Global Information Assurance Certification Paper

William J. Von Hagel, Jr.
Capitol SANS Track 6 Certification
2/20/01

Red Hat Linux 6.2 Installation Checklist

The following checklist is provided to secure an Intel Pentium II computer with 128MB RAM and 3 Ethernet cards using Red Hat Linux version 6.2. This checklist will secure the abovementioned system from an "out of the box" to an "Internet Ready" state. This machine will be used as a transparent firewall that forwards three services via incoming ports to a "DMZ" zone (SSH, Sendmail and HTTP) and only SSH to a VPN. Also, PortSentry will be used for intrusion detection to enhance the security of this system.

This installation is unique, as multiple users and such will not have access to the computer. However, in case someone does acquire physical access to the computer, user-specific security measures have been added into the checklist.

Step I: Pre-Installation Configuration

1) ___ Disconnect the computer from the network (pull the Ethernet plug)

Bios Security: Reboot the computer and enter the BIOS

1) ___ Change '*Installed OS'* to '***Other***'
2) ___ Change PS/2 Mouse from '*Auto*' to '***Disabled***'
3) ___ '***ESC***' and '***Save Changes***'

Step II: Installation of Red Hat 6.2

Partitioning

1) ___ Insert Red Hat Linux version 6.2 CD-ROM into the computer
2) ___ '***Reboot***' computer and boot from the CD-ROM
3) ___ Hit ***<enter>*** at the LILO prompt
4) ___ Choose Language
5) ___ Choose Keyboard Type
6) ___ For '*Installation Type*' choose '***Install Custom System***'
7) ___ Choose '***Disk Druid***' for partition setup

Since partitioning is a very religious debate, I will only offer my partition table as an example for you to follow if you choose. However, making /var, /home, and / separate partitions is advised. /var is especially important on this system for firewall logging. Also, /var should be a separate partition since /var usually contains files that change when the system is run normally. It is also important especially for this machine, that you double the RAM size for the swap partition.

Mount Point	Device	Requested	Actual	Type

/boot	hda1	14M	14M	Linux native
/var	hda5	4000M	4000M	Linux native
/usr	hda6	2000M	2000M	Linux native
/home	hda7	5861M	5861M	Linux native
/	hda9	1000M	1000M	Linux native
/tmp	hda10	500M	500M	Linux native
	hda11	256M	256M	Linux swap

8) ___ '***Save Changes***' to the partition table
9) ___ Under '*Choose Partitions To Format*', '***select All***' and '***select the check box to check for bad blocks during format***' (Just to be on the safe side)
10) ___ Under '*LILO Configuration*', just hit '***Next***' (There is no need to pass special options to the kernel at boot time)
11) ___ Under '*LILO Configuration (2)*', Install the boot loader to '***/dev/hda Master Boot Record (MBR)***'
12) ___ Do '***NOT***' allow LILO to install another OS (There is none)
13) ___ Assign the hostname to the computer (e.g. Beavis)
14) ___ Unclick '***Use bootp/dhcp***' and enter information about the computer
15) ___ Click '***No Mouse***'
16) ___ Configure clock
17) ___ Pick a '***good***' password. (Definition of good provided by SANS password guidelines)
18) ___ Add a user '***Admin***' for administration purposes (Do not only use the root account)
19) ___ Under '*Passwords*,' select '***Enable Shadow Passwords***' and '***Enable MD5 Passwords***'. Do NOT select '***Enable NIS***'

Package Selection

1) ___ Only select '***Networked Workstation***' and '***Utilities***' from the package list presented by the Red Hat Installer
2) ___ Select '***Individual Packages***'
3) ___ Remove the following packages:

Applications/Internet	finger, ftp, fwhois, rsh, rsync, talk, telnet
Applications/Publishing	ghostscript, ghostscript-fonts, groff-perl, mpage, pnm2ppa, rhs-printfilters
Applications/System	arpwatch, bind-utils, rdate, screen
Documentation	indexhtml
System Environment/Base	chkfontpath, shapecfg, yp-tools
System Environment/Daemons	Xfree86-xfs, lpr, nfs-utils,

	pidentd, portmap, ypbind
System Environment/Libraries	Xfree86-libs, libpng
User Interface/X	urw-fonts

There are many reasons for removing the above-mentioned packages. First of all, a firewall has no need to run a graphical interface. There is also no reason to run applications such as telnet, talk, ftp, finger, etc.

4) ___ Click '***OK***' to begin formatting file system partitions
5) ___ Label a disk '***Red Hat v6.2 Boot Diskette***' and '***Create Boot Diskette***'
6) ___ Remove all media (floppy and CD-ROM) and reboot the system (Installation finished)

Step III: After Installation Configuration:

Although there are numerous ways to perform after installation configuration, making sure the Ethernet cards are installed and functioning properly is crucial. In my firewall, I used 2- Dlink Ethernet cards and 1 RealTek rtl8139 Ethernet card.

1) ___ '***Log on***' as root
2) ___ Edit (vi /etc/conf.modules) and enter the following lines:
alias eth0 rtl8139
alias eth1 ne2k-pci
alias eth2 ne2k-pci
3) ___ '***Power off***' computer
4) ___ '***Insert RealTek Ethernet Card***' into the computer in the first PCI slot.
5) ___ '***Boot the computer***' and allow Red Hat's '*kuzu*' program to auto detect the card.
6) ___ '***Power off***' computer
7) ___ '***Insert D-Link Ethernet Card***' into the computer in the second PCI slot.
8) ___ '***Boot the computer***' and allow Red Hat's '*kuzu*' program to auto detect the card.
9) ___ '***Power off***' computer
10) ___ '***Insert D-Link Ethernet Card***' into the computer in the third PCI slot.
11) ___ '***Boot the computer***' and allow Red Hat's '*kuzu*' program to auto detect the card.

