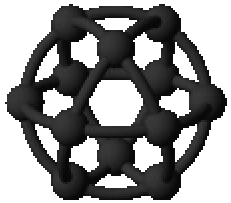
Secure Shell (SSH) and TCP Wrappers



Oct 5 2001

Jaqui Lynch
Circle4 Consulting
jaqui@circle4.com
http://www.circle4.com/jaqui/

Agenda

- General Security
- TCP Wrappers
  - Installation & Configuration
- Secure Shell (SSH)
  - Installation & Configuration
- References

General Security

- Ensure NTP is running and the time is correct
- Install sudo, lslk, lsof, cops, saint ….
- /etc/ftpusers
- .rhosts & .shosts
- /etc/hosts.equiv
- /etc/tftpaccess.ctl
- /etc/syslog.conf
- /etc/inetd.conf
- /etc/hosts.allow
- /etc/hosts.deny

/etc/syslog.conf

Both ssh and tcp wrappers make extensive use of logging

mail.debug	/usr/local/logs/mailog
*.emerg	/usr/local/logs/syslog
*.alert	/usr/local/logs/syslog
*.crit	/usr/local/logs/syslog
*.err	/usr/local/logs/syslog
auth.notice	/usr/local/logs/syslog
*.info	/usr/local/logs/messages
daemon.info	/usr/local/logs/infolog

Ensure you create each log by using touch
stopsrc –s syslogd
startsrc –s syslogd

TCP Wrappers

- Written by Wietse Venema
- www.porcupine.org
- Current Version is 7.6
  - With or without IPv6 support
- Wraps network services to provide logging, banners, and other options
- Wrappers improve security and logging
- Reverse dns lookup can be used to disallow access
- Allows tripwires
- Ensure system logging is up and working then install wrappers

TCP Wrappers Configuration

- vi Makefile
  - STYLE = -DPROCESS_OPTIONS # Enable language extensions.
  - FACILITY= LOG_DAEMON # LOG_MAIL is what most sendmail daemons use
  - SEVERITY= LOG_INFO
  - Causes tcpd to log everything to daemon.info
- make clean
- make aix
- cp tcpd /usr/local/bin
- cp tcpd.h to ssh source directories or to /usr/include
- cp libwrap.a /usr/local/lib
- vi inetd.conf, hosts.allow, hosts,deny
- refresh –s inetd

/etc/inetd.conf

ftp stream tcp6 nowait root /usr/local/bin/tcpd /usr/sbin/ftpd -u 002 -l ftpd

telnet stream tcp6 nowait root /usr/local/bin/tcpd /usr/sbin/telnetd telnetd –a

exec stream tcp6 nowait root /usr/local/bin/tcpd /usr/sbin/rexecd rexecd

dtspc stream tcp nowait root /usr/local/bin/tcpd /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd

xmquery dgram udp wait root /usr/local/bin/tcpd /usr/bin/xmservd xmservd -p3

rlogin stream tcp6 nowait root /usr/local/bin/tcpd /bin/false
netstat stream tcp nowait nobody /usr/local/bin/tcpd /bin/false

Delete everything else out of inetd.conf – don't just comment it out.
You should also check inetd.conf regularly

/etc/hosts.deny

ALL:ALL

Or:
ALL:ALL spawn (echo –e "\n Tcp Wrappers \: Refused \n \
By\: $(uname –n) \n Process\: %d (pid %p) \n \
Host\: %c \n Date\: $(date) \n \
" | mail –s tcpw@$(uname -n). %u@%h ->%d. admin@sys.com)

The above causes an email to be sent to the system admin whenever a connection is refused.

