

[close window](#)

e-Newsletter Exclusive

[Print](#) 

# TCP Wrappers Provide Robust Logs

Security tool is simple to install on AIX

May 2009 | by [Jaqui Lynch](#)

Many people don't realize most of the services listed in `/etc/inetd.conf` don't produce any log entries, thus making it difficult to know when someone is attempting to attack servers or gain unauthorized access, even if the attempt is rejected. By default, Linux installations use a program called TCP Wrappers to wrap network services so they can control access and get decent log records. This tool can also be installed on AIX to provide robust control and logging.

## Logging

Logs are a critical part of any security system—as they provide the audit trail of what has been happening. The syslog daemon (syslogd) starts up by default on AIX, but the log configuration file isn't set up to actually log anything. Additionally, in order to correlate logs, it's important to synchronize the time on all of your systems. Typically administrators set up network time protocol (NTP) to do this. Once NTP is running, the first step is to correctly set up logging in `/etc/syslog.conf`. I prefer to set up a separate file system for logs (e.g., `/usr/local/logs`), rather than using the default of `/var/spool`. If the default is used and `/var` fills up, the system may crash; but if a separate file system is used and that fills up, then the system will just stop logging. Although file-system usage should be monitored, it's still wise to store logs in their own file system to protect against large logs bringing down the system.

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, or any combination thereof. The most commonly used method is writing to a file in a file system. Once the file system is set up, the next step is to code a `/etc/syslog.conf` file. An example 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/maillog
daemon.info  /usr/local/logs/infolog
```

In the file above, all emergency, alert and error messages are written to the `/usr/local/logs/syslog` file. Sometimes messages will be sent to the console. Since most

customers use monitors on a switch, those messages tend to get missed. I usually send all my logs to a file, use scripts to scan them and e-mail or page people as necessary. In the above sample, authentication messages go to a separate log (authlog), mail messages to a separate maillog and daemon messages to infolog. daemon.info is where certain daemons, including TCP Wrappers, are configured to log to. This log separation makes it easier to search for patterns and problems. Once the syslog.conf file is coded, each log file being used needs to be created using the touch command—for example:

```
touch /usr/local/logs/syslog.
```

Although you can refresh the syslog daemon, this doesn't always cause syslogd to start using the new files. I tend to stop and start the syslog daemon as follows:

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

## TCP Wrappers

Once logging is correctly set up, it's time to install TCP Wrappers. To get the most out of TCP Wrappers, it's necessary to analyze the /etc/inetd.conf file and determine which services are needed. Additionally, /etc/rc.tcpip and /etc/inittab should also be analyzed and unnecessary services should be removed. A copy should be made of the files and then all unnecessary services should be deleted from the file—not commented out. If they're commented out, it's possible the next maintenance set will uncomment them, whereas it's rare a patch would add a service back. Any time maintenance is applied, the file should be checked to make sure nothing was added. Another alternative is to leave the services there, but to set them all to /bin/false so they would never execute successfully. In order to do this, /bin/false has to be added to the valid shells entry in /etc/security/login.cfg. Instead of:

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

Try:

```
ftp stream tcp6 nowait root /bin/false
```

Once this is done and the system has been rebooted, then it's time to move on to installing TCP Wrappers.

## Installing TCP Wrappers

The purpose of TCP Wrappers is to wrap a service, such as Telnet, so you can perform security checks before allowing or disallowing access to the service. This program is called by inetd before calling a service and 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. There are two ways to install the wrapper: One is to replace the current services and the second is to install the tcpd program into /usr/local/bin. Using the latter method ensures the wrapper is still there after maintenance. A working C compiler is needed to compile TCP Wrappers. The first step is to download the [program](#). This is the version that's IPV4 and IPV6 enabled. The file needs to be gunzipped and then untarred. Assuming we untar the files into /usr/local/soft, the install steps would be:

```
cd /usr/local/soft/tcp_wrappers_7.6-ipv6.4
vi Makefile
    /AIX - uncomment REAL_DAEMON_DIR line
    Uncomment STYLE=-DPROCESS_OPTIONS
    Change FACILITY=LOG_MAIL to LOG_DAEMON
    Check SEVERITY is LOG_INFO
    Uncomment IPV6=-DHAVE-IPV6
make aix
cp tcpd /usr/local/bin
cp tcpd.h /usr/local/include
```

Once that's done, the `inetd.conf` entry for a service such as `ftp` would be changed as follows:

```
OLD: ftp stream tcp6 nowait root /usr/sbin/ftpd -l ftpd
NEW: ftp stream tcp6 nowait root /usr/local/bin/tcpd /usr/sbin/ftpd -l
ftpd
```

The program `tcpd` has been inserted to execute before the `ftpd` program. It's also possible to configure a service to take advantage of the wrapper logging without being able to execute:

```
login stream tcp6 nowait root /usr/local/bin/tcpd /bin/false rlogind
```

In the line above, the wrapper will log attempts to use the service, but the service has been set to execute `/bin/false` rather than the real daemon. Because the wrapper logs for any service that calls it, this is a way to get log entries for attempted break-ins without allowing the service to run.

TCP Wrappers uses two files to control access. `/etc/hosts.deny` controls the denying of access, and `/etc/hosts.allow` controls the allowing of access. In order to keep things simple, it's best to put `all:all` in the `/etc/hosts.deny` file. This means all access to the listed services is denied unless it's explicitly allowed in the `hosts.allow` file—which can be configured to post a banner for each service, whether the service is granted or not. This is one way to ensure users see the acceptable use policy (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
```

You shouldn't disconnect from the system until you're certain the wrappers are working and you'll be able to log on.

It takes very little time to secure an AIX system's services. However, one of the biggest issues is the resistance to using open-source software. If a system isn't running TCP wrappers, then attempts to access it most likely aren't being logged. IBM provides a

precompiled version of TCP Wrappers on the expansion pack and that's certainly an option. I prefer to compile it myself, so I know exactly what options were chosen.

IBM Systems Magazine is a trademark of International Business Machines Corporation. The editorial content of IBM Systems Magazine is placed on this website by MSP TechMedia under license from International Business Machines Corporation.

©2009 MSP Communications, Inc. All rights reserved.

---