In order to compile and install several programs later, follow the directions below to install several other packages that we will erase later. At a command prompt (as root), enter the following commands:

1) ___ ***mount /dev/cdrom /mnt/cdrom***

2) ___ ***cd /mnt/cdrom/RedHat/RPMS***
3) ___ ***rpm –i autoconf-2.13-5.noarch.rpm m4-1.4-12.i386.rpm automake-1.4-5.noarch.rpm dev86-0.14.9-1.i386.rpm bison-1.28-1.i386.rpm byacc-1.9-11.i386.rpm cdecl-2.5-9.i386.rpm cpp-1.1.2-24.i386.rpm cproto-4.6-2.i386.rpm ctags-3.2-1.i386.rpm egcs-1.1.2-24.i386.rpm ElectricFence-2.1-1.i386.rpm flex-2.5.4a-7.i386.rpm gdb-4.18-4.i386.rpm kernel-headers-2.2.12-20.i386.rpm glibc-devel-2.1.2-11.i386.rpm make-3.77-6.i386.rpm patch-2.5-9.i386.rpm inetd-0.16-4.i386.rpm***

In order to remove some packages/ services, several services need to be stopped first. The checklist below will aid in removing unnecessary software from the computer:

1) ___ ***/etc/rc.d/init.d/apmd stop***
2) ___ ***/etc/rc.d/init.d/sendmail stop***
3) ___ ***/etc/rc.d/init.d/kudzu stop***
4) ___ ***rpm –e –nodeps pump mt-st eject bc mailcap apmd kernel-pcmcia-cs linuxconf getty_ps isapnptools setserial kudzu raidtools gnupg Redhat-logos redhat-release gd pciutils rmt sendmail dump cpio***

The following are configuration settings that will help increase the general security of the box. Some of the following instructions help against the "human factor."

1) ___ Enable colors while searching the directories by editing (vi /etc/profile) and enter the following lines:
Enable Color ls
eval `dircolors /etc/DIR_COLORS –b`
export LS_OPTIONS='-s –F –T 0 --color=yes'
2) ___ Then, enable colors in root's .bashrc file (vi /root/.bashrc) by adding the following line:
alias ls = ' ls --color=auto'

Update Software for Security Reasons

1) ___ Set up a computer to the internet (not the one we are securing)
2) ___ Surf to '***www.redhat.com/apps/support/updates.html***'
3) ___ Click on '***All Red Hat Errata***'
4) ___ Under '*Version 6.2*' click on '***Security Advisories***' and download the following packages: (Do not worry about the sections entitled '*Bug Fixes*' and '*Package Enhancements*' as they do not contain any relevant packages for our system.)
inetd, PAM, bash, ncurses (non-development), iputils (internet testing), traceroute, man, kernel, kernel-headers

5) ___ Move these files onto the linux machine via a CD-ROM or multiple floppy diskettes and upgrade all packages using the directions below:
mount /dev/cdrom /mnt/cdrom
cp /mnt/cdrom/*.rpm /root/fixes/
rpm –Fvh *

Securing the */etc/inetd.conf* file

1) ___ Do not allow anyone except root to access the file:
chmod 600 /etc/inetd.conf
2) ___ Remove all services that we do not need (All of them) by editing (vi /etc/inetd.conf) and commenting out the following services:
ftp, telnet, shell, login, talk, ntalk, finger, auth
3) ___ Send a SIGHUP signal
killall –HUP inetd
4) ___ Set the file immutable (unchangeable)
chattr +i /etc/inetd.conf

Disable ALL Console Access

1) ___ Create the following script (disabling.sh) as root:
(vi /root/disabling.sh)
#!/bin/sh
cd /etc/pam.d
for i in * ; do
sed '/[^#]. *pam_console.so/s/^/#/' < $i > foo && mv foo $i
done
2) ___ Type in the following commands to activate the script: (This script comments out all lines that refer to pam_console.so under /etc/pam.d)
chmod 700 /root/disabling.sh
./disabling.sh
3) ___ Remove the script from your computer:
rm /root/disabling.sh

Deny access to ALL as default

1) ___ Edit (vi /etc/hosts.deny) and add the following line:
ALL: ALL@ALL***, PARANOID***
2) ___ Check that no errors are reported by running:
tcpdchk

Remove Bad Aliases

1) ___ Edit (vi /etc/aliases) and comment out the following aliases:

games, ingress, system, toor, uucp, manager, dumpster, operator, decode

2) ___ Run the following to activate the new aliases file:
/usr/bin/newaliases

Only allow '*admin*' to su to '*root*'

1) ___ Edit (vi /etc/pam.d/su) and add the following lines to the top:
auth sufficient /lib/security/pam_rootok.so debug
auth required /lib/security/pam_wheel.so group =wheel

2) ___ Add '*admin*' to the '***wheel***' group by the following command:
usermod –G10 admin

Set Resource Limits (DoS attacks)

1) ___ Edit (vi /etc/security/limits.conf) and change/ add the following lines:
**** Hard core 0***
**** Hard rss 5000***
**** Hard nproc 20***

2) ___ Edit (vi /etc/pam.d/login) and add the following to the bottom:
session required /lib/security/pam_limits.so

