Global Information Assurance Certification Paper

Deploying Redhat 7.1 as a platform for Enterprise Security Management

William Letourneau
October 20, 2001

SANS GCUX Practical Assignment - Version 1.7 (revised August 13, 2001)
Securing Unix Step by Step

Table of Contents

Introduction

This document has been prepared as part of the certification requirements for the SANS GIAC Certified UNIX Security Administrator program. The first section provides a brief description of the security issues relevant to deploying a Linux server as a platform for a Java-based enterprise security management application. The second section gives instructions detailing how to secure Redhat Linux 7.1 from out-of-the-box to a secured, deployment ready state.

The purpose of the server in question will be for use with an enterprise security management application (which I will refer to in this document as ESM); however in the more general sense it can be used as a guide to securely running an arbitrary network available application in a chroot jail under Redhat 7.1.

In the spirit of the SANS sanitization guidelines the name of the application in question has been changed to Enterprise Security Manager or simply ESM. Not to be confused with Enterprise Security Policy Administration, more information on Enterprise Security Management can be found at http://www.itactics.com/pdfs/ESM_White_Paper.pdf. For the purpose of this document we need only know that ESM is a java based application which handles security event normalization, correlation, filtering, log consolidation, and so on. It is part of a suite of applications that provide a wide range of security management functionality, however we will be focusing only on the secure deployment of the 'ESM' Central Server application running under Redhat 7.1.

General Security Considerations

Policies and Procedures

Even a server that has been properly secured and patched, with a hardened operating system, a host based firewall, and a file integrity checker can be rendered completely vulnerable by a lapse in helpdesk policy or sloppy password practices by the administrator.

While a detailed examination of effective security policies is beyond the scope of this document, it is important that they be considered when planning the rollout of your new server. Most of this section should be redundant with your existing security policies and will serve only as a reminder to ensure that your ESM deployment fits well within existing standards for your organization.

If on the other hand you are an administrator in an organization with no clearly defined security policies, now is a good time to start putting some in place - the following list should give you a starting point for issues you will want to consider.

Defining Security Policy

- Not all security problems have a technical solution, some can only be addressed by effective policies
- Reference consensus documents such as http://www.faqs.org/rfcs/rfc2196.html or http://www.sans.org/newlook/resources/policies/policies.htm
- It is important that all stakeholders understand security and follow related policies
- Publish your security policies and make users acknowledge them
- Conduct security awareness quizzes and compliance audits
- There may be legal requirements for your industry relating to security policies; for example government, health care (HIPPA), finance, and so on…

Password Policies

It is important that your users and administrators alike understand the importance of choosing good passwords, and handling their passwords with the same care that they would handle their bank PIN or passport. Their password is their digital identity, and they are responsible for what happens with it.

- Good passwords contain a mix of upper and lower case letters, have at least one number, and at least one symbol.
- It should be at least 6 characters long on systems limited to 8 character passwords. On MD5 enabled Linux systems, it should be at least 8 characters long (preferably much longer).
- The more complex the password, the longer it will take for an attacker to guess it using brute force methods.
- Never choose passwords which are derived from your name, login ID, machine names, birthday, family members, pets, and so on.

- Never use passwords that are derived from common words in your environment – including anything in the dictionary, technical jargon common to your work, a second language you speak, and so on.

If possible, implement technologies that enforce good password choices when users pick them, and educate users as much as possible about the importance of passwords, including their responsibilities. Better still investigate more secure technologies such as two-factor authentication (something you have + something you know), one-time password systems, Public Key authentication, and so on.

Creating Security Procedures

Define detailed Incident Handling and Forensic Audit plans ahead of time

- Ensure that the right people are aware of these so they can be managed according to procedure from the moment you become aware that security has been breached.
- This is especially important if you intend to prosecute intruders

Security Notices/ Warning Banners

- Having posted "authorized use only/ no expectation of privacy" banners may also be important if you wish to prosecute
- Modify boilerplates (and run them past your legal department)

Backup and recovery plans

- Using only your documentation and offsite backups, can someone else recover from the complete loss of your server and be back up and running quickly?
- Are your offsite backups in a safety deposit box or some other place where you cannot get to them outside of working hours?

Before you start – What are your requirements?

