The telnet service gives users the ability to log in to the system remotely, using a standard telnet client. As it is with the FTP service, the telnet protocol is inherently insecure due to its passing of user IDs, passwords and data without encryption. A common, secure alternative is the `sshd` component of the OpenSSH application package. If there is a need for the providing of the telnet service, some level of security can be achieved via the techniques discussed later in this tutorial.

- `shell stream tcp6 nowait root /usr/sbin/rshd rshd`
- `kshell stream tcp nowait root /usr/sbin/krshd krshd`
  The `shell` and `kshell` entries both provide the services for the `rcp` and `rsh` client programs. There is a large difference, though, in the security of the services provided by the different daemons. The `kshell` service, provided by the `krshd` daemon, uses Kerberos to verify the identity of the client and to authenticate the user. The `shell` service, however, uses an extremely weak method of authentication, and should not be enabled unless the system is not connected to a public network.

  If an organization uses Kerberos, disable or delete the entry for `shell`, as its availability negates any benefit gained by the deployment of Kerberos. If Kerberos is not in use, the `sshd` component of the OpenSSH application package provides a much more secure drop-in replacement for the rshd daemon; in this case, both entries should be disabled.

- `login stream tcp6 nowait root /usr/sbin/rlogind rlogind`
- `klogin stream tcp nowait root /usr/sbin/krlogind krlogind`
  As with the `shell/kshell` services, the `login` and `klogin` entries both provide a similar service: the ability for remote users to log in to the system without specifying a password. Again, the `krlogind` daemon uses Kerberos for authentication and encryption, while the standard `rlogind` uses the same insecure method for user authentication as the shell service, with no encryption of data passed over the network.

  The `login` service should be disabled or deleted from `inetd.conf`, as the Kerberized `klogin` service is much more secure. If Kerberos is not in use at a Web site, disable both services and use ssh.

- `exec stream tcp6 nowait root /usr/sbin/rexecd rexecd`
  The `exec` service is yet another holdover from the time when networks were friendly, and security was not a prime concern. The `rexecd` daemon gives remote users the ability to run commands on the system, with poor authentication and no encryption of passwords or data. This entry should be disabled, as there are numerous secure alternatives for this service.

- `bootps dgram udp wait root /usr/sbin/bootpd bootpd /etc/bootptab`
- `tftp dgram udp6 SRC nobody /usr/sbin/tftpd tftpd -n`
  The `bootpd` and `tftpd` daemons make available the ability to do network booting of remote systems. Unless it is required that the system provide this service, the `bootps` and `tftp inetd.conf` entries should be disabled or deleted.

- `ntalk dgram udp wait root /usr/sbin/talkd talkd`
  Similar in purpose to the `writesrv` service started from `/etc/inittab`, the `talkd` daemon receives messages from users on remote systems, displaying them on the terminals of local users. This daemon has been successfully exploited in the past, and should be disabled.

- `rstatd sunrpc_udp udp wait root /usr/sbin/rpc.rstatd rstatd 100001 1-3`
- `rusersd sunrpc_udp udp wait root /usr/lib/netsvc/rusers/rpc.rusersd rusersd 100002 1-2`
  The `rstatd` and `rusersd` services provide information about the system and the users; this information should not be available outside of the local network. They should both be disabled if the system is connected to a public network.

- `rwalld sunrpc_udp udp wait root /usr/lib/netsvc/rwall/rpc.rwalld rwalld 100008 1`
  The `rwalld` daemon accepts incoming messages and writes the message to the terminals of all logged-in users. This `inetd.conf` entry should be disabled or deleted.

- `sprayd sunrpc_udp udp wait root /usr/lib/netsvc/spray/rpc.sprayd sprayd 100012 1`
  The `sprayd` daemon, in conjunction with the `spray` command, can provide network performance statistics. It can also provide a platform for denial-of-service attacks. It should be disabled or deleted.

