Global Information Assurance Certification Paper

Securing a Redhat Linux Workstation in an Open Network

GIAC Certified Unix Systems Administrator (GCUX)
Practical Examination Version 1.9
Jason Ormes

Introduction

The primary focus of this document is to show how a Linux based desktop workstation can be secured in an open network environment. As this is the least secure form of network, these steps would also be very appropriate for a more secure environment. To the security conscious individual this seems like a extremely bad situation, but it is quite often the norm for educational and research facilities where the use of a "firewall" or other such filters would hamper the collaborative nature of these intuitions.

Intended Audience

The document is intended for anyone who will be configuring or managing a Linux workstation. The nature of a desktop workstation generally being a user-managed system necessitates that some of the information in this document will be very basic so that it is useful to the user/administrator with little or no Unix experience.

Conventions Used

Some of the conventions that will be used in this document are as follows:

1. Terms will be defined and link at the end of the document in the Glossary.
2. Source code and Command examples will be enclosed in a blue text box like this:

```
[root@workstation /root]# cat /etc/hosts.deny
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
ALL: ALL: banners /etc/banners
[root@workstation /root]#
```

3. The systems' host name will be referred to as "workstation" and all IP addresses will be sanitized. You will need to use your correct host name and addresses when using this document to setup your own system.
4. All passwords used will be replaced with *your_password_here*. Please don't use this as your password.

System Description

This System will be a workstation for the general users in a research/educational type facility. As such, it will be used to check and send email, browse the internet, develop applications, process documents with various document packages, some application development, and do some low load level data analysis that would be considered too minor for use of a server or farm system.

Software Used

Software that will be installed on the system

RedHat 7.3	ftp://ftp.redhat.com/pub/redhat/linux/7.3/en/iso/i386/
IPTables	Comes with the Distribution
LogSentry	http://www.psionic.com/products/logsentry.html
PortSentry	http://www.psionic.com/products/portsentry.html
Sudo	Comes with the distribution

Software that will be used to verify the security of the workstation

Nmap	http://www.insecure.org/nmap/nmap_download.html

Hardware Used

The example system used in this document has the following specifications:

Processor	AMD K6-233mhz
RAM	80MegaBytes
Hard Disk	8.5 GigaBytes
Video Card	TNT2 16Megabyte card
Ethernet Card	3COM 3c59x
Mouse	Microsoft PS/2 Intellimouse
Monitor	Viewsonic E771
Keyboard	Standard 104 Key Keyboard
External Tape Drive	8mm tape drive system

This is a fairly old and generic system that I'm not too sure could handle much in the way of data analysis, but any system with hardware that is listed on the Redhat hardware compatibility list http://hardware.redhat.com/hcl/ should work.

Security Risk Analysis

Preventing unauthorized access and abuse of this system is the primary security goal for this system. Because this is on an open network, the system has a real Internet accessible IP address. This opens it up to attacks from all over the world and throws most of the security onto the local system instead of external systems such as network isolation and firewalls. Logging activity such as failed access attempts will be useful in preventing unauthorized access to the system. Additionally it will be very important to keep the systems software updated frequently to avoid exploits. Since this isn't a "critical system" security will be important, but a critical action plan will not be necessary.

Installation Procedure

1. Obtaining the RedHat 7.3 Distribution Disks. The 2 primary ways to get disks for this distribution are to either purchase them from a local computer store or download them from ftp.redhat.com. For this document we will be downloading them, so using which ever ftp program you like connect with ftp.redhat.com and go to /pub/redhat/linux/7.3/en/iso/i386/. Download the files named:

   valhalla-i386-disc1.iso
   valhalla-i386-disc2.iso
   valhalla-i386-disc3.iso

   Burn these to CDROM using whichever CD burning software you prefer.

   Note: if your system is too old to boot from CDROM as mine is then you will need to download and create a boot floppy. The boot image is located at this location /pub/redhat/linux/7.3/en/os/i386/images, the image name you will most likely need is boot.img. Write this image to disk, you can do it in Linux by issuing the following command:

```
[root@workstation /root]# dd if=boot.img of=/dev/fd0 bs=1440
```

   This will burn the image to the disk. If you are using windows you need to download the rawwritewin.exe from /pub/redhat/linux/7.3/en/os/i386/dosutils and use it to burn the image to the disk.

2. Place CD 1 in the disk drive and either boot from it or the floppy as needed. If your system doesn't boot from the media as planned, you may need to correct this by entering the computers bios and change the boot order.
3. If you've booted correctly, A Redhat Welcome Screen will appear giving you several options of how to boot. The most used options are the default

one of “graphical mode”, “text mode”, and “expert mode”. The graphical mode attempts to launch a small version of Xwindows and uses the mouse to prompt you for installation questions. Text mode is less system intensive and seems to be a bit more stable than the graphical mode. If you have trouble with the graphical mode crashing or not detecting your video card then this is the mode that I recommend and it will be the one that I use for this document. The last option expert mode is used if you need even more control over the installation. If you need special drivers for the disk drive or network adapter this is a useful mode. When you are ready Enter “text” after the “boot:” prompt and press enter.