Hosts.allow Options

- Telnetd: 123.123.123.4 : options
- Options are:
  - RFC931
    - Does an ident lookup to the originator
  - BANNERS path/filename
    - Displays a banner whether service is granted or not
  - SPAWN (commands)
    - Used to execute a command such as safe_finger and then mailing the response to a security person
    - Only used for denied connections

/etc/hosts.allow

Log but don't really protect

ftpd : all
sshd : all
rshd : all
krshd : all
tftpd : all
bootpd : all
rlogind: all
krlogind: all
telnetd : all
dtspcd : all

/etc/hosts.allow

Log and protect
#Allow all on my network or localhost
All : 123.123.123.0/255.255.255.0 localhost

ftpd : .abc.com,123.123.123.4 EXCEPT jaqui.abc.com
sshd : all : BANNERS /etc/motd
telnetd : 123.123.123.0/255.255.255.0
xmservd : .abc.com,123.123.123.4
rexecd : LOCAL,.abc.com,123.123.123.4
dtspcd : .abc.com,123.123.123.4

sshdfwd-x11: ip addrs : options (allow fwd X11)
Sshdfwd-23 : ip addrs : options (allow fwd 23)

#Or can concatenate services:
ftpd telnetd : abc.com

daemon.info log output for a Saint attack 1/2

Jan 18 15:16:41 jlsys2 snmpd[9872]: EXCEPTIONS: ps2pe: End of file: A file or directory in the path name does not exist. (123.123.123.4)
Jan 18 15:16:41 jlsys2 last message repeated 10 times
Jan 18 15:16:41 jlsys2 bootpd[27048]: stat on "bootpd": No such file or directory
Jan 18 15:16:41 jlsys2 bootpd[27048]: received short packet
Jan 18 15:16:41 jlsys2 last message repeated 10 times
Jan 18 15:16:42 jlsys2 snmpd[9872]: NOTICE: SMUX relation started with (123.123.123.4+38310+3)
Jan 18 15:16:42 jlsys2 inetd[17694]: Running server /usr/local/in/tcpd. The error number is A file or directory in the path name does not exist..
Jan 18 15:16:53 jlsys2 snmpd[9872]: EXCEPTIONS: ps2pe: End of file: A file or directory in the path name does not exist. (SMUX 123.123.123.4+38310+3)
Jan 18 15:16:53 jlsys2 rshd[25574]: refused connect from jlsys.abc.com
Jan 18 15:16:53 jlsys2 ftpd[28024]: refused connect from jlsys.abc.com
Jan 18 15:16:53 jlsys2 rexecd[5902]: refused connect from jlsys.abc.com
Jan 18 15:16:53 jlsys2 telnetd[26378]: refused connect from jlsys.abc.com

daemon.info log output for a Saint attack 2/2

Jan 18 15:16:53 jlsys2 rlogind[19812]: refused connect from jlsys.abc.com
Jan 18 15:16:53 jlsys2 krshd[27450]: refused connect from jlsys.abc.com
Jan 18 15:17:01 jlsys2 rshd[27452]: refused connect from jlsys.abc.com
Jan 18 15:17:07 jlsys2 telnetd[19814]: refused connect from jlsys.abc.com
Jan 18 15:17:07 jlsys2 rshd[26380]: refused connect from jlsys.abc.com
Jan 18 15:17:24 jlsys2 telnetd[5906]: refused connect from jlsys.abc.com
Jan 18 15:17:25 jlsys2 snmpd[9872]: EXCEPTIONS: authentication error: invalid community name: private
Jan 18 15:17:35 jlsys2 snmpd[9872]: EXCEPTIONS: authentication error: invalid community name: secret
Jan 18 15:17:36 jlsys2 telnetd[29928]: refused connect from jlsys.abc.com
Jan 18 15:17:42 jlsys2 ftpd[29930]: refused connect from jlsys.abc.com
Jan 18 15:17:45 jlsys2 snmpd[9872]: EXCEPTIONS: authentication error: invalid community name: write
Jan 18 15:17:55 jlsys2 snmpd[9872]: EXCEPTIONS: authentication error: invalid community name: test1

SSH (Secure Shell)

- 2 major versions – SSH1 and SSH2
- SSH1 being phased out
- Handles secure logins and file transfers
- No more clear text passwords
- License is free if you are a university user or are using it for noncommercial use
- Commercial users need a license (not sure about openssh)

Secure Shell

- SSH encrypts logins
- SCP allows secure file copies
- Ensure system logging is up and working
- First install the wrappers – there is a new version that can now handle IPv6
- Then configure ssh with the wrappers – I usually install v1.2.31 and then v2.4.0
- Creates a channel for running a shell with end-to-end encryption for the session