- `pcnfsd sunrpc_udp udp wait root /usr/sbin/rpc.pcnfsd pcnfsd 150001 1-2`

The `rpc.pcnfsd` daemon receives connections from Personal Computer-Network File System (PC-NFS) clients. This service primarily used for printing from desktop computers with non-Unix-based operating systems to print spools on the AIX server. There have been several vulnerabilities in this daemon, and there are more modern, secure alternatives.

Unless the environment requires the use of PC-NFS, this `inetd.conf` entry should be disabled.

- `echo stream tcp nowait root internal`
- `discard stream tcp nowait root internal`
- `chargen stream tcp nowait root internal`
- `daytime stream tcp nowait root internal`
- `time stream tcp nowait root internal`
- `echo dgram udp wait root internal`
- `discard dgram udp wait root internal`
- `chargen dgram udp wait root internal`
- `daytime dgram udp wait root internal`
- `time dgram udp wait root internal`
  The above services are handled internally by the `inetd` daemon. While none of the internal services have been the source of a large vulnerability, each entry represents another open port and an opportunity to host a denial-of-service attack. Therefore, disable these entries unless there is a specific need for the service that they provide.

- `ttdbserver sunrpc_tcp tcp wait root /usr/dt/bin/rpc.ttdbserver rpc.ttdbserver 100083 1`
  The ToolTalk Database Server (`rpc.ttbdserver`) is a component of the Common Desktop Environment (CDE), used by AIXWindows. This daemon has a history of security vulnerabilities, and it should be disabled, as it is unlikely that a server would require the availability of CDE.

- `ssalld sunrpc_tcp tcp wait root /usr/sbin/rpc.ssalld rpc.ssalld 300667 1`
  The `rpc.ssalld` daemon is the part of the SSA network agent for the IBM StorWatch Serial Storage Expert (StorX) application. Unless StorX is in use and is monitoring this host, the `ssalld` entry should be disabled or deleted from `/etc/inetd.conf`.

- `instsrv stream tcp nowait netinst /home/netinst/bin/instsrv instsrv -r`

`/tmp/netinstalllog /home/netinst/scripts`
The `instsrv` service is a part of the Network Installation Tools, useful only for providing service to RS/6000s running AIX version 3.2 or earlier. This `inetd.conf` entry should be disabled, and the fileset that it is associated with, `boos.compt.Netinst`, should be removed from the system.

- `dtspc stream tcp nowait root /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd`
- `cmsd sunrpc_udp udp wait root /usr/dt/bin/rpc.cmsd cmsd 100068 2-5`
  Both the `dtspcd` daemon and the `rpc.cmsd` daemon are part of the Common Desktop Environment. `dtspc` enables the launching of applications from remote hosts. The `cmsd` service communicates with the `dtcm` client calendaring application. Neither of these services should be active.

Section 4. Securing remaining services

The syslogd daemon

The key to ensuring that a system remains secure is by constantly monitoring messages posted by the various system daemons and other programs. Most of these programs report errors and other messages through the `syslogd` daemon, which is responsible for dispatching these messages based on their severity and their facility (or source). While the `syslogd` daemon is active by default in AIX, the configuration of `syslogd` does no logging of any messages it receives, silently discarding them.

Syslogd: facilities, priorities, and destinations

The `/etc/syslog.conf` configuration file determines how messages from a particular source and of a set priority will be dealt with. Every system program that uses `syslogd` will transmit messages at a predefined facility. Those facilities are:

`kern`	kernel messages
`user`	various user-level programs
`mail`	sendmail
`daemon`	system daemons, including ftpd
`auth`	authorization messages

`authpriv`	authorization messages whose viewing should be restricted to root
`syslog`	messages generated internally by the syslog daemon
`lpr`	lpd (printer subsystem)
`news`	nntp (news) server messages
`uucp`	uucp subsystem messages
`cron`	crond messages
`local0 - local7`	facilities available for administrator-defined use

Each message sent to `syslogd` also has a severity or priority attached to it. Those priorities, in order of severity from highest to lowest, are:

`emerg`	daemon or subsystem failure has occurred or is pending
`alert`	immediate action is required to prevent failure
`crit`	a critical condition has occurred
`err`	an error has occurred
`warning`	a warning has occurred
`notice`	a normal, but significant event has occurred
`info`	informational messages
`debug`	debug-level messages