4. You should be presented with a Welcome to Redhat Linux screen. Feel free to read through the presented information and press enter to continue. As a note, to navigate in this interface you press the tab key to move between objects, the space bar selects items and the return key triggers buttons.
5. Language Selections: Pick the language that you desire. English will be used for this document. Select the “ok” button when you are ready to move on. The arrow keys will help you navigate through the list of languages.
6. Keyboard Model: For most applications the “us” selection will do. If you are using a unique language keyboard feel free to pick the one that you desire. Again like the language selection, the arrow keys will move you through the list. Select the “ok” button when you are finished.
7. Mouse Selection: Choose the type of mouse that you have. If your mouse is not in the list pick one of the generic ones that best suits your needs. If you have a two-button mouse make sure that you select the Emulate 3 Buttons so that you can take advantage of the entire mouse feature. Select the “ok” button when finished.
8. Installation type: Redhat gives several options of pre-selected packages to be installed on your system. Using one of the pre-selected groups doesn’t give you any information as to which package groups are being installed. The custom selection will give you a list of all the packages to be installed and will allow you to add and remove packages as necessary. We will be choosing the custom installation for this document. Move your selection down to “Custom” and select the “ok” button.
9. Disk Partition Setup: You will be asked to select a method of partitioning your disks. With Autopartion, Redhat gives you a suggested partitioning scheme that you can modify. Disk Druid gives you a graphical interface to lay out the partitions. Fdisk is a text-based partitioner that can be used if you are having problems with the other two. We have no control with the Autopartion and would like to be able to size them, as we want so we will select “Disk Druid” button.
10. Using Disk Druid can be an interesting adventure. Use the “tab” key to navigate between fields and buttons. Use the “spacebar” to turn check boxes on and off. Use the “enter” key to execute the buttons. Partitioning for a workstation is pretty straightforward. Setup your swap space to be

twice the size of your RAM unless you have a specific application that needs additional swap space to execute. "/boot" is used to install the kernel files by Redhat, by putting this in a separate partition we can avoid a problem with the 1024 Block limit size of some older BIOS's. "/var" is used for logging, we set this in a separate partition from "/" to prevent the system from crashing if the logs over flow due to natural causes or denial of service attacks. "/" Needs to be big enough for all of the applications that you will be using. "/home" is for user home areas, this prevents a user from overflowing the system. Our completed partition table will look something like this:

Mount Point	Size	Type
Swap	128M	swap
/boot	50M	ext3
/	3000M	ext3
/var	1000M	ext3
/home	4000M	ext3

Select the "ok" button when you are happy with your configuration.

11. Boot Loader Configuration: A boot loader is used to select which operating system you would like to boot. The choices are Grub, Lilo, and none. If you select none you will need a boot disk to boot your operating system. Lilo has been a round for a long time and is well tested. Grub is a newer boot loader for the Linux operating system. For this document we will be selecting Lilo. Select Lilo and then select the "ok" button.
12. On the next menu it asks where you would like the boot loader installed. The most convenient place is on the Master Boot record. Select that option and then select the "ok" button to continue.
13. The following string asks if there are additional kernel parameters that you need to pass. Most likely you won't, but If you have a specific piece of hardware that you know will need them. This is where you enter them. They can also be added later if needed.
14. Finally the boot loader configuration utility asks what you would like the label to read to boot your system. The default is "linux" If you want to be more specific, such as listing the kernel version used, this is where you want to do it. Continue when you are finished with this option.
15. Network Configuration for eth0: Eth0 is the device name for the first Ethernet card in the system. There are options to configure this system as a DHCP node or with a Static IP address. We are going to leave the default for now, but turn off "the activate on boot". There are some things that need to be done to this system to make it secure before we place it on the open network. Select the "ok" button when finished.
16. Firewall Configuration: This section lets you set the starting point for the ipchains/iptables firewalls that come with Redhat. The default is medium protection. We will be modifying the values for them but the medium is a good point to start with. Select the "ok" button when finished.
17. Language Support: Additional languages that will be used on this system can be selected here, go through and highlight any of them that you think you will need. Select the "ok" button when you are done.

18. Time Zone Selection: Select a location nearest to you to set the time zone on your system. Select the "ok" button when complete.
19. Root Password: Set the starting root password. This password needs to follow some best practices rule. A combination of random upper/lower case characters, numbers, and symbols should make it fairly secure.
20. Add User: Don't add a user at this time. We need to get the system secure and setup some skeleton configuration files for them, before we add any users.
21. Authentication Configuration: It's a good Idea to use the Shadow password system which keeps the real password hash's out of the world readable "/etc/passwd" file. Using the md5 password requirement just adds an additional layer of security to the way the passwords are encrypted on the system. The MD5 hash is more secure than the crypt () function that would normally be used by the computer. The NIS, LDAP, and Kerberos configuration should be configured by hand so that you know exactly what is being put into the configuration. Select the "ok" button when finished.
22. Package group support: As a general/development workstation we will be selecting the following packages. Printing Support, Classic X Window System, Xwindow System, KDE, Sound and Multimedia Support, Network Support, Messaging and Web Tools, Graphics and Image Manipulation, Authoring and Publishing, Emacs, Utilities, Software Development, and Kernel Development. If you are not planning on doing any programming or kernel development, you can save a lot of disk space by eliminating the software and kernel development packages. Select the "ok" button when complete
23. Video Card Configuration: This screen presents you with what the installer thinks are the best settings for your video card. Change these if you need to. Select the "ok" button when complete.
24. Installation to begin: This states that a complete log of everything that is being installed will be in "/root". We will look at this log to check for problems after the installation. Select the "ok" button when complete.
25. Boot Disk: This screen wants to know if you would like to create an emergency boot disk. This disk is useful if your hard disk should become unbootable. I would recommend doing this. Select the "yes" button to continue.
26. Monitor Configuration: Select the appropriate settings here for you monitor. You should be able to get these from the manual.
27. X Configuration: Choose a color depth and resolution that you like. We are going to us KDE as our default desktop, and we would like to boot to the Graphical selection. I would recommend using the test feature to make sure that the settings work correctly.
28. Complete: Ok we made it through the basic install. Remove the floppy disk and CDROM, then reboot.

