

Secure Shell 1 & 2 Recommended Configuration
Johns Hopkins Network & Information Security
June 2003

SSH - Secure Shell, is a replacement for common insecure remote access programs, telnet, rsh etc. With ssh everything you do remotely is encrypted and secure. With telnet, and other programs of the like, your passwords go through the network in plain text.

TCPWRAPPERS

Tcpwrappers allow you to restrict specific services by source host. It is useful because you can make changes without having to reboot the server or restart any services. You can use either name or ip address information. It is recommended that unless you trust your dns information to use the ip address rather than the name. If you decide to use name, you may want to specify in /etc/hosts the proper ip name translation information. This can protect you from dns spoofing. If you expect ip address information to be changing, you may want to use host names for convenience sake.

Tcpwrappers uses two files: /etc/hosts.allow and /etc/hosts.deny

Blanket deny ALL: ALL statement. /etc/hosts.allow overrides /etc/hosts.deny, so deny all and allow specifically. You can have email messages sent if access is attempted /etc/hosts.deny should look like this:

```
ALL:ALL: /usr/bin/mailx \  
-s "%s: connection attempt from %c" \  
user@domain.com
```

To allow ssh connection from xxx.xxx.xxx.xxx, /etc/hosts.allow would look like this:

```
sshd: xxx.xxx.xxx.xxx
```

You can add additional ip addresses to the same line, or you can make separate lines.

You can allow entire domains or subnets also:

```
sshd: domain.jhu.edu
```

```
sshd: 128.220.xxx.xxx
```

One thing to be careful for, is depending on how the sshd process is called it may be sshd2 instead of sshd. Please check your process list for confirmation. If it is sshd2, use that instead of sshd in the above lines.

Tcpwrappers need to be installed first. If you are running RedHat Linux, most likely tcpwrappers is already installed. You can find the source for many OS's by searching on <http://www.freshmeat.net/>. The source for Solaris can be found at <ftp.porcupine.org/pub/security>. You will need a compiler and gunzip to install tcpwrappers and ssh. Which can be found at <http://www.sunfreeware.com/> for Solaris. Ensure the compiler, /usr/ccs/bin, and the libraries are in you're path.

For Tcpwrappers to compile correctly, you will need to edit the Makefile for your operating system to compile correctly. You will need to edit the Makefile for your OS type. Uncomment the REAL_DAEMON_DIR line and you may need to add CC=gcc to the section about your OS.

make sys-type i.e.. sunos5

Once tcpwrappers has finished compiling copy libwrap.a to /usr/lib and tcpd.h to /usr/include.

Then obtain the source for ssh. SSH can be found at <ftp://ftp.ssh.org/pub/ssh/>.

Configure and install ssh with tcpwrappers support.

```
./configure -with-libwrap  
make  
make install
```

Edit /etc/ssh2/sshd2_config

Some important changes to make to the ssh server:

```
PermitRootLogin      no
```

```
AllowedAuthentications  password
```

There is a potential on some versions of SSH to be able to determine valid usernames. You can prevent this by setting the Password Guesses to 1.

```
# PasswordGuesses      3
```

There is also /etc/ssh2/ssh2_config. This file configures the client end of ssh. Defaults are mostly fine for this. You may need to change the port number or add ssh1 compatibility.

Once configured, create a script in /etc/init.d to start sshd

```
case "$1" in  
'start')  
    if [ -x /usr/local/sbin/sshd -f /etc/ssh2/sshd2_config ]; then  
        /usr/local/sbin/sshd  
    fi  
    ;;  
'stop')  
    kill 'cat /etc/ssh2/sshd_22.pid'  
    ;;  
*)  
    echo "Usage: $0 { start | stop }"  
    ;;  
esac  
exit 0
```

note: the name of the pid file will relate to the port number you are running the daemon on.

Change the mode of the file so it can be executed.

```
chmod 744 sshd
```

Next make a link to /etc/rc2.d/ so it will restart on boot up.

```
ln -s /etc/init.d/sshd /etc/rc2.d/S72sshd
```

*note: Depending on which runlevel your system starts at you may need to use a different rc.d directory. Linux systems typically run in level 3 for console access and 5 for GUI, where Solaris runs at level 2 for console and 5 for GUI.

This guide will cover the installation and setup of SSH 1.2.31

<ftp://ftp.ssh.org/pub/ssh/ssh-1.2.31.tar.gz>

```
zcat ssh-1.2.31.tar.gz | tar xf -
```

Now we are going to configure it, by running the following...

```
./configure --prefix=/usr/local --with-etcdir=/etc/ssh
```

If you get an error, like

configuring with X but xauth not found

type

```
whereis xauth
```

If it returns something like

```
/usr/X11R6/bin/xauth
```

type

```
export PATH=$PATH:/usr/X11R6/bin
```

run the configure step again.

Now compile and install ssh

```
make
```

```
make install
```

To configure sshd, the file is in /etc/ssh/ called sshd_config so go there and open it in a text editor.

SyslogFacility **AUTH**

This is where ssh will feed it's log to, the default is DAEMON which is usually sent to /var/log/messages. Change this to AUTH so it will log to your /var/log/authlog. Be sure to edit /etc/syslog.conf to add logging for auth.info, and that the /var/log/authlog file exists with correct permissions and ownership. When editing the syslog.conf, ensure the white spaces are tabs and that the columns line up with the existing lines.

Using Port Forwarding

You can use ssh to encrypt other services, either local or remote.

To secure a pop connection to a remote server, this is what you would do.

ssh -R 1110:pop.server:110 localhost sleep 3600

Then configure your pop client to point to localhost:1110 and you're all set.

One other thing to note.. if you are using tcpwrappers, be sure to add in permission for your localhost to talk to sshdfwd-1110. Or whichever port you are using, depending on the service. The first port number can be whatever you want it to be really, because you cannot forward low ports as a common user, it must be high. Just be sure to set the pop client and /etc/hosts.allow to point to the same port as the forwarding port.

You can shoot the process into the background with a CTRL-Z and bg, you can't use the & on the command line, because you need to enter your password first.