What SSH protects you from

- This is directly from the SSH FAQ at Tigerlair
- IP spoofing, where a remote host sends out packets which pretend to come from another, trusted host. Ssh even protects against a spoofer on the local network, who can pretend he is your router to the outside.
- IP source routing, where a host can pretend that an IP packet comes from another, trusted host.
- DNS spoofing, where an attacker forges name server records
- Interception of cleartext passwords and other data by intermediate hosts
- Manipulation of data by people in control of intermediate hosts
- Attacks based on listening to X authentication data and spoofed connection to the X11 server
- In other words, ssh never trusts the net; somebody hostile who has taken over the network can only force ssh to disconnect, but cannot decrypt or play back the traffic, or hijack the connection.
- Also - Eavesdropping & connection hijacking
- You are not protected if you set encryption to none

SSH does not protect from:

- IP & TCP attacks
  - Syn flood, TCP RST, bogus ICMP, TCP desynchronization
- Denial of service attacks
- Security holes in network services
- Viruses
- Trojan horses
- Bad passwords (password cracking)
- Coffee spills

Installing ssh

- ftp and untar it
- make distclean
- Ensure you copy tcpd.h to /usr/include
- ./configure –with-libwrap=/usr/local/lib/libwrap.a --with-x
- make
- make install
- Install ssh-1 completely, then install ssh-2

Starting sshd

- Set /usr/local/sbin/sshd to start at boot (rc.local or inittab)
- Or start from inetd.conf
  - 1. Add to /etc/services
  - ssh tcp/22
  - 2. Add to /etc/inetd.conf
  - ssh stream tcp nowait root /usr/local/sbin/sshd sshd –i
  - Can add a wrapper if so desired

Configure options

- ./configure –help
- --with-x
- --enable-X11-forwarding
- --with-etcdir=/usr/local/etc
- --with-libwrap=/usr/local/lib or –with-tcp-wrappers
- --enable-tcp-port-forwarding
- --enable-debug

DB2 for AIX users

- Ssh1
  - Prior to running the configure
  - rm /var/adm/lastlog
- Ssh2
  - Prior to running the configure
  - touch /var/adm/lastlog
  - vi /etc/security/lastlog and clear out the junk

Commands

- ssh Replaces rsh
- slogin Replaces rlogin
- scp Replaces rcp
- sftp Replaces FTP
- sshd Daemon on server
- Key Management
  - ssh-agent
  - ssh-add

Commands

- ssh –l jaqui system.com
  - Starts an ssh session to system.com
- scp jaqui@remotesys.com:file file1
  - Copies file to jaqui on remotesys.com as file1
- sftp – only does binary transfers
- ~ctrl Z
  - Puts ssh session in background so you can do something on local machine

Authentication – SSH2

- HostBased
- Public-key
  - DSA, RSA, OpenPGP
- Password
  - Host login password, SecureID, s/key, …
- F-secure client tries:
  - Public-key
  - Password
- All rhosts authentication was dropped in SSH2

Keys

- Can authenticate using password or keys
- Keys require theft of 2 components – identity file on disk and the passphrase in your head
- Password requires only theft of password
- Private key and public key with passphrase
- ssh-keygen
  - Generates your private/public key pair
- Passphrase should be 10-15 characters long
- Copy public key (identity.pub) to all of your remote accounts
  - vi ~/.ssh2/authorization on server
  - Vi ~/.ssh2/identification on client

Using ssh-agent

- Allows you to connect to other systems without passwords during this login session
- Copy your public key files to other systems
- Locally invoke ssh-agent to run in background
- Choose keys you will need
- Load those keys into ssh-agent using ssh-add and typing in the passphrase
- Remains till you logout or kill ssh-agent

/etc/ssh2_config

 # ssh2_config
SSH 2.0 **Client** Configuration File
*:

Port	22
Ciphers	AnyStdCipher
IdentityFile	identification
AuthorizationFile	authorization
RandomSeedFile	random_seed
VerboseMode	no
PasswordPrompt	"%U's password: "
#LocalForward	"110:pop3.ssh.fi:110"
#RemoteForward	"3000:foobar:22"
Ssh1AgentCompatibility	none
#Ssh1AgentCompatibility	traditional or ssh2