What is the purpose of the machine?

- If there are redundant or similar services, can one be eliminated?
- Is there a clear business rationale to the deployment?
- What does it need to do? (List your needs and your wants.)
- List the minimum functional specs the machine must have to be considered viable?

Weigh your risks

Security usually involves tradeoffs in terms of ease of use – so it is important to consider up front what level of security you want to implement. Factors to consider are:

- Availability - Who needs access, when, how often (uptime requirements, maximum acceptable recovery time), how do they need to get access (LAN, VPN, Internet), and so on.

- Integrity (system is safe from integrity breaches- crackers, viruses, disk corruption, user errors, natural disaster, and so on.)
- Confidentiality (Information is only available to intended parties)

You will have to balance ease of use (and hence "availability") against the measures you put in place to safeguard the integrity and confidentiality. After all, the most secure server is the one that is disconnected, unplugged and hermetically sealed in a bomb shelter – but you have to make sure that the system is not so "secure" as to prevent your users from doing their job.

Selection of Methods and Tools

- History shows us that security solutions which are derived from broad consensus and have weathered public scrutiny are more trustworthy and reliable than "Security through Obscurity" or Proprietary "Black Boxes".
- Some methods or tools will affect the balance between ease of use and security more than others
- Always investigate to see if alternatives exist
- Time and Skills: Is your plan feasible based on resources and time available? Does it make sense to hire an expert who can do it faster and train your people at the same time?

Costs vs. Benefits

It is hard to show that Security is a good Return On Investment to management – be prepared to make your case to management. Since Cost/ Benefits analysis of security looks poor on the surface try some of these approaches

- It's like an insurance policy on the data and services this machine offers
- It's like an insurance policy on the reputation of your company
- The cost of security is protection against the cost of losing these things
- Will you be legally responsible if a lapse in your security is used as a springboard to attack a third party?

Physical Security

Poor Physical Security can render all of your electronic measures useless. Some factors to consider:

- Limit who has access to the server room/ audit access (swipe card logs to your syslogs is ideal)
- Case locks
- Access to CD ROM/ Floppy/ Console/ Power Switch/ removable drive racks
- Detecting physical security breaches (reboots in your remote syslogs, visible signs of tampering)
- Hardware redundancy/ failover
- Redundant power/ internet
- Server room infrastructure failure alarms (for air conditioning, power, and so on.)
- Protection from Fire/ Theft/ Water (old Fire Extinguisher systems)/ Disasters

Technical Considerations

BIOS passwords

- Power on password - setting one may mean a trip to the office at 3am some weekend if your machine is rebooted or power cycled.
- Use a BIOS settings password and document your settings

Choose a Linux Distribution (To Redhat or not to Redhat?)

- Redhat continues to work hard to appeal to the corporate customer,
- it is the defacto Linux distribution of most "business class" applications (they tend to distribute installs routines as .RPMs and install instructions tend to be Redhat centric) however,
- many other distributions exist - some may be better suited to your particular needs.

Install in a sterile environment

- Document every step of your install - you might not be the guy doing it next time or you might not remember an important detail when you try to rebuild your server in six months
- The machine should be completely disconnected from the network and certainly not on the Internet.
- Validate your install media, especially if you downloaded it instead of buying it from a reputable distributor. In most cases this will involve checking the cryptographic checksums (such as MD5 hashes) of the .ISOs (CD images) or the digital signatures (such as GPG keys) of .RPMs (Redhat packages).
- Ensure that all operating system hardening and security measures are complete before connecting to network
- If testing is required, do it in an "isolated" lab, not on the production network.
- Partition your drives so that applications or services that may consume space "as needed" do not threaten the integrity of your other operating system functions (such as /boot and / "root") – other applications or services could also be impacted by filling a volume
- Is a journaling file system (such as reiserfs) appropriate? There are performance and reliability gains, but it is a new technology to Linux.
- Is an encrypting file system (such as tcfs – Transparent Cryptographic File System or DES/IDEA loopback devices) appropriate? There is a performance cost, but on systems accessed by many users, it may be worthwhile. (Also a good idea for laptops)

Install less not more

- Always choose the "custom install" – it may involve more work but it is the only way to assure yourself that you are not installing unnecessary packages.