/etc/lilo.conf

1) ___ Edit (vi /etc/lilo.conf) and change/ add the following:
*Timeout=**00***
Restricted *(after default=linux)*
Password=pickyourfavoritepassword

2) ___ Only allow root to see the file (there is a plaintext password in it):
chmod 600 /etc/lilo.conf

3) ___ Update lilo and set the config file immutable
/sbin/lilo –v
chattr +i /etc/lilo.conf

Disable CTL-ALT-DEL

1) ___ Edit (vi /etc/inittab) and comment out the following line with a '#':
ca::ctrlaltdel:/sbin/shutdown –t3 –r now

2) ___ Activate the change:
/sbin/init q

Remove All OS Description Information

1) ___ Edit (vi /etc/rc.d/rc.local) and place a '#' in front of all the following lines:

echo "" > /etc/issue
echo "$R" >> /etc/issue
echo "Kernel…
cp –f /etc/issue /etc/issuenet
echo >> /etc/issue

2) ___ Get rid of the files that contain OS info:
rm –f /etc/issue
rm –f /etc/issue.net

Remove SU-ID bits from files

1) ___ Find all SU-ID files:
***find / -type f \ (-perm –04000 –o –perm –02000 \) \-exec ls –lg {} ***

2) ___ Remove the following SU-ID files from the list:
chmod a-s /usr/bin/chage
chmod a-s /usr/bin/gpasswd
chmod a-s /usr/bin/wall
chmod a-s /usr/bin/chfn
chmod a-s /usr/bin/chsh
chmod a-s /usr/bin/newgrp
chmod a-s /usr/bin/write
chmod a-s /usr/sbin/usernetctl
chmod a-s /usr/sbin/traceroute
chmod a-s /bin/mount
chmod a-s /bin/umount
chmod a-s /bin/ping
chmod a-s /sbin/netreport

Require root password when entering single mode

1) ___ Edit (vi /etc/inittab) and place the following line after the '***si::sysint…***' line:
~~:s:wait:/sbin/suloggin

2) ___ Enter the following to make the change effective:
init q

Configure real-time logging on VTY's 7 and 8 (ALT-F7, ALT-F8)

1) ___ Edit (vi /etc/syslog.conf) and append the following lines to the file:

****.info;mail.none;authpriv.none***	***/dev/tty7***
authpriv.*	***/dev/tty7***
****.warn;*.err***	***/dev/tty7***
kern.*	***/dev/tty7***
mail.*	***/dev/tty8***

2) ___ Create and set permissions to log files and restart logging:

touch /var/log/syslog /var/log/kernel
chmod 700 /var/log/syslog /var/log/kernel
killall –HUP syslogd

3) ___ Configure log rotation for these 2 new log files by editing (vi /etc/logrotate.d/syslog) and adding the following:

```
/var/log/kernel {
    compress
    postrotate
        /usr/bin/killall –9 klogd
        /usr/sbin/klogd &
    endscript
}

/var/log/syslog {
    compress
    postrotate
        /usr/bin/killall –HUP syslogd
    endscript
}
```

4) ___ Edit (vi /etc/logrotate.conf) and modify log rotation behavior. Since a firewall has the need to keep lots of logging information, I recommend increasing the default values to capture more information.

General Security Measures

1) ___ Change the default password length by editing (vi /etc/login.defs) and changing the line '*PASS_MIN_LEN 5*' to '***PASS_MIN_LEN 8***'

2) ___ Set the login timeout for the root account by editing (vi /etc/profile) and placing the following line after the '*HISTSIZE*' line: (2 hours)
TMOUT=7200

3) ___ Stop regular users from using certain services by typing the following commands:
rm –f /etc/security/console.apps/halt
rm –f /etc/security/console.apps/poweroff
rm –f /etc/security/console.apps/reboot
rm –f /etc/security/console.apps/shutdown

4) ___ Do not allow the system to respond to '*pings*' by editing (vi /etc/rc.d/rc.local) and adding the following line to the beginning:
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

5) ___ Use better secure host name resolvation by editing (vi /etc/host.conf) and adding the following:
 # Lookup names via DNS first, then fall back to /etc/hosts
order bind, hosts

 # We have machines with multiple IP addresses
multi on
 # Check for IP spoofing
nospoof on

6) ___ Do not allow users to alter the /etc/services file:
chattr +i /etc/services

7) ___ Do not allow '*root*' to logon directly through a console by editing (vi /etc/securetty) and commenting out all the lines with a '#'

8) ___ Disable ALL default vender accounts by the following commands:
userdel adm
userdel lp
userdel sync
userdel shutdown
userdel halt
userdel news
userdel uucp
userdel operator
userdel games
userdel gopher
userdel ftp

9) ___ Disable ALL default group accounts with the following commands:
groupdel adm
groupdel lp
groupdel news
groupdel uucp
groupdel games
group dip
groupdel pppusers
groupdel popusers
groupdel slipusers

10) ___ Do not allow password files to be changed by the following commands:
chattr +i /etc/passwd
chattr +i /etc/shadow
chattr +i /etc/group
chattr +i /etc/gshadow

11) ___ Limit shell logging by editing (vi /etc/profile) and changing the '*HISTSIZE*' line to the following:
HISTSIZE=20

12) ___ Set the correct permissions for script files:
chmod –R 700 /etc/rc.d/init/d/

13) ___ Only allow '*root*' and '*admin*' to logon to the console (explicitly) by editing (vi /etc/security/access.conf) and adding the following line:
-:ALL EXCEPT root admin :console

14) ___ Change the default '*syslog*' settings by editing (vi /etc/syslog.conf) file and add/ change the following lines: (note: separators must be tabs.)
.warn;*.err /var/log/syslog
kern.* /var/log/kernel

General System Optimizations

The following part of this checklist corresponds to making Linux run better/ faster on the specific architecture of an Intel Pentium II computer.