The `syslogd` daemon dispatches the message to a given destination, based upon the configuration in the `/etc/syslogd.conf` configuration file. Destinations may be:

`file name`	records the message in the log file specified
`@hostname`	transmits the message to the syslogd daemon running on the specified hostname
`user`	writes the message to the terminal of the specified user name
`*`	writes the message to the terminals of all logged-in users
`errlog`	transmits the message to the AIX error logging facility (errdemon) for inclusion in the errpt

Configuration of the syslog.conf file

There is no single correct way to configure the `syslogd` daemon; the administrator needs to determine a site's configuration based on the standard practices used by their organization. The following example, however, will provide a good starting point.

```
#########################################################################
## /etc/syslog.conf                                                    ##
#########################################################################


# record messages from all facilities at severity "alert" or higher in
# the AIX errlog ...

*.alert                                          errlog

# record messages from all facilities at severity "err" or higher in
# the log file /var/adm/errorlog

*.err                                            /var/adm/errorlog

# record messages from the "mail" facility at severity "info" or
# higher in the log file /var/adm/mail.log

mail.info                                       /var/adm/mail.log

# record messages from the "auth" and "authpriv" facilities at
# severity "info" or higher in the log file /var/adm/auth.log

auth,authpriv.info                               /var/adm/auth.log

# record messages from the "daemon" facility at severity "info" or
# higher in the log file /var/adm/daemon.log

daemon.info                                      /var/adm/daemon.log


#########################################################################
```

The `syslogd` daemon will not write to a file if it does not already exist, so be sure to touch any log files that have been specified. In addition, ensure that the permissions on log files are set so that only authorized users can view their contents. After making changes to the `syslog.conf` file, the `syslogd` daemon must be restarted by typing **`refresh -s syslogd`** . It is important that the log files are pruned on a regular basis, or they will grow until the filesystem is filled up. Pruning or rotating of log files should be handled by a daily cleanup script.

Hardening sendmail

The sendmail mail transport program has a reputation of being the swiss cheese of software, filled with exploitable vulnerabilities. In fact, the first ever advisory released by the Computer Emergency Response Team (CERT) in 1997 dealt with a problem with sendmail and `ftpd`.

Fortunately, current versions of sendmail have a much better track record of security, and any issues that are found are dealt with promptly by the developers of the sendmail application and by IBM. AIX 4.3.3 ships with sendmail 8.9.3 (part of the `bos.net.tcp.client` fileset) and while this is not the latest available version, it is stable and relatively secure.

In order to fully secure the sendmail service, several changes to the default configuration must bemade.

Disabling the SMTP service

The first step in securing sendmail is to decide if it is even necessary to run sendmail on the particular host. Many organizations centralize their email service so that all incoming mail is routed to a small number of systems that are dedicated to receiving and routing incoming mail. If this is the case, then sendmail should not be running as a daemon on other hosts.

Sendmail is started from the file `/etc/rc.tcpip`; following the steps in this tutorial means automatic startup should already be disabled, by commenting out the line:

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

in `/etc/rc.tcpip`. Ensure that it is not running by issuing the command **`lssrc -s sendmail`** ; the output should be similar to:

```
Subsystem         Group        PID          Status
sendmail          mail                      inoperative
```

After disabling sendmail, the host will no longer accept connections on port 25. Emails generated from this system, though, will no longer be sent to their destination; they will remain in the spool directory until the mail queue is manually processed. The following lines should be added to root's `crontab` file:

```
# process the outgoing mail queue twice an hour.
10,40 * * * * /usr/lib/sendmail -q > /dev/null 2>&1
```

The above `cron` job will invoke sendmail twice an hour, at 10 minutes and 40 minutes past the hour, in `queue-processing` mode. Sendmail will not accept incoming network connections in this mode, but it will go through the mail queue and dispatch any messages it finds.

Closing the open relay

It is probably not feasible to disable incoming mail service on every system within an organization. If a system has been designated as a mail server, there are steps that can be taken to ensure that the system is not used to route mail not generated by or destined for an organization.