- Install only those packages or services that you need - better to have to add something later (and gain an understanding of your dependencies) rather than install too much and expose yourself to vulnerabilities or bugs you didn't expect

Make two boot (recovery) disks:
It's a little extra hassle, but since floppies aren't terribly reliable or you might find yourself needing the disk that's stored with offsite.

- Keep one with your backups and one locked up onsite
- Change the boot manager (probably LILO) to only allow the desired option (You'll use your disk to get to any other options if there's an emergency)

The tools that Redhat forgot:
Install additional packages/ applications not provided by the vendor if there is a need that is not addressed by the distribution you select. You may also find that many Linux distributions do not keep right up to date on all the packages you are interested in – the reason for this is because big shops like Redhat do integration and system testing before bundling them in. The downside is that you might find yourself on the bleeding edge – updating one package to the latest and greatest might break another package you care about.

Address recent bugs and vulnerabilities:
The pace of change is such that there may well be a number of fixes to problems that have occurred since the distribution you just installed was officially released. It's important to not only do all updates recommended by your vendor, but to put a plan in place to keep up to date on patches.

Take advantage of Bastille-Linux – this automated hardening script will not only increase the security of your system in many important ways, but it serves as a tutor to the user along the way.

Once you finish configuring the server, take an initial full backup of the system and test the restore procedure now, while you still have the luxury.

Implementation

There is no perfect and complete guide detailing how to secure a system - security is a moving target and requires a commitment to keeping up-to-date on the latest patches and information. A network-connected system that is not being actively patched and maintained is probably insecure.

Since security technologies change quickly, you should check the last revision date on this document before implementing the following steps. For example, this document will potentially contain dangerously inaccurate information by the time the next major revision of Redhat is released.

Step-by-Step setup instructions

If you have copies of the Red Hat 7.1 install media from a trusted source - such as from a store bought package you can probably skip this step, (but it's still a good habit to get into). Take your .iso files and validate the MD5 checksums against those published on ftp.redhat.com.
You can do this at the bash prompt by typing:

```
md5sum seawolf-i386-disc1.iso
```

This will run for several seconds and return a checksum in the form of an alphanumeric cryptographic hash. If the hash value that you've just computed matches the published value, you've got a clean copy – if not, you may only be off by a single bit, but you don't have exactly what the vendor shipped.
Here are some values for Redhat 7.1 (codenamed "Seawolf") .iso files:

```
edc2d5e1ab6093e3d486cc38dc12511a  seawolf-i386-SRPMS.iso
596b1575773e88e066326f6741312a6f  seawolf-i386-disc1.iso
f27b912299572a542cd663b712444445  seawolf-i386-disc2.iso
59f3333435378fb1645700731c91bc54  seawolf-i386-powertools.iso
```

Also you will need the Bastille Linux hardening scripts, preferably downloaded and burned onto a CD from a known clean system. Bastille uses GPG to digitally sign their files rather than providing an MD5 checksum. You'll need their public key (which is found on their website) and here are the instructions lifted directly from their site:

```
gpg --verify Bastille-1.2.0-1.1mdk.src.rpm.asc Bastille-1.
2.0-1.1mdk.src.rpm
```

Expected Results:

```
gpg: Signature made Thu 14 Jun 2001 10:17:11 AM EDT using DSA key
ID CDCCEA5C
gpg: Good signature from "Jay Beale (Bastille Linux Release key)"
gpg: /home/jay/.gnupg/trustdb.gpg: trustdb created
```

System Configuration used

The following server hardware was used during the creation of this install guide:

> Abit BP6 motherboard, Dual Celeron 500MHz CPUs, 512MB RAM, 20GB 7200 rpm IDE drive, 3C905 10/100 NIC, TEAC 4x4x32 CDRW, 16MB Voodoo3 2000, 3-Button PS/2 mouse

In the likely scenario that your server hardware differs from this, ensure you check for compatibility with the Redhat 7.1 hardware compatibility list (http://hardware.redhat.com/hcl/genpage2.cgi)
This may also entail some minor modifications to these instructions - typically this involves only accepting the defaults for device drivers suggested by the installer.

Once your hardware is in place, ensure you are physically disconnected from the network by unplugging the network cable. The best place to configure your machine may be in your office, rather than on the production rack to ensure that the machine is under your control until you have finished hardening it.