1) ___ Edit (vi /etc/profile) and add/change the following: (logout/ login to incorporate changes)
CFLAGS='-O9 –funroll-loops –ffast-math –malign-double –mcpu=pentiumpro –march=pentiumpro –fomit-frame-pointer –fno-exceptions'

1a) ___ Add the following to the end of the '***EXPORT***' line:
CFLAGS LANG LESSCHARSET

2) ___ Edit (vi /etc/rc.d/rc.local) and add the following line: (This allows Linux to get better performance with virtual memory)
Echo "100 1200 128 512 15 5000 500 1884 2" > /proc/sys/vm/bdflush

3) ___ Edit (vi /etc/rc.d/rc.local) and add the following line to change '*buffermem*' parameters (VM performance increase)
echo "80 10 60" > /proc/sys/vm/buffermem

4) ___ Edit (vi /etc/nsswitch.conf) and remove all NIS references from the file. (***Delete them***)

5) ___ Do not log the 'last access time' for files that the system accesses a LOT. ('atime' attribute) by typing in the following command:
chattr –R +a /var/spool/

Give 'root' unlimited number of processes

1) ___ Edit (vi /root/.bashrc) and add the line:
ulimit –u unlimited

2) ___ Verify that the change has taken effect:
ulimit –a

Give 'root' permission to open more files then the default

1) ___ Edit (vi /root/.bashrc) and add the line:
ulimit –n 90000

2) ___ Verify that the change has taken effect:
ulimit -a

Firewall Script

The firewall script below uses a config file (chains.config) where all variables are declared. The only thing a user needs to do is edit the config file, and the "chains" script will work to provide the following: (specific example)

- Allow only ssh/ http/ sendmail to come into the server.
- Redirect this traffic (ssh/ http/ sendmail) to eth1 interface (DMZ Zone) and allow only ssh to run to eth2.
- Eth1 is the interface to a single computer that runs ONLY a mail server and an web server
- Eth2 is the interface to a secure private network consisting of multiple computers.

chains.conf

```
MASQURADE="yes"
INET_INTERFACE="eth0"
LOCAL_INTERFACE="eth1"
LOCAL_NETWORK="10.10.10.0/24"
LOCAL_SERVICES=""
LOCAL_BLOCKED=""
DMZ_INTERFACE="eth2"
DMZ_NETWORK="10.10.11.0/24"
DMZ_SERVICES="22 80"
DMZ_BLOCKED=""
BLOCK_PING="yes"
```

chains file

```
#!/bin/bash
#
# $Header: /home/pyg/bin/RCS/chains,v 1.11 1999/12/29 03:58:13 pyg Exp
$
# Chad Knepp <pyg@cs.svsu.edu>
# 7/28/99
#
# This script hopefully generates reasonably secure ipchains rules for
# Linux boxes running 2.2.x kernels that have the
# following options:
#    CONFIG_FIREWALL=y
#    CONFIG_IP_FIREWALL=y
# and might have the following:
#    CONFIG_IP_MASQUERADE=y
# while you are at it, also do (for the future):
#    CONFIG_NETLINK=y
#    CONFIG_IP_FIREWALL_NETLINK=y
#
# If you have a dialup connection to the internet, this should be run
# immediately after making a connection to the Internet.  If the
# connection is interrupted this script must be re-run or ppp0 will
# not work at all until re-run!  This script should be a part of any
# keep-alive.
```

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
export PATH

CONFIG="./chains.config"
if [ $1 ]
then
    . $1
else
    if [ ! -e "$CONFIG" ]
    then
      echo "FATAL:  Can't find a config file."
      exit 1
    fi
    . $CONFIG
fi

# Valid [necessary] config fields (see $CONFIG)
#
# INET_INTERFACE="ppp0"
# MASQURADE="yes"
# LOCAL_NETWORK="10.0.0.0/24"
# LOCAL_INTERFACE="eth0"
# LOCAL_SERVICES="22 113"
# TRUSTED_LOCAL_SERVICES=""
# BLOCKED="1024 2049 6000 6010:6013 7100"
# REMOTE_BLOCKED=""
# BLOCK_PING="yes"
# DEBUG="yes"


if [ "$UID" != 0 ]
then
    echo "FATAL:  You must be root."
    exit 1
fi


MYIP=`ifconfig $INET_INTERFACE|/bin/grep inet|/bin/sed s/\:/\
/|/usr/bin/awk '{print $3}'`
#GATEWAY=`ifconfig $INET_INTERFACE|/bin/grep inet|/bin/sed s/\:/\
/g|/usr/bin/awk '{print $5}'`


if [ ! $MYIP ]
then
    echo "FATAL: Not connected, no such interface as $INET_INTERFACE."
    exit 1
fi

echo -n "Configuring ipchains."