The default behavior of sendmail in AIX 4.3.3 permits what is known as *open relaying*. Basically, this means that the mail server will accept and process mail sent from outside an organization to addresses that are also outside thatorganization. Exploiting open relays is the most common technique used by spammers to send

email to thousands of addresses, all originating from a single mail server.

Before enabling sendmail in daemon mode (assuming it was disabled per the instructions earlier in this tutorial), the sendmail configuration file must be modified to prevent this misuse of a given system. The following steps detail the process of generating the updated `sendmail.cf` configuration file:

1. Ensure that the filesets `bos.net.tcp.adt` and `bos.adt.base` are installed.

2. Change the current working directory to `/usr/samples/tcpip/sendmail/cf`, and create a backup of the file `aix433.mc`.

3. Edit the file `aix433.mc`, making the changes detailed in the following table:

divert(0)dnl	This line should not be changed.
OSTYPE(aix43)dnl	This line contains an error; change the text within the parenthesis from `aix43` to `aix433`
FEATURE(genericstable)dnl	This line should be deleted unless the genericstable feature is required.
FEATURE(mailertable)dnl	This line should be deleted unless the mailertable feature is required.
FEATURE(virtualusertable)dnl	This line should be deleted unless the virtualusertable feature is required.
FEATURE(domaintable)dnl	This line should be deleted unless the domaintable feature is required.
FEATURE(allmasquerade)dnl	This line should be deleted unless the allmasquerade feature is required.
FEATURE(promiscuous_relay)dnl	Delete this line to disable open relaying.
FEATURE(accept_unresolvable_domains)dnl	Delete this line to increase security.
FEATURE(accept_unqualified_senders)dnl	Delete this line to increase security.
DOMAIN(generic)dnl	Without an organization-specific domain.m4 file, this line should not be changed.
MAILER(local)dnl	This line should not be changed.
MAILER(smtp)dnl	This line should not be changed.
MAILER(uucp)	This line should be deleted unless the routing of mail via UUCP is required.

After making the above changes and saving the file, the contents of the `aix433.mc` file should be similar to:

```
divert(0)dnl
OSTYPE(aix433)dnl
DOMAIN(generic)dnl
MAILER(local)dnl
MAILER(smtp)dnl
```

4. Build a new sendmail configuration file by executing the command:
   `m4 ../m4/cf.m4 aix433.mc > /tmp/sendmail.cf.new`

5. Often, several mail hosts will receive incoming mail for all of the other systems within the organization, through the use of mail exchange (MX) records in the domain name server configuration (DNS). If this is the case, create the file `/etc/sendmail.cw` and add to that file the hostname of each system for which this host will be processing incoming mail for. Note that if this host has multiple hostnames (due to aliases or multiple IP addresses), each of those names must be listed in this file as well, or mail sent to addresses at those hostnames will be bounced.
   If there are no hostnames in the `sendmail.cw` file, create the file anyway (leaving it empty), or edit the `/tmp/sendmail.cf.new` file and comment out with a pound sign (#) the line:

```
Fw/etc/sendmail.cw
```

6. If there are specific domains for which relaying needs to be permitted, create the directory `/etc/mail`, and create the file `/etc/mail/relay-domains`, adding the names of those domains to that file.
   If there are no domains for which relaying will be permitted, edit the `/tmp/sendmail.cf.new` file and comment out with a pound sign (#) the line:

```
FR-o /etc/mail/relay-domains
```

7. Confirm that there are no errors in the new sendmail configuration file, by running the command:

```
endmail -C/tmp/sendmail.cf.new -bt < /dev/null
echo $?
```

   If the sendmail command exited with a 0 (zero) code (reflected in the output of the `echo $?` command), then the new configuration file is ready to be used.

8. Make a backup of the file `/etc/sendmail.cf`, and copy the file `/tmp/sendmail.cf.new` to `/etc/sendmail.cf`.

9. Start sendmail in daemon mode by running the command:

```
startsrc -s sendmail -a "-bd -q30m"
```

10. Re-enable the starting of sendmail at system boot time by removing the sendmail entry commenting in `/etc/rc.tcpip`.

Sendmail will no longer allow the relaying of mail from hosts outside of a domain to addresses not in that domain.