The following instructions assume that you will be setting up a single machine, by hand. However, if you will be setting up several similar machines it makes sense to use ksconfig to create "kickstart" files to automatically perform the installs according to your specifications on each machine. ksconfig is found on the Redhat CDs or here is a link to the current version of ksconfig at the time of this writing – check rpmfind.net for the latest version:
`ftp://speakeasy.rpmfind.net/linux/rawhide/1.0/alpha/RedHat/RPMS/ksconfig-1.9.8-3.noarch.rpm`

Installing the Operating System

Power the machine on and enter the BIOS by pressing DEL when prompted.
Under the BIOS Features Setup menu, verify that the Boot Sequence is CDROM, C, A.
Under Password Setting, enter a unique BIOS password.
Put the Red Hat install media (disk 1) in the drive and then save and exit from the BIOS.

The machine will boot from the CD and begin the install routine. The following instructions have a colon (:) in front of steps you must take during the install.

Welcome to Red Hat Linux 7.1!
:enter (Install in graphical mode)
Language Selection
:next (english)
Keyboard Configuration
:next (Generic 105, English, Enable dead keys)
Mouse Configuration
:next (3 buttom mouse ps/2)
Welcome to Red Hat Linux
:next
Install Options
:Install > Custom System
:next
Disk Partitioning
:Manually partition with Disk Druid
Partitions
:Delete any partitions that are present
:Add

/boot	15M
/	2047M
/tmp	2047M
/var	2047M
/usr	2047M
swap	1024M
/home	(Use remaning space) 10338M

:next
Choose Partitions to Format
:check off all boxes, including check for bad blocks

:next
LILO config
:take defaults -> Create boot disk, Install LILO, on MBR /dev/hda, Linear mode, Kernel Parameters: hdc=ide-scsi, Default boot image, Boot label 'linux', root partition is /dev/hda5
:next
Network Configuration
:eth0, uncheck DHCP, leave activate on boot
:enter IP (192.168.0.2), netmask (255.255.255.0), network(192.168.0.0), broadcast(192.168.0.255), hostname (esm-cs), gateway(192.168.0.1), Primary (x.x.x.33) and Secondary DNS (x.x.x.34). Several fields will auto populate - ensure they are correct. (Obviously, you will want to replace my example IPs with the appropriate ones for your environment.)
:next
Firewall Configuration
:choose 'No firewall' (we'll set one up later)
:next
Language Support
:next (english)
Time Zone Selection
:select 'UTC Offset' = UTC-05 US Eastern, check Use Daylight Savings Time
:next
Account Configuration
:enter good root passphrase including mixed case, numbers and symbols (up to 256 characters)
:enter an account for yourself - again up to 256 character passphrase
:next
Authentication Configuration
:enable MD5, enable shadow, everything else disabled
:next
Selecting Package Groups
:deselect everything except X Windows, Gnome, KDE and Networked Workstation
:check 'select individual packages'
:next
Individual Package Selection
:Add the following
  Amusments > Graphics
        xlockmore
  Applications > Archiving
        cdrecord - for burning logfiles off to CD
  Applications > Editors
        vim-X11
        vim-enhanced
  Applications > Internet
        Lynx
        Mozilla

Mozilla-psm
nc (netcat)
tcpdump

Applications > Multimedia
cdda2wav - for xcdroast
xcdroast - for burning CDs

Applications > Publishing
xpdf

Applications > System
linuxconf
mkisofs - for burning CDs
mtools - to read DOS floppies
?nut - network UPS tools - install if you are on a UPS
?nut-client (if the UPS is serial attached to another machine)
procinfo - process info tool
rpmfind (manages your local rpm database and helps download)
screen - multiple logins on one terminal
sudo
symlinks - symlink checker
sysstat - system monitoring tool
tripwire - file integrity checker
vlock - console locker
xcpustate - system monitoring tool
xosview - system monitoring tool
xsysinfo - system monitoring tool

Applications > Text
dos2unix - file converter for DOS/MAC files to UNIX
unix2dos

Development > Debuggers
lslk - lists locked files (active inodes)
lsof - lists open files
ltrace - useful application debugger/dependancy checker
memprof - memory profiler/ mem leak checker
strace - useful application debugger
sysreport - reports current system hardware and configuration