Installation of errata

The first thing we need to do is install the errata or updates. This can be downloaded from ftp.redhat.com. Download these onto an existing computer and burn them to a CDROM. Install the files doing the following.

```
[root@workstation root]# mount /mnt/cdrom
[root@ workstation root]# cd /mnt/cdrom
[root@ workstation cdrom]# rpm -Fvh *.rpm
Preparing...                ########################################### [100%]
   1:glibc-common           ########################################### [  2%]
   2:glibc                  ########################################### [  4%]
   3:cpp                    ########################################### [  6%]
   4:dateconfig             ########################################### [  9%]
   …

  43:util-linux             ########################################### [100%]
[root@ workstation root]# cd /root
[root@ workstation root]# umount /mnt/cdrom
[root@ workstation root]# eject
```

This can take a while so be patient. The flags on rpm are:

-F this Freshens the install. This means it only installs the rpms that have a corresponding application installed. If you use the –U for upgrade it will install the application if its not already installed.

-v is for verbose. This tells you what its doing, otherwise it just sits there and I'm not always sure that its working.

-h is to print the hash marks on the screen (hash marks are the #'s that are drawn for progress.

Turning off services.

The next thing we need to do is turn off the unneeded and unused services. Redhat comes with a good command called chkconfig. This is located in /sbin. This command lists the current status of services on your system. What run level they execute at and additionally it lets you turn them on and off for specific run levels. The syntax is used as follows

```
[root@ workstation root]# chkconfig --list
keytable        0:off   1:on    2:on    3:on    4:on    5:on    6:off
atd             0:off   1:off   2:off   3:on    4:on    5:on    6:off
syslog          0:off   1:off   2:on    3:on    4:on    5:on    6:off
gpm             0:off   1:off   2:on    3:on    4:on    5:on    6:off
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
kudzu           0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

```
netfs           0:off   1:off   2:off   3:on    4:on    5:on    6:off
network         0:off   1:off   2:on    3:on    4:on    5:on    6:off
random          0:off   1:off   2:on    3:on    4:on    5:on    6:off
rawdevices      0:off   1:off   2:off   3:on    4:on    5:on    6:off
apmd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
ipchains        0:off   1:off   2:on    3:on    4:on    5:on    6:off
iptables        0:off   1:off   2:on    3:on    4:on    5:on    6:off
crond           0:off   1:off   2:on    3:on    4:on    5:on    6:off
anacron         0:off   1:off   2:on    3:on    4:on    5:on    6:off
lpd             0:off   1:off   2:on    3:on    4:on    5:on    6:off
ntpd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
portmap         0:off   1:off   2:off   3:on    4:on    5:on    6:off
xfs             0:off   1:off   2:on    3:on    4:on    5:on    6:off
xinetd          0:off   1:off   2:off   3:on    4:on    5:on    6:off
rhnsd           0:off   1:off   2:off   3:on    4:on    5:on    6:off
autofs          0:off   1:off   2:off   3:on    4:on    5:on    6:off
nfs             0:off   1:off   2:off   3:off   4:off   5:off   6:off
nfslock         0:off   1:off   2:off   3:on    4:on    5:on    6:off
nscd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
identd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
radvd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
snmpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
snmptrapd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
vncserver       0:off   1:off   2:off   3:off   4:off   5:off   6:off
yppasswdd       0:off   1:off   2:off   3:off   4:off   5:off   6:off
ypserv          0:off   1:off   2:off   3:off   4:off   5:off   6:off
ypxfrd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
xinetd based services:
        chargen-udp:    off
        chargen:        off
        daytime-udp:    off
        daytime:        off
        echo-udp:       off
        echo:   off
        services:       off
        servers:        off
        time-udp:       off
        time:   off
        kotalk: off
        ktalk:  off
        finger: off
        rexec:  off
        rlogin: off
        rsh:    off
        ntalk:  off
```