Improving FTP security

Along with sendmail, the File Transfer Protocol (FTP) daemon is the most commonly exploited application on Unix servers. Unlike sendmail, though, vulnerabilities in the `ftpd` leading to complete system compromise have been found in the version included in AIX 4.3.3. For this reason (and others), unless it is absolutely necessary to offer FTP service, it should be completely disabled in `/etc/inetd.conf`.

If service requirements mandate the availablilty of FTP, there are a number of steps that can be taken that can reduce, if not eliminate, the risks inherent in this protocol.

1. Ensure that the AIX fix `IY04477` is fully installed on the system by running the command `instfix -ik IY04477`.
   This fix closes the vulnerability that has been widely reported, as noted in the IBM Emergency Response Service Security Vulnerability Alert number ERS-SVA-E01-1999:004.1. This APAR is included in Maintenance Level (ML) 02, and so should be installed on any system not currently at this ML level.

2. Ensure that the user names of those accounts that are *not* permitted to use the ftp service are listed in the file `/etc/ftpusers`. This list (with one user name per line) should include root, as well as all of the predefined administrative accounts: `daemon`, `bin`, `sys`, `adm`, `uucp`, `guest`, `nobody`, `lpd`, `invscout`, `imnadm`, `ipsec`, `nwroot`, `nwuser`, `nwprint`, `nwldap`, `ldap`, `nuucp`, and `netinst`.

3. Modify the entry for the ftpd server in `/etc/inetd.conf`, adding the following arguments:

```
ftp     stream  tcp6    nowait  root    /usr/sbin/ftpd   ftpd
                                 -l -u077
```

The `-l` enables logging by the ftpd daemon, at the priority `info`, while the `-u077` mask changes the permissions of uploaded files from the default of `027`.

Providing anonymous FTP service is highly advised against. However, if this service is required, use the shell script `/usr/samples/tcpip/anon.ftp` (installed as part of the `bos.net.tcp.client` fileset) to properly set up the anonymous FTP user account and directories.

Setting network options

Though not specific to a particular service, there are options that can be set to control how certain network protocols behave on an AIX system. These network options are displayed and set using the no command. Several of these network options should be changed from their default settings on any system that is connected to a public network (such as the Internet).

Network options need to be set at each system boot, as they do not "stick" across reboots. There is no official method of setting these options; the script below offers one possible method:

1. Create the following script and save it as `/etc/rc.no`.

```
#!/bin/ksh
          # ------------------------------------------------------------------------
          # /etc/rc.no : sets network options to improve performance and security
          # ------------------------------------------------------------------------

          echo "Setting network options"

          # protection against SYN flood attacks ...

          /usr/sbin/no -o clean_partial_conns=1

          # protection against ICMP redirects ...

          /usr/sbin/no -o ipignoreredirects=1

          # protection against illegal access via source routing ...

          /usr/sbin/no -o ipsendredirects=0
          /usr/sbin/no -o ipsrcroutesend=0
          /usr/sbin/no -o ipsrcrouteforward=0
          /usr/sbin/no -o ip6srcrouteforward=0
          /usr/sbin/no -o tcp_pmtu_discover=0
          /usr/sbin/no -o udp_pmtu_discover=0
```

2. Create an entry in `/etc/inittab` to run the above script directly after the `brc` entry, by manually editing the inittab file or by running the command `mkitab -i brc no:2:once:"/etc/rc.no > /dev/console 2>&1"`

3. Execute the `rc.no` command manually, so that its settings take effect for the current system environment.

Section 5. Summary

Reviewing the hardened system

Reviewing the hardened system

On the example system used for this tutorial, these changes have been made:

- In `/etc/inittab`:
  - The following entries were disabled: `rcnetw`, `writesrv`, `httpdlite`, `imnss`, `imqss`, `dt`, and lines l2 through l9 which referenced the System V script directories.
  - The `/etc/rc.nfs` script was modified to prevent the startup of the `rpc.statd` and `rpc.lockd` daemons.
  - An entry was added to execute the `/etc/rc.no` script, setting network options.
- In `/etc/rc.tcpip`:
  - The lines of the script responsible for starting the `snmpd`, `dpid2`, and `x_st_mgrd` daemons were disabled.
- In `/etc/inetd.conf`:
  - The following services were disabled: `shell`, `kshell`, `login`, `klogin`, `exec`, `bootps`, `tftp`, `rstatd`, `rusersd`, `rwalld`, `sprayd`, `pcnfsd`, `echo`, `discard`, `chargen`, `daytime`, `time`, `ttdbserver`, `ssalld`, `instsrv`, `dtspc`, and `cmsd`.
  - The entry for `ftp` was modified to invoke the ftp daemon with the arguments `-l -u077`.
- Sendmail configuration was modified to prevent the open relaying of third-party mail.

As a result of those changes, the number of open ports on the system was reduced

significantly:

Port Number	Protocol	Well-Known Name	Daemon/Application	Started from
21	tcp	ftp	/usr/sbin/ftpd	/etc/inetd.conf
23	tcp	telnet	/usr/sbin/telnetd	/etc/inetd.conf
25	tcp	smtp	/usr/sbin/sendmail	/etc/rc.tcpip
111	tcp	sunrpc	/usr/sbin/portmap	/etc/rc.tcpip
111	udp	sunrpc	/usr/sbin/portmap	/etc/rc.tcpip
514	udp	syslog	/usr/sbin/syslogd	/etc/rc.tcpip

Am I secure now?

Unfortunately, the answer to that question is "no". It is not enough to simply turn off some services and hope that what remains is secure. Constant monitoring of system logs and network services is required to guard against attempts at compromise and to check for signs of successful intrusion.

The steps outlined in this tutorial cover only what is possible with the tools provided by the operating system. No advantage was taken of the many open source and other free tools available. These software applications include OpenSSH, which provides secure replacements for so many of the insecure default services, and other tools like `nmap`, useful in determining the network footprint of a system and detecting unauthorized services that may be active.

Network security cannot be achieved by just keeping outsiders at bay. A truly safe system must have safeguards against illegitimate user activity, improper file access control, and a host of other issues. The steps outlined in this tutorial provide a necessary start to the never-ending process of developing a safe, secure, and productive computing environment.

A footnote on network security

Within the six days that the example system used in this tutorial was up and running, there were four distinct attempts at unauthorized access. System administrators who don't believe an attack is possible are not monitoring their logs close enough.

Resources

- Participate in the discussion forum for this content.
- The IBM RedBooks UNIX Portal provides a wealth of information on all aspects of AIX administration. Of particular interest to those looking to improve system security is:
  - SG24-5971 Additional AIX Security Tools on IBM (e)server pSeries, IBM RS/6000, and SP/Cluster
- SecurityFocus is an excellent community news site, focusing on computer security incidents, news, and features.
- The CERT Coordination Center, provided by Carnegie Mellon University, is research and development center, concentrating on Internet security vulnerabilities and publishing security alerts affecting all computing platforms.
- In conjunction with the Federal Bureau of Investigation (FBI), the SANS Institute also publishes The Twenty Most Critical Internet Security Vulnerabilities, updated regularly as new problems spring to the forefront.
- OpenSSH is an open source implementation of the Secure Shell standard suite. It provides secure replacements for telnet, ftp, rsh, rlogin, and rcp, as well as a method of encrypting other TCP/IP network protocols transparently. OpenSSH should be included in every Unix distribution by default.
- Nmap is a port scanner that can be used to determine the network footprint of a system, and detect services that shouldn't be running. Nmap is an important tool in the arsenal of the security-conscious administrator.
- Nessus is a remote network security auditing tool. By scanning the hosts with nessus, administrators will know all the vulnerable spots, hopefully before the bad guys do.

About the author

Sandor W. Sklar
Sandor W. Sklar is a Unix systems administrator at Stanford University, in beautiful Northern California. When not poking through his systems for real or imagined security holes, he enjoys spending time with his wife and two children.