/etc/sshd2_config 1/2

 # sshd2_config (SERVER DAEMON)
*:

Port	22	
ListenAddress	0.0.0.0	
Ciphers	AnyStd	
IdentityFile	identification	
AuthorizationFile	authorization	
HostKeyFile	hostkey	
PublicHostKeyFile	hostkey.pub	
RandomSeedFile	random_seed	
MaxConnections	32	
ForwardAgent	yes	
ForwardX11	yes	
IgnoreRhosts	yes	applies to .rhosts and .shosts
IgnoreRootRhosts	yes	

/etc/sshd2_config 2/2

PasswordAuthentication	yes
PasswordGuesses	3
PermitRootLogin	yes
PubkeyAuthentication	yes
ForcePTTYAllocation	no
VerboseMode	no
PrintMotd	yes
UserConfigDirectory	"%D/.ssh2"
SyslogFacility	AUTH
Ssh1Compatibility	yes
KeepAlive	yes
RequireReverseMapping	yes
UserKnownHosts	yes

subsystem definitions

subsystem-sftp	sftp-server

Additional Features

ChRootUsers	jaqui	
ChRootGroups	group1	
PermitRootLogin	yes or no	
DenyUsers	jaqui	
Allowusers	jaqui@system.com	
SilentDeny	no	log denys verbosely
PrintMotd	yes	
CheckMail	yes	
PermitEmptyPasswords	no	
ForcedEmptyPasswdChange	yes	

/etc/sshrc run at login time
If /etc/nologin exists sshd will only allow root to login

Subsystems

subsystem definitions

subsystem-sftp	sftp-server
subsystem-backups	/bin/tar /dev/rmt0 /home
Subsystem-imap	/usr/sbin/imapd

Now …..

ssh –s backups server.com	(uses ssh to backup /home to rmt0 on remote)

Types of Forwarding

- TCP port forwarding
  - Allows forwarding any TCP based service such as telnet or ftp or nntp over secure channels
- X forwarding
  - Comprises additional features for securing the X protocol
- Agent forwarding
  - Allows ssh clients to access public keys on remote systems

Cool SSH Tips

- Backup using tar via an ssh tunnel
- Tunnel ssh through your firewall
- Add ssh to the rdist/rsync configs and tunnel them
- Run ppp over an ssh tunnel
- Socks support
- AFS/Kerberos support
- X11 forwarding
- >=2.0.13 supports PGP keys
- Does NOT work with UDP protocols such as NIS or NFS

Tunneling Telnet and FTP

On ssh server.com
ssh –R 1234:localhost:23 –l jaqui ssh.client.com

This maps port 1234 (note >1024) on your host to ssh.client.com port 23 (telnet) and starts an encrypted session

Now from your host
telnet localhost 1234

You are now connecting via a secure tunnel back to the server

ssh –L 1234:localhost:21 ssh.host.com
Now from localhost
ftp localhost 1234

X11 Forwarding

./configure –with-x –enable-X11-forwarding

sshd_config
X11Forwarding yes

sshd2_config
ForwardX11 yes

hosts.allow
sshdfwd-x11: ip addrs: options

Known Problems

- AIX ML5 to ML7 (fixed at ML8)
  - Bug with /usr/lib/drivers/ptydd
  - Keep the old ml3 version and use it after the upgrade
  - The ssh session will die if you open a second one
- Don't ask how to implement slogin without passwords
- Use Deja news (www.deja.com)
  - comp.security.ssh
  - comp.security.firewalls

SSH References

- www.tigerlair.com/ssh/faq/ssh-faq.html
- www.ssh.org
- www.ssh.com
- www.openssh.com
- www.vandyke.com
- www.f-secure.com

References

- SG24-5971 Additional AIX Security Tools on AIX & SP
- SG24-5521 Exploiting SP security – keeping it alive
- www.porcupine .org (TCP Wrappers)
- http://www.ipsec.com/tech/archive/secsh.html
- SSH The Secure Shell, Barrett & Silverman
  - ISBN 0-596-00011-1