# Flush the old rules
#
ipchains -F input
```

```
ipchains -F output
ipchains -F forward


# Set up IP spoofing protection
#
if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]
then
    for FILTER in /proc/sys/net/ipv4/conf/*/rp_filter
    do
      echo 1 > $FILTER
    done
    echo -n "."
else
    echo
    echo "WARNING: Could not set up IP spoofing protection."
fi
ipchains -A input -l -j DENY -s 127.0.0.0/8 -i ! lo


# Setup IP masqurading
#
if [ "$MASQURADE" = "yes" ]
then
    if [ -e /proc/sys/net/ipv4/ip_forward ]
    then
      ipchains -P forward REJECT
      ipchains -A forward -i $INET_INTERFACE -j MASQ
      /bin/echo 1 > /proc/sys/net/ipv4/ip_forward
      echo -n "."
    else
      echo
      echo "WARNING: Could not set up IP masqurading."
    fi

    # Accept anything on the local network (eth0 and localnet)
    if [ $LOCAL_NETWORK ]
      then

      # Paranoia at it's prime (this is probably unnecessary)
      #
      ipchains -A input -l -j DENY -i $LOCAL_INTERFACE -s !
$LOCAL_NETWORK 2>/dev/null
    fi

    # Accept anything on the local network (eth0 and localnet)
    if [ $DMZ_NETWORK ]
      then

      # Paranoia at it's prime (this is probably unnecessary)
      #
      ipchains -A input -l -j DENY -i $DMZ_INTERFACE -s ! $DMZ_NETWORK
2>/dev/null
    fi
else
    /bin/echo 0 > /proc/sys/net/ipv4/ip_forward 2>/dev/null
fi
```

```
# Start by denying every IP that is not $MYIP yet appears to be me
# input and output via interface.  This means that nothing will work on
# interface if the IP changes due to a reconnection which will force
the
# script to be re-run.  This is known as Playing It Safe(tm)
#
ipchains -A input -j DENY -s 0/0 -d ! $MYIP -i $INET_INTERFACE
ipchains -A output -j DENY -s ! $MYIP -d 0/0 -i $INET_INTERFACE


# Lay down the law
#
if [ "$LOCAL_SERVICES" ]
then
    for PORT in $LOCAL_SERVICES
    do
      ipchains -A input -j ACCEPT -p tcp -s 0/0 -d $MYIP $PORT -i
$INET_INTERFACE
      ipchains -A output -j ACCEPT -p tcp -s $MYIP $PORT -d 0/0 -i
$INET_INTERFACE
      ipchains -A input -j ACCEPT -p udp -s 0/0 -d $MYIP $PORT -i
$INET_INTERFACE
      ipchains -A output -j ACCEPT -p udp -s $MYIP $PORT -d 0/0 -i
$INET_INTERFACE
    done
fi
if [ "$DMZ_SERVICES" ]
then
    for PORT in $DMZ_SERVICES
    do
      ipchains -A input -j ACCEPT -p tcp -s 0/0 -d $MYIP $PORT -i
$INET_INTERFACE
      ipchains -A output -j ACCEPT -p tcp -s $MYIP $PORT -d 0/0 -i
$INET_INTERFACE
      ipchains -A input -j ACCEPT -p udp -s 0/0 -d $MYIP $PORT -i
$INET_INTERFACE
      ipchains -A output -j ACCEPT -p udp -s $MYIP $PORT -d 0/0 -i
$INET_INTERFACE
    done
fi
echo -n "."

if [ "$TRUSTED_LOCAL_SERVICES" ]
then
    for PORT in $TRUSTED_LOCAL_SERVICES
    do
      HOST=`/bin/echo $PORT|/bin/sed s/\+/\ /|/usr/bin/awk '{print
$1}'`
      SERVICE=`/bin/echo $PORT|/bin/sed s/\+/\ /|/usr/bin/awk '{print
$2}'`
      ipchains -A input -j ACCEPT -p tcp -s $HOST -d $MYIP $SERVICE -i
$INET_INTERFACE
      ipchains -A output -j ACCEPT -p tcp -s $MYIP $SERVICE -d $HOST -i
$INET_INTERFACE
```

```
      ipchains -A input -j ACCEPT -p udp -s $HOST -d $MYIP $SERVICE -i
$INET_INTERFACE
      ipchains -A output -j ACCEPT -p udp -s $MYIP $SERVICE -d $HOST -i
$INET_INTERFACE
    done
    echo -n "."
fi

if [ "$REMOTE_BLOCKED" ]
then
    for PORT in $REMOTE_BLOCKED
    do
      ipchains -A input -j REJECT -s 0/0 $PORT -d $MYIP -i
$INET_INTERFACE
      ipchains -A output -j REJECT -s $MYIP -d 0/0 $PORT -i
$INET_INTERFACE
    done
    echo -n "."
fi

if [ "$LOCAL_BLOCKED" ]
then
    for PORT in $LOCAL_BLOCKED
    do
      if [ "$DEBUG" = "yes" ]
      then
          ipchains -A input -l -j REJECT -p tcp -s 0/0 -d $MYIP $PORT -
i $INET_INTERFACE
          ipchains -A output -l -j REJECT -p tcp -s $MYIP $PORT -d 0/0
-i $INET_INTERFACE
          ipchains -A input -l -j REJECT -p udp -s 0/0 -d $MYIP $PORT -
i $INET_INTERFACE
          ipchains -A output -l -j REJECT -p udp -s $MYIP $PORT -d 0/0
-i $INET_INTERFACE
      else
          ipchains -A input -j REJECT -p tcp -s 0/0 -d $MYIP $PORT -i
$INET_INTERFACE
          ipchains -A output -j REJECT -p tcp -s $MYIP $PORT -d 0/0 -i
$INET_INTERFACE
          ipchains -A input -j REJECT -p udp -s 0/0 -d $MYIP $PORT -i
$INET_INTERFACE
          ipchains -A output -j REJECT -p udp -s $MYIP $PORT -d 0/0 -i
$INET_INTERFACE
      fi
    done
    echo -n "."
fi