Documentation
kdoc - KDE docs
man-pages - documentation

System Environment > Daemons
ntp - Network Time Protocol package including NTP daemon
openssh-server - Open SSH daemon
xinetd - secure replacement for inetd

System Environment > Libraries
gd - graphics library - needed for linuxconf

User Interface > Desktops
vnc - client for remote control application

vnc-doc
User Interface > X
XFree86-xf86cfg - X configuration tool
ttfonts

Unresolved Dependencies - none (screen does not appear)

X Configuration
:Voodoo3 (generic)
:next
:DDC Probed Monitor (BRG02ab)
:next
Custom Configuration
:16 bit, 1024x768, KDE, default to text login
:next
About to Install
:note - a complete log of what you installed will be in /tmp/install.log
:next
When prompted, Insert Disk 2.
:ok
Boot Disk Creation
:insert floppy labeled "hostname" boot disk
:next
Congratulations
:exit
:when CD ejects, remove your floppy and CD

When the machine reboots, enter the BIOS and set the Boot Sequence to 'C, CDROM, A', then save and exit.

Tripwire

Tripwire is a file integrity-checking tool that allows you to build a database of the cryptographic checksums of important files and their attributes. This allows you to quickly assess exactly which files an attacker may have modified, added, or deleted, with a high degree of confidence. The tripwire configuration file and checksum database are also cryptographically signed, making very difficult for an attacker to compromise those. As a side benefit, tripwire also serves to show you which files get changed or added on your system by install routines for software or other administrators who may have changed things.
Setting up tripwire is by and large out of scope for this document, but since it is such a useful tool, a brief primer is in order – however, I am intentionally glossing over a number of details that are important if you wish to use this tool to enhance your security and break-in readiness. Please take the time to read the man pages and associated documentation. This being said, the version of tripwire installed with Redhat 7.1 is very

easy to use and comes with a pre-generated configuration file which will monitor all important files on a system with all packages installed (the "everything" install).

To setup Tripwire:
```
/etc/tripwire/twinstall.sh
```
You will be prompted to enter passphrases for the site and the local machine.
Generate the initial database of checksums as follows:
```
/usr/sbin/tripwire --init
```
And that's it! (For our purposes over the balance of this document at least…)
Any time you wish to check the state of your system run the following command and it will generate a full report:
```
/usr/sbin/tripwire --check
```

Installing Bastille

The Bastille Linux package is a group of perl scripts designed to meet more than one objective - to educate a system administrator about a broad spectrum of Linux security considerations, and to make the requested security changes based on choices presented to the administrator. The next several pages paraphrase and expand upon the security considerations presented by the Bastille scripts for our particular installation. Despite the fact that some of the Bastille explanations are repeated in this document, it is very important to read through the hardening scripts in interactive mode the first time so that you understand what it is doing, and gain a better understanding of the underlying strategy behind each of the steps taken. This being said, if you are configuring many servers, you may want to use the automated hardening script mode - my advice is to generate your own config file (answer file) for use with this process so that the systems are secured as expected.

While you are working through Bastille, other questions may appear (or fail to appear) depending on what else you have installed on your system or if you have given different answers than those provided below.

As previously detailed, you have the necessary Bastille files burned on CD, and you've already checked the signatures.

Mount your CD and perform the following installs:
```
rpm -ivh perl-Tk-800.022-11.i386.rpm
rpm -ivh Bastille-1.2.0-1.1mdk.noarch.rpm Bastille-Tk-module-
1.2.0-1.1mdk.noarch.rpm
```

Once this is complete, start the process as follows:
```
/usr/sbin/InteractiveBastille
```

Title Screen
(A brief introduction to Bastille appears on the first screen.)
A: next