```
talk:   off
telnet: off
rsync:  off
```

As you can see there are quite a few services on this system by default. Most of them are off, but a few that are a cause for concern are on. I think we should turn off the following:

Sendmail – this is the system that handles incoming mail. Since this is a users workstation it doesn't need to receive incoming mail. It will still be able to send and check email. It is just one less hole for users to get into.

Lpd – we are not going to have a locally attached printer to this device. By turning off lpd we will avoid issues with print spool exploits.

Portmap – this is the rpc listener daemon. We don't need this since we are not going to run any rpc daemons.

Rhnsd – this is redhats automated updater. I don't think its safe to let the system autoupdate from a remote site. It is safer to download and install the errata by hand or configure your own system that gets rpms from a local known secure site.

Nfslock – This is not needed since we will not be running nfs.

Sshd – turning this off doesn't affect the users ability to connect out. It just prevents outside access.

Ipchains – this is the old firewall from the 2.2 kernel time. We are going to be using iptables, which comes with the new kernel 2.4; therefore we will turn this one off.

The syntax to turn these off is as follows:

```
[root@ workstation root]# chkconfig --level 123456 sendmail off
[root@ workstation root]# chkconfig --level 123456 lpd off
[root@ workstation root]# chkconfig --level 123456 portmap off
[root@ workstation root]# chkconfig --level 123456 rhnsd off
[root@ workstation root]# chkconfig --level 123456 nfslock off
[root@ workstation root]# chkconfig --level 123456 sshd off
[root@ workstation root]# chkconfig --level 123456 ipchains off
```

Again run the chkconfig command to list the services and verify that the ones that you wanted to turn off have done so. At this point a reboot is probably a good idea, it will make sure the daemons are off and make sure that all updated applications are functioning properly.

IPTables.

The next thing we need to configure before connecting this system to the network is iptables. Iptables is the new stateful[1] firewall that comes with the 2.4 kernel. Stateful means that it remembers which packets it has sent and can modify its rules accordingly sending of DNS requests is an example where this is helpful. When the packet is sent out, iptables remembers this and allows the return packets to come back in.

By default iptables comes with three built chains: INPUT, OUTPUT, and FORWARD you can add additional chains if you like, but these cannot be removed. Some useful flags[1] for iptables:

-N Creates a new chain.
-X Deletes an empty chain.
-P Changes the policy on a builtin chain.
-L Lists the rules in a chain.
-F flushes the rules out of a chain.
-Z zeros the counters on the rules.
-A appends a new rule to a chain.
-I inserts a new rule at a chosen spot.
-R replaces a rule.
-D deletes a rule.

The first thing we need to do is list the current rules in the tables. The proper syntax is as follows:

```
[root@ workstation root]# iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

[root@ workstation root]# iptables -L OUTPUT
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

[root@ workstation root]# iptables -L FORWARD
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

[root@ workstation root]#
```

As you can see there aren't any rules in the table by default. Here are some real quick rules that will deny all incoming traffic but allow outgoing traffic[1].

```
iptables -N block
iptables -A block -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A block -m state --state NEW -i ! eth0 -j ACCEPT
iptables -A block -j DROP
iptables -A INPUT -j block
iptables -A FORWARD -j block
```

```
iptables-save > /etc/sysconfig/iptables
```

This should drop all new inbound packets but allow outbound and inbound packets that are related to the outbound packets to connect. The last line saves the rules to the iptables config file so that they are persistent. If you don't do the last command the rules will be lost on a reboot. Notice that we create a new chain named "block" and have the INPUT and FORWARD chains forward to it. This allows us to have only 1 rule set to deal with and have it affect both of the original chains. To test this it should be safe to enable the network and initialize the network connection. Edit the /etc/sysconfig/network-scripts/ifcfg-eth0 with your favorite editor. Change the line that reads:

ONBOOT=no

To read

ONBOOT=yes

Then restart the computer.

Try pinging the system or otherwise connecting to it from another workstation. You will see that the connections are dropped. You don't even get a refusal message. This is the way it should be to protect your system from outside intrusion. Additionally make sure that all of your outbound connections are working such as telnet, ftp, ssh, ping.

TcpWrappers Configuration

Tcpwrappers is an ipaddress based security tool. You may be wondering why we would need this if we have a successful iptables configuration running. It is always a good idea to have a layered defense, especially considering this system is going to be on an open network with no additional security beyond itself. Tcpwrappers comes with the RedHat Distribution and is installed with the installation that we selected above. The configuration files for tcpwrappers are /etc/hosts.allow and /etc/hosts.deny. These files are read in that order as well. It checks the hosts.allow to get a list of its good hosts and daemons. Then checks the hosts.deny for a list of the bad hosts. Finally all others are defaulted to deny. Most system administrators configure this to do a default deny by putting a blanket deny statement in hosts.deny and then explicitly entering the friendly hosts in hosts.allow. For our workstation we will be initially using a complete deny system. We will be opening up sshd and nfsd eventually. To deny all hosts we place :

ALL:ALL

In the hosts.deny file. This denies ALL deamons to ALL hosts. In the hosts.allow file to allow a daemon such as sshd we would add this entry.

Sshd:ALL

This allows traffic to sshd from ALL if you wanted to restrict this as most people would we would replace the ALL with either the ipaddress or a subnet of address to allow. For example:

Sshd: 111.222.333.

This would allow connections from all machines in the 111.222.333 domain.

An example host.deny might look like this

```
#
# Hosts.deny   This file describes the names of the hosts which are not
#              allowed to use the local INET services as decided by the
#              /usr/sbin/tcpd server.

ALL:ALL: banners /etc/banners
```

This file denies ALL deamons to ALL hosts. The "banners /etc/banners" on the end of the line is an additional nice touch which displays the /etc/banners file to the connection as its being refused. It's a good idea to put your access policy or the policy from your legal group in this file.

An example hosts.allow might look like this

```
#
# Hosts.allow   This file describes the names of the hosts which are
#               allowed to use the local INET services as decided by the
#               /usr/sbin/tcpd server.

Sshd: 111.222.333. : banners /etc/banners
```

This file allows access to the sshd deamon from the 111.222.333. subnet and displays the banner to them.

LogSentry

LogSentry, which used to be called logcheck is a very nice application for automating the analysis of the log files on your system. These files can get very large and its easy to miss something important. LogSentry runs through cron on a schedule set by the admin. It goes through the /var/log/messages file looking for specific keywords and reports these lines in a concise well formatted report via email. To get LogSentry go to http://www.psionic.com/products/logsentry.html and download it. It's licensed under the GPL so it can be freely distributed as needed. The installation procedure is pretty straightforward and there is a good document included named INSTALL. We will go through the procedure.

1. Extract the archive into a temporary directory on your system.

2. We need to make a slight modification to the /etc/syslog.conf. This file is used to handle how the syslogd daemon logs information. In the configuration that RedHat distributes it logs a lot of information, but splits it into multiple files such as /var/log/secure, /var/log/maillog, and /var/log/cron

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

The modification we have to make is to the line that logs to /var/log/messages. We want to have it log all messages to this file so we will do the following.

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info                                                  /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
```

```
*.emerg                                         *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                  /var/log/spooler

# Save boot messages also to boot.log
local7.*                                        /var/log/boot.log
```

Be very careful when editing this file, spaces can cause problems. You must use tabs in this file. After this change you will need to restart the syslogd with the following command;

/etc/init.d/syslog restart

3. Edit logcheck.sh, which is in the systems/linux folder, and find the CONFIGURATION SECTION.
   Set the SYSADMIN to the user or email that you would like to have it logged. Root is the default and a good idea.

4. Exit from the file return to the root folder and type: make linux

5. We need to test the system so execute /usr/local/etc/logcheck.sh and then check the root accounts mail for a reply from logcheck.

6. Edit your crontab to run the application automatically. Open /etc/crontab in your editor. And add the logcheck line to the end of the file:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
00,15,30,45 * * * * /usr/local/etc/logcheck.sh
```

The last line of this file is set to execute the script every 15 minutes. This is good time to start with. If you find that you are being overrun with email by this script change the times to hourly or daily.

The program should be functioning at this point. The files to configure this application are in /usr/local/etc. they are:

Logcheck.hacking — this file contains the keywords that the script uses to report explicit hacking. Be careful what

keywords you use in here. A blatant example that you wouldn't want to place in this file would be the keyword "failed". This would cause it to ignore any failed attempt to do anything. This is not a good idea.

Logcheck.violations — is used for keywords that report a security violation.

Logcheck.violations.ignore — is used to rule out likes in the logcheck.violations file.

Logcheck.ignore — is a list of keywords that are completely ignored by logcheck.

The README.Keywords file that comes with the distribution gives you an overview of where the default keywords came from and what you might want to tweak.

Port Sentry.

Port Sentry is a network socket listener and reporting agent. It sits patiently waiting for connections to your system and reports them to the address of your choosing. Additionally it can do some things to defend against possible hostile incidents. Why should we install this if we have iptables configured correctly? Two reasons, the first reason is to monitor and report connections that you would like to allow through iptables. The second reason is that no security system is foolproof, the more layers of security you have the more secure you are and the less likely you will have to clean up after a breach of security. On the event that you do have a problem with security, Port Sentry also gives you additional logging to review and attempt to piece together a trail.

Installation of this application is fairly straightforward. The first thing to do is aquire the application from http://www.psionic.com/products/portsentry.html. We will be using version 1.1 since the latest version 2.0 is still in beta and may still have several bugs. The steps to installation are as follows.

1. Unpack the archive into a temporary folder and read through the README.install file completely so that you have a good idea of where you are going as we go through the installation.
2. Open up the portsentry_config.h file and we are going to check the values of several default variables used by the system.
   CONFIG_FILE - this is the path to where the config file should be stored. The default location should be fine unless you are running low of disk space and would like to put it elsewhere.
   WRAPPER_HOSTS_DENY – This is the location of the hosts.deny file for tcpwrappers. Don't change this.

SYSLOG_FACILITY – The syslog facilities that we are going to use, the default is to report messages to the default system messages. We are going to change this to LOG_LOCAL0 so that we can log Port Sentry entries to its own file. This will make for easier analysis.

SYSLOG_LEVEL – is the level of message that Port Sentry will send messages as. Don't change this.

3. Open the /etc/syslog.conf and add the local0.* line to the end of the file.

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                         /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info                                          /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                      /var/log/secure

# Log all the mail messages in one place.
mail.*                                          /var/log/maillog


# Log cron stuff
cron.*                                          /var/log/cron

# Everybody gets emergency messages
*.emerg                                         *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                  /var/log/spooler

# Save boot messages also to boot.log
local7.*                                        /var/log/boot.log

# Port Sentry logs
local0.*                                        /var/log/portsentry.log
```

Make sure that you don't put any spaces in this file, use tabs.

4. Execute "/etc/init.d/syslog restart". This will restart the syslog daemon with the changes that have been made. Type "touch /var/log/portsentry.log" to create the log file.

5. Open the portsentry.conf. We are going to look at its settings and make changes as necessary.

   TCP_PORTS and UDP_PORTS – These are the ports that Port Sentry listens on and reports connections and scans of. We are going to change the default settings from the "aware" settings to the "anal" settings because of the openness of our network. Remove the # in front of the two entries for the "anal" settings and put #'s in front of the two entries of the "aware" settings. If this gives you too much

reporting then change the settings, but be aware that if you listen on fewer ports than you are less secure.

ADVANCED_PORTS_TCP and ADVANCED_PORTS_UDP – this tells the system which ports to watch for stealth scans. The manual states that the 1024 ports is a good setting. We will be changing this to 65535 which is the total ports on the system. Because the network is wide open we need to watch all of these ports. The iptables should prevent a lot of the random false positives associated with setting this.

ADVANCED_EXCLUDE_TCP and ADVANCED_EXCLUDE_UDP – these variables are a comma-separated list of explicit ports that Port Sentry will not report on. If we weren't running ipchains to block all incoming traffic we would probably add stuff such as port 139, which is the netbios port that windows clients often poll when browsing the network. We will be setting these blank because we want to know if anything is getting through iptables.

IGNORE_FILE – the path to the file that contains the address of nodes that Port Sentry thinks should be ignored. No need to change this entry unless you plan to install Port Sentry in an alternate location.

HISTORY_FILE – this path is to the Port Sentry history file. Don't change it.

BLOCKED_FILE – the path to the file that contains a list of addresses that are being blocked by Port Sentry. Again, leave this alone unless you are installing it in an another location.

RESOLVE_HOST – a "1" entry tells Port Sentry to do DNS resolution on remote hosts. Set it to "0" if your DNS server is particularly slow.

BLOCK_UDP and BLOCK_TCP – a "1" in this entry tells Port Sentry to block scans of UDP and TCP. We will be blocking scans, no sense in letting the bad guys see which ports we have open if they've already made it past iptables.

KILL_ROUTE – this is where we tell the system which way to kill a scanning connection. We will be using the "Generic_LINUX setting. The documentation recommends using the ipfw application to kill the system but if the bad guys have managed to get past iptables already then I think its good to fall back to the old system calls like /sbin/route. So remove the # in front of the "generic Linux" setting.

KILL_HOSTS_DENY- this configures how PortSentry will alter tcpwrappers config files. The default is

"ALL: $TARGETS"

We are going to alter this to additionally display our system banner

"ALL: $TARGETS: banners /etc/banner

KILL_RUN_CMD_FIRST – This can be used to execute a command before the connection is killed. Please don't put any retaliatory commands in here.

SCAN_TRIGGER – This sets the threshold at which we start blocking scans. I recommend setting this to "2" otherwise you could be blocking connections that you want to be able to connect.

PORT_BANNER – this entry allows you to display a banner to the bad guys. Letting the bad guys know that you are there is a bad idea. It is better to just drop the connection and leave them guessing.

Save and Close the file.

6. Compile the application. Type the following commands: make Linux and make install.

```
[[root@ workstation portsentry-1.1]# make linux
SYSTYPE=linux
Making
cc -O -Wall -DLINUX -DSUPPORT_STEALTH -o ./portsentry
[root@lusin portsentry-1.1]# make install
Creating psionic directory /usr/local/psionic
Setting directory permissions
Creating portsentry directory /usr/local/psionic/portsentry
Setting directory permissions
chmod 700 /usr/local/psionic/portsentry
Copying files
cp ./portsentry.conf /usr/local/psionic/portsentry
cp ./portsentry.ignore /usr/local/psionic/portsentry
cp ./portsentry /usr/local/psionic/portsentry
Setting permissions
chmod 600 /usr/local/psionic/portsentry/portsentry.ignore
chmod 600 /usr/local/psionic/portsentry/portsentry.conf
chmod 700 /usr/local/psionic/portsentry/portsentry
```

7. Startup and test Port Sentry. To start Port sentry type the following command

```
[[root@ workstation portsentry-1.1]# /usr/local/psionic/portsentry/portsentry –atcp
[[root@workstation portsentry-1.1]# /usr/local/psionic/portsentry/portsentry -audp
```

This launches Port Sentry in advanced mode. Check the /var/log/portsentry.log to verify that it started correctly.

8. Lastly we need to set Port Sentry up to run on a reboot so open your /etc/rc.d/rc.local file in your favorite editor and add the lines that start PortSentry to it. Reboot to verify that this works.
9. Lastly because Port Sentry sometimes blocks ports unintentionally, test the systems various network functions to make sure they are working correctly.

Sudo

Sudo is an application to grant privileges above and beyond the normal scope of access to normal users. Using sudo is a much more secure than giving the root password out to other users. Not only does it give you very fine grained support as to which commands you will let a user run, it also logs the use of the command in /var/log/secure. This is very helpful when trying to diagnose a problem with a machine that a user claims, "I did nothing".

Sudo comes preinstalled on Redhat 7.3 and is very easy to configure. The configuration tool that we will be using to do this is called "visudo" it opens whatever editor that you have defined in $EDITOR and then does a syntax check before it will let you save the file. I like to be very specific in how I grant permissions to users. That way revoking them is easier, but if you want to be less restrictive you can grant the user complete root access through sudo and it will give you a nice log to go back to when the user breaks something.

1. open the tool visudo by entering the command: visudo.
2. add a line at the end of the file that says:

   username ALL = /bin/mount /mnt/cdrom, /bin/umount /mnt/cdrom

   Make sure you replace username with the name of the user that you want to grant access to. This line gives the user the ability to mount and unmount the cdrom, which is ordinarily reserved for root.
3. Save the file and that's all there is to configuring sudo.

The manual pages are very extensive and give great directions to make any little changes that you would like to make.

Maintenance

Now that the system is installed and configured a maintenance plan is needed to keep it in top form. Such things as Backups, updates/patches, and log reviews will be a necessary part of keeping this system going. The following sections will discuss how to implement such things.

Backups

Backups are an essential part of maintaining the integrity and security of a system. With a good set of backups you have the ability to restore a system to a known secure state when you believe you have a security breach. Additionally with hardware failures you can easily get your system back to a functioning state. The type of backup system will depend greatly on the size of the system you need to backup. Its always a good idea to make a complete backup of the entire system before you turn users loose on it. This way you can use the image to install other systems with the same hardware without having to reconfigure everything from scratch. For our desktop system a simple tape backup drive connected directly to the computer makes for easy backups. To write the file system to a tape we are going to use the "tar" command. The syntax will be as follows:

```
[[root@ workstation portsentry-1.1]# tar –cvf /dev/st0 /
```

The flags chosen were the:

- -c This flag is to create a new archive.
- -v This flag is for verbose mode. This lets you know what its doing.
- -f This flag is states where to put the file as you can see its followed by the /dev/st0 which is the location of our tape drive (the first scsi device). This may vary depending on where you place the drive.

The final part of the command "/" is stating which volume to backup. Our system has multiple volumes so we will need to backup several of them. I have written a little script to do this:

```
#This script is a simple backup script that uses tar to backup all of the
#volumes on the system.

#backup /boot
mt –t /dev/st0 rew
tar –cvf /dev/st0 /boot
mt –t /dev/st0 rew

#backup /
mt –t /dev/st0 fsf 1
tar –cvf /dev/st0 /
mt –t /dev/st0 rew
```

```
#backup /var
mt –t /dev/st0 fsf 2
tar –cvf /dev/st0 /var
mt –t /dev/st0 rew

#backup /home
mt –t /dev/st0 fsf 3
tar –cvf /dev/st0 /home
mt –t /dev/st0 rew
mt –t /dev/st0 offline
```

This simple script backs each file system onto a separate volume of the tape. To restore the files we do the following commands.

```
[[root@workstation portsentry-1.1]# cd /tempfolder
[[root@workstation portsentry-1.1]# mt –t /dev/st0 fsf 1
[[root@workstation portsentry-1.1]# tar –xvf /dev/st0
[[root@ workstation portsentry-1.1]# mt –t /dev/st0 rew
[[root@ workstation portsentry-1.1]# mt –t /dev/st0 offline
```

Its important to restore files to a temporary area and not to the original location. By doing this you can pick the files that you need to restore without overwriting files that you do not need. The second line:

#mt –t /dev/st0 fsf 1

Fast Forwards the tape to the second volume on the tape, in our case the "/" volume.

Now that we have a working backup script we need to add this to the /etc/crontab file to execute every night.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
00,15,30,45 * * * * /usr/local/etc/logcheck.sh
30 23 * * * /usr/local/bin/backups
```

The final line of the crontab file has been modified to run the backups every night at 11:30pm.

We will be rotating the tapes on a bi-weekly basis. If you feel you might need longer tape life then buy more tapes and implement the system as needed.

Updates

Keeping this system up to date will be a key to making sure it's secure. By being open to the world it will be under attack constantly by "Bad Guys" looking for the latest vulnerability. Redhat has an excellent website http://rhn.redhat.com/errata/rh73-errata.html that is an up to date list of errata for the Redhat 7.3 distribution. This site should be checked frequently (a good recommendation would be at least once a week) and new updates applied to the system as fast as possible. Redhat does have a utility on the system called "up2date" which can install the updates automatically, but I feel that its safer to know what your installing so that you don't accidentally overwrite something critical. The procedure for installing these files is the same as the procedure used to install the original errata at the beginning of the setup.

Log Review

Log review on this system is very important. You will have to be extremely vigilant in watching for anything that is out of the ordinary. By installing Log Sentry and Port Sentry you have given yourself a great advantage in doing this. Reviewing the reports from LogSentry and PortSentry should become a daily routine just like drinking your morning coffee. Additionally you may want to store the reports somewhere either in hard copy on a separate media so that if the system is compromised you will have a history to go back to.

Configuration Verification

The final step in this process is to verify that everything is working and secure. We will first proceed by testing each layer of defense that we put on the system. After we are sure that the machine is well protected we will then verify that it does what we initially stated it would do.

The first layer that needs to be tested is the iptables layer since it is the outermost layer of the system. To test this from the outside we will be port scanning the system using nmap. To test it from the inside we will use the Netscape browser to connect to the outside world.

1. The first thing that we should do before verification is to check RedHat's errata website to see if there is any new errata. The address is http://rhn.redhat.com/errata/rh73-errata.html. The reason for doing this is to ensure that we have all of the most recent security updates. We do this first so that our security confidence isn't altered after we have verified that it is secure. We should redo this verification any time we install or upgrade software on the computer so that we can be re-assured that the system is still secure. To install these updates use the procedure in above for installing errata.
2. Nmap can be downloaded from http://www.insecure.org/nmap/nmap_download.html. The installation

instructions are pretty straight forward. Once you have it installed issue this command:

```
[[root@ workstation portsentry-1.1]# nmap –sT –sU –sR
111.222.333.444
```

3. The command flags used with nmap are as follows:
   - –sT Tells nmap to scan for tcp ports
   - -sU Tells nmap to scan for udp ports
   - -sR Tells nmap to scan for rpc ports

This scan may take a while so be patient.

4. The output of the file looks like this:

```
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap/ )
All 3013 scanned ports on workstation (111.222.333.444) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 3510 seconds
```

As can be seen iptables did its job and dropped all connection attempts as if they are not there.

5. the next thing we are going to do is disable iptables and make sure that the next layer of defense is working, namely PortSentry.
   To disable iptables on the system do the following:

```
[[root@ workstation portsentry-1.1]# /etc/init.d/iptables stop
```

6. Execute the nmap command that was used to scan the system earlier. Look through the /var/log/portsentry.log. You should see the entries from the scans. Additionally check the /usr/local/psionic/portsentry/portsentry.blocked.* files to verify that the system has properly blocked the attacking system. If all went well then we should re-enable iptables by the following command:

```
[[root@ workstation portsentry-1.1]# /etc/init.d/iptables start
```

7. Checking log sentry: The easiest way that I have found to test LogSentry is to create a normal user account. Log in to that account and attempt to "su" to root. This will trip the LogSentry Script. Additionally you should have gotten a report from LogSentry regarding the port scan that PortSentry detected.
8. The next thing we are going to do is check to make sure we aren't listening on any ports that we shouldn't be. To do this we are going to use the command "lsof".

```
[root@ workstation log]# lsof -i
COMMAND PID USER   FD   TYPE DEVICE SIZE NODE NAME
dhcpcd  456 root    4u  IPv4    795       UDP *:bootpc
X       832 root    1u  IPv4  40207       TCP *:x11 (LISTEN)
[root@PPD54048 log]#
```

The "-I" flag that was used with lsof gives us the Internet protocol ports that we are listening on. As you can see, we are listening on 2 ports. The dhcpcd port is needed by the dhcp daemon. If we were not using dhcp this port would not be open. Additionally the Xwindows server is listening. This is needed as well to allow remote X applications to be executed. So it looks like at this point our system is in good shape.

9. The next step after we are sure that the machine is secure from the outside is to make sure that all of the services that we need to use still work. It wouldn't do us any good to lock it down so securely that the system is useless. We decided at the beginning of this document that this system would be used for browsing the web, checking email, connecting to remote systems, and running remote Xwindows applications. So lets test each of these to make sure that they still work. To do this run Netscape and verify that you can connect to some remote web page (This works on the test system for this paper), configure your favorite email program to check an external email account and verify that it works, use ssh to connect to a remote Unix system and run an application such as emacs or xclock. These tests should verify that the system is functioning properly.

Conclusion

At this point the system should be secure enough to turn it loose to a user on an open network or any network for that matter. There are other things that could be done to the system such as restricting the iptables even further so that only certain protocols are allowed out of the system. This puts a heavy burden on the admin and can often frustrate the user. At most facilities the job of the admin is to facilitate the users ability to get his job done, therefore making the system less user friendly would seem counter productive. As always, the system is only as secure as the admin is vigilant. I think it's a good idea to watch such websites as http://www.securityfocus.com and http://www.slashdot.org to see when

new security threats are possible. The biggest thing to help insure security is communication. If your facility has a central security group, be sure to communicate with them anything that you are suspicious of. I hope this document has been helpful. Please feel free to contact me if you need further assistance.

Glossary

Open Network	A network scheme that is open to the Internet with very little if any filtering between the users workstation and the rest of the hostile world.
Critical System	A computer system that if it becomes non-functional would affect the day-to-day operation of the organization.
Stateful	A term used to describe the way a firewall will remember when it sends a packet to a given site so that it will let the return packet back in without any additional configuration.

References

1. Linux 2.4 Packet Filtering – HOWTO at linuxsecurity.com http://www.linuxsecurity.com/resource_files/firewalls/packet-filtering-HOWTO/index.html
2. Security Focus – Home of the valuable Bugtraq mailing list. http://www.securityfocus.com
3. Redhat Errata: Security Alerts, Bugfixes, and Enhancements site - http://www.redhat.com/apps/support/errata/
4. Nmap website - http://www.insecure.org/nmap/
5. Exploit world Linux site - http://www.insecure.org/sploits_linux.html
6. GIAC GCUX Manual – A wealth of knowledge here.

Last Updated: October 22nd, 2017

Upcoming Training

SANS 2018	**Orlando, FL**	**Apr 03, 2018 - Apr 10, 2018**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**