# Deny certain parts of ICMP
#
if [ "$BLOCK_PING" = "yes" ]
then
    echo
    echo -n "blocking ping"
    if [ "$DEBUG" = "yes" ]
    then
```

```
      ipchains -A input -l -j REJECT -p icmp -s 0/0 8 -i
$INET_INTERFACE
      ipchains -A output -l -j REJECT -p icmp -s 0/0 0 -i
$INET_INTERFACE
      ipchains -A input -l -j REJECT -p icmp -s 0/0 5 -i
$INET_INTERFACE
      ipchains -A output -l -j REJECT -p icmp -s 0/0 5 -i
$INET_INTERFACE
      ipchains -A output -l -j REJECT -p icmp -s 0/0 11 -i
$INET_INTERFACE
    else
      ipchains -A input -j REJECT -p icmp -s 0/0 8 -i $INET_INTERFACE
      ipchains -A output -j REJECT -p icmp -s 0/0 0 -i $INET_INTERFACE
      ipchains -A input -j REJECT -p icmp -s 0/0 5 -i $INET_INTERFACE
      ipchains -A output -j REJECT -p icmp -s 0/0 5 -i $INET_INTERFACE
      ipchains -A output -j REJECT -p icmp -s 0/0 11 -i $INET_INTERFACE
    fi
    echo -n "."
fi


# Configure type of service (does this do any good at all?)
#
#ipchains -A output -p tcp -d 0.0.0.0/0 ssh -t 0x01 0x10
#ipchains -A output -p tcp -d 0.0.0.0/0 telnet -t 0x01 0x10
#ipchains -A output -p tcp -s 0.0.0.0/0 ftp-data -t 0x01 0x02


# Deny evrything less than 1024
#
if [ "$DEBUG" = "yes" ]
then
    ipchains -A input -l -j REJECT -p tcp -s 0/0 -d $MYIP 0:1023 -i
$INET_INTERFACE
    ipchains -A output -l -j REJECT -p tcp -s $MYIP 0:1023 -d 0/0 -i
$INET_INTERFACE
    ipchains -A input -l -j REJECT -p udp -s 0/0 -d $MYIP 0:1023 -i
$INET_INTERFACE
    ipchains -A output -l -j REJECT -p udp -s $MYIP 0:1023 -d 0/0 -i
$INET_INTERFACE
else
    ipchains -A input -j REJECT -p tcp -s 0/0 -d $MYIP 0:1023 -i
$INET_INTERFACE
    ipchains -A output -j REJECT -p tcp -s $MYIP 0:1023 -d 0/0 -i
$INET_INTERFACE
    ipchains -A input -j REJECT -p udp -s 0/0 -d $MYIP 0:1023 -i
$INET_INTERFACE
    ipchains -A output -j REJECT -p udp -s $MYIP 0:1023 -d 0/0 -i
$INET_INTERFACE
fi
echo " done"

exit 0
```

Setup NTP

1) ___ Obtain the source code '***ntp-4_0_99k.tar.gz***' from NTP's web site

2) ___ In the root directory, unzip the file and install the program:

```
gunzip ntp-4_0_99k.tar.gz
tar –xvvf ntp-4_0_99k.tar
cd ntp-4.0.99k
./configure
make check
make install
make clean
make distclean
```

3) ___ Create a configuration file (vi /etc/ntp.conf) and add the following lines:

```
driftfile /etc/ntp.drift
server 127.127.1.1
fudge 127.127.1.1 stratum5
server 128.115.14.97        # clock.llnl.gov
server 128.4.1.20           # pogo.udel.edu
server 192.43.244.9         # ncar.ucar.edu
```

4) ___ Create nptd as a service:

```
touch /etc/rc.d/init.d/ntpd
chmod 700 /etc/rc.d/init.d/ntpd
```

5) ___ Edit (vi /etc/rc.d/init.d/ntpd) and add the following lines:

```
if [ -f $CONFFILE ]; then
        if [ -x /usr/local/bin/ntpdate ]; then
                SERVERS=`awk '/^server|peer/ {print $2}' \
                        $CONFFILE | grep –v ^127`
                /usr/local/bin/ntpdate $SERVERS
        fi
        if [ -x /usr/local/bin/xntpd ]; then
                echo "Starting NTP."
                /usr/local/bin/xntpd –c $CONFFILE
        fi
fi
```

6) ___ Create the relevant symlink (from the /etc/rc.d/rc3.d/ directory)

```
ln –s ../init.d/ntpd S99ntpd
```

Step IV: Install Security Related Programs:

Psionic Logcheck

1) ___ Download the source from: http://www.psionic.com/abacus/logcheck

2) ___ Edit (vi /etc/syslog.conf) and edit the following line so that the system will log ALL information:

```
*.info                          /var/log/messages
```

3) ___ Ensure that all files except '***wtmp***' in '***/var/log***' have permissions

set to '***600***'

4) ___ Install the program:
/root/logcheck-1.1.1/make linux

5) ___ Edit (vi /usr/local/etc/logcheck.sh) alter the last 2 lines to read the following: (since sendmail is not installed)
cp $TMPDIR/checkreport.$$ /root/checkreport.$$

TARA (Looking from the Inside Out)

1) ___ Download the source from TARA's website and move the file into '***/root***' directory

2) ___ Unzip and install the program using the following commands:
gunzip tara-2.09.tar.gz
tar –xvvf tara-2.09.tar
cd tara-2.0.9
make install

3) ___ Need to create TARA/ Tiger's working directories using the following commands:
mkdir –m 700 –p /usr/spool/tiger/bin
mkdir –m 700 –p /usr/spool/tiger/logs
mkdir –m 700 –p /usr/spool/tiger/work

4) ___ Run '***tiger***' with the following command: (report is stored in '***/var/spool/tiger/logs/security.report.host.domain.date-time***')
./tiger

5) ___ Fix all FAIL messages generated by the Tiger report. If you have followed ALL the above directions, the only fail messages will pertain to port numbers in the '*/etc/services*' file. To correct this, edit (vi /etc/services) and change all port numbers according to the Tiger report.

6) ___ Run tiger again to ensure that there are NO fail messages.

Looking from the Outside In

From another computer, it would be very helpful to run SATAN, SARA, SAINT and nmap port scanner to help pinpoint weaknesses in the system that has just been created. However, the installation and configuration of these programs is out of scope for this document. This is left as an exercise to the reader.

Psionic Portsentry

1) ___ Obtain the source code from http://www.psionic.com/ and place it in the '***/root/***' directory.

2) ___ Use the following commands to unzip portsentry:
gunzip portsentry-1_0_tar.gz
tar –xvvf portsentry-1_0_tar
cd portsentry-1.0/

3) ___ Edit (vi /root/portsentry-1.0/portsentry.conf) and make the

following changes:
ADVANCED_EXCLUDE_TCP="22,25,80"
ADVANCED_EXCLUDE_UDP="22,25,80"
Uncomment the following line for LINUX!:
#Kill_ROUTE…

3) ___ Install portsentry:
make linux
make install

4) ___ Activate PortSentry:
/usr/local/psionic/portsentry/portsentry –atcp
/usr/local/psionic/portsentry/portsentry –audp

5) ___ Make sure that PortSentry started with no errors: (Should say that Psionic PortSentry is active and listening to ports)
cat /var/log/messages

6) ___ Start PortSentry upon reboot by editing (vi /etc/rc.d/rc.local) and adding the following 2 lines to the bottom:
/usr/local/psionic/portsentry/portsentry –atcp
/usr/local/psionic/portsentry/portsentry –audp

Tripwire

1) ___ Obtain the source code from http://www.tripwire.org/ and place it in the '***/root/***' directory.

2) ___ Use the following commands to unzip tripwire
gunzip tripwire-2_3-47_i386_tar.gz
tar –xvvf tripwire-2_3-47_i386_tar
rpm –i tripwire-2_3-47_i386.rpm

3) ___ Run the install script: (don't forget to enter '*good*' passwords)
/etc/tripwire/twinstall.sh

4) ___ Edit (vi /etc/tripwire/twpol.txt) and comment out lines with a '#' that contain the following programs:
/sbin/accton /sbin/dosfsck /sbin/dump.static /sbin/fsck.msdos /sbin/ftl_check /sbin/ftl_format /sbin/mkdosfs /sbin/mkfs.msdos /sbin/mkpv /sbin/mkraid /sbin/mtx /sbin/parted /sbin/pcinitrd /sbin/raidstart /sbin/resize2fs /sbin/restore.static /sbin/scsi_info /sbin/tapeinfo /sbin/adjtimex /sbin/dhcpcd /sbin/getty /sbin/ifcfg /sbin/ifport /sbin/ifuser /sbin/ip /sbin/iptables /sbin/ipx_configure /sbin/ipx_interface /sbin/ipx_internal_net /sbin/iwconfig /sbin/iwpriv /sbin/iwspy /sbin/portmap /sbin/uugetty /sbin/vgetty /sbin/ypbind /bin/ping /sbin/linuxconf /sbin/linuxconf-auth /sbin/remadmin /sbin/rescuept /sbin/rmt /sbin/rpc.lockd /sbin/rpc.statd /sbin/rpcdebug /sbin/cardctl /sbin/cardmgr /sbin/isapnp /sbin/lspci /sbin/pnpdump /sbin/probe /sbin/pump /sbin/setpci /sbin/shapecfg /sbin/rtmon /sbin/getkey /bin/aumix-minimal /bin/mt /bin/rpm /bin/setserial /bin/sfxload /bin/zsh /bin/zsh-3.0.8 /sbin/askrunlevel /sbin/dnsconf

/sbin/fixperm /sbin/fsconf /sbin/kallsyms /sbin/mailconf /sbin/managerpm /sbin/modemconf /sbin/mount.ncp /sbin/mount.ncpfs /sbin/mount.smb /sbin/mount.smbfs /sbin/netconf /sbin/raid0run /sbin/raidhotadd /sbin/raidhotremove /sbin/raidstop /sbin/rdump.static /sbin/rrestore.static /sbin/userconf /sbin/uucpconf /bin/xnmap /bin/ksh /bin/psh /usr/kerberos/bin/rsh /bin/Rsh /bin/shell /bin/tsh /bin/tcsh /bin/bash2 /etc/group /var/spool/cron/crontabs /etc/csh.cshrc /etc/csh.login /etc/tsh_profile /var/lock/subsys/sendmail /var/lock/subsys/gpm /var/lock/subsys/httpd /var/lock/subsys/sound /var/lock/subsys/smb /var/lock/subsys/anacron /var/lock/subsys/autofs /var/lock/subsys/canna /var/lock/subsys/firewall /var/lock/subsys/identd /var/lock/subsys/jserver /var/lock/subsys/kudzu /var/lock/subsys/reconfig /var/lock/subsys/xfs /var/lock/subsys/xinetd /var/lock/subsys/ypbind /var/run /var/spool/lpd/lpd.lock /etc/issue.net /etc/issue /lib/modules/preferred /root/mail /root/Mail /root/.xsession-errors /root/.xauth /root/.tcshrc /root/.sawfish /root/.pinerc /root/.mc /root/.gnome_private /root/.gnome-desktop /root/.gnome /root/.esd_auth /root/.elm /root/.cshrc /root/.amandahosts /root/.addressbook.lu /root/.addressbook /root/.Xresources /root/.Xauthority /root/.ICEauthority /etc/httpd/conf /var/lib/nfs/rmtab /usr/sbin/fixrmtab /etc/smb.conf /etc/gettydefs /etc/yp.conf

5) ___ Create baseline database: (Database created is in /var/lib/tripwire/name.twd
tripwire –m i

6) ___ Perform an integrity check:
tripwire –m c

7) ___ Backup policy files and database onto a floppy disk:
mount /dev/fd0 /mnt/floppy
cp /etc/tripwire/* /mnt/floppy
cp /var/lib/tripwire/* /mnt/floppy

Step V: Final Touches (We're Done!!!!)

The only things that are left to do is remove all compilers and such from the computer. If you have followed the above directions, then only a few programs should be there

1) __ Remove all compilers and "sharp" programs
rpm –e autoconf m4 automake dev86 bison byacc cdecl cpp cproto ctags egcs ElectricFence flex gdb kernel-headers glibc-devel make

2) ___ Move the 'rpm' binary to a safe place for later use (if necessary)
mount /dev/fd0 /mnt/floppy
mv /bin/rpm /mnt/floppy
umount /mnt/floppy

Step VI: A DMZ Computer

The second part to the above checklist would be to secure a second PC that resides in the DMZ zone and provides SSH, Apache, Bind and Sendmail. Since ***most*** of the security configuration for the firewall box described above is applicable to this second PC and the security checklist for a second PC is outside the scope of the practical, I did not repeat it below. I have included the security checklist for the individual services (SSH, Apache, Bind and Sendmail) of interest.

Apache Web Server

This configuration sets up a very basic web server for static pages only. If you plan to run CGI scripts then another set of configurations must be added. Also, if you need SSL support you need to alter the following configuration. All of this information can be found at Apache's web site (www.apache.org).

1) ___ Obtain source code from (www.apache.org) and place the source code in '*/root/*' directory.
2) ___ Unzip and install the program:
gunzip apache_1_3_14_tar.gz
tar –xvvf apache_1_3_14_tar
cd apache_1.3.14
./configure –-prefix=/web
make
make install
/web/bin/apachectl start
3) ___ Edit (vi /web/conf/httpd.conf) and add/ change the following
<Directory />
Options None
AllowOverride None
order allow,deny
deny from all
</Directory>

<Directory /web/htdocs>
Options SymLinksIfOwnerMatch
AllowOverride AuthConfig
order allow,deny
allow from all
</Directory>
4) ___ Activate changes

/web/bin/apachectl stop
/web/bin/apachectl start

Bind

1) ___ Obtain source code and place it in '*/root*' directory (Version 8.2.3)
2) ___ Unzip and install the program (This can take a LONG time)
gunzip bind-src.tar.gz
tar –xvvf bind-src.tar
cd src
make depend
make all
rm .settings
make install
3) ___ Setup configuration
mkdir /etc/namedb
4) ___ Edit (vi /etc/named.conf) and add the following lines:
options {
directory "/etc/namedb";
version "Go Away.";
};
5) ___ Make '*named*' a boot script by editing (vi /etc/rc.d/init.d/named) and adding the following lines:
if [-f /usr/sbin/named –a –f /etc/named.conf]; then
/usr/sbin/named;
fi
6) ___ Set permissions on the file and add a symlink:
chmod 700 /etc/rc.d/init.d/named
cd /etc/rc.d/rc3.d
ln –s ../init.d/named S99named

Sendmail

1) ___ Obtain source code from www.sendmail.org (Version 8.9.3) and place it in the '*/root*' directory
2) ___ Unzip and install the program
gunzip sendmail_8_9_3_tar.gz
tar –xvvf sendmail_8_9_3_tar
cd sendmail-8.9.3/src
sh Build
3) ___ Edit (vi /etc/sendmail.cf) and add/change the following line to prevent the sendmail program from unauthorized access:
O PrivacyOptions=authwarnings,noexpn,novrfy,restrictmailq
4) ___ Do not allow sendmail's version to be queried by users. Edit (vi /etc/sendmail.cf) and change the following line to read:
O SmtpGreetingMessage=Go Away.

5) ___ Change permissions of queue directory
chmod 0700 /var/spool/mqueue

SSH

1) ___ Stuff
2) ___ Unzip and run the program:
gunzip ssh-2_4_0_tar.gz
tar –xvvf ssh-2_4_0_tar
cd ssh-2.4.0
./configure
make
make –n install
make install
make distclean
3) ___ Edit (vi /etc/rc.d/rc.local) and add the following line to allow sshd to be run upon system boot:
 # Boot sshd
/usr/local/sbin/sshd
4) ___ Edit (vi /etc/ssh2/ssh2_config) and change the defaults if necessary.

Last Updated: July 24th, 2017

Upcoming Training

SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**