Firewall
(In this module we will be configuring and activating the native Linux iptables firewall, which is part of the modern 2.4 kernel installed on our system. We do this despite the fact that we do not route traffic between networks, but simply to act as a host based firewall as part of our "defense in depth" strategy for hardening the system - it allows us to filter out various kinds of potentially undesirable traffic right at the IP stack, before it can reach any services that may be running on our system. Many of the questions in this section are geared toward the packet filtering ipchains firewall provided older Linux kernels but are irrelevant to the stateful inspection iptables firewall. The reason for the apparent reduction in complexity is that unlike packet filtering, stateful firewalls maintain information on each session passing through the firewall. The most obvious result of this is that a stateful firewall will know if an incoming packet is actually a legitimate response to an outbound request or if it is an unsolicited packet which should be discarded - this reduces the complexity of the rules set while increasing security since you no longer have to leave holes in your firewall for incoming responses.)
Q: Would you like to run the packet filtering script?
A: yes, next
(An explanation of notation appears.)
A: next

Q: Do you need the advanced networking options?
(Since we are not routing traffic and this is only for use as a host-base firewall, we will not require advanced options such as routed, and Network Address Translation.)
A: no, next

Q: DNS Servers
(With a packet filtering firewall, this would allow you to define a list of DNS servers that this host could receive responses from - since our firewall is stateful, this question is irrelevant because only DNS traffic which comes in response to a request from this host will be allowed in)
A: <blank>, next

Q: Public Interfaces
(At this point we need to list all network interfaces that are public/untrusted - in a typical firewall the untrusted interfaces would be those connected to the internet while the trusted interfaces would be those on the inside (corporate) network. In our case we are protecting the host by taking the paranoid stance that all interfaces are untrusted, and applying firewall rules to all of them. We will even apply the firewall rules to interfaces that do not exist now, but could be added later since this incurs no additional resource cost.)
A: eth+ ppp+ slip+, next

Q: TCP Services to Audit
(This question allows us to audit, or monitor, any attempt to connect to our host on any number of ports commonly used to attack computers over the network. We wish to

monitor this type of activity even though we will not be vulnerable to most of this activity simply because it serves as an alarm that someone is trying to 'check all your doorknobs' to see if they can find any unlocked doors. The default list suggested is a good set - we choose to audit all connections to the ssh port even though some of it may be legitimate since it is still a good control to have in place to log whenever the machine is administered over the network.)
A: telnet ftp imap pop-3 finger sunrpc exec login linuxconf ssh

Q: UDP services to audit
(The same explanation as the previous question applies, only this time to UDP services. The default choice of port to monitor underlines the idea of monitoring ports for attack even when you know you are not vulnerable; the suggested port is for a common Windows trojan - Back Orifice - which would never be on our system.)
A: 31337, next

Q: ICMP services to audit
(The same explanation as the previous two questions apply - however, in this case we will choose not to audit the 'obvious' traffic of echo-requests (pings) since it is much more likely that your logs will fill up with legitimate requests than you catching an attacker who pings your machine to see if it is on the network.)
A: <none>, next

Q: TCP service names or port numbers to allow on public interfaces
(This question asks us to list those ports on which we will explicitly allow unsolicited (externally generated) connection attempts. Our Enterprise Security Manager will be listening on 25 (mail) and 13337 (intra-ESM chatter), and we will be running an OpenSSH daemon (server) on port 22 for remote administration. We are not interested in getting any other externally motivated TCP traffic.)
A: 22 25 13337, next

Q: UDP service names or port numbers to allow on public interfaces
(The same logic as the previous question applies - we need ESM to be able to listen to UDP traffic on 162 (SNMP traps), and 514 (remote syslogs) but are not interested in any other UDP traffic)
A: 162 514, next

Q: Force passive mode?
(We are being asked if we should allow FTP client requests originating from this host to use regular (active) FTP mode or passive mode. The question itself is irrelevant because the technical consideration in question does not apply to our stateful iptables firewall, but it should be noted that either way FTP is just a bad protocol from a security perspective - starting with clear text passwords to a long list of vulnerabilities and exploits in many common FTP servers. When possible avoid the use of FTP and instead use SFTP over SSH to trusted hosts, or HTTP downloads from public file servers.)
A: no, next

Q: TCP services to block
(This is another question that makes no difference to us because of the advantages of iptables. For clarity, we will erase the defaults)
A: <blank>, next

Q: UDP services to block
(Again, this question is irrelevant due to iptables. We will erase the defaults for clarity.)
A: <blank>, next

Q:ICMP allowed types
(Since the stateful nature of iptables does not extend to ICMP, we need to explicitly state which types of incoming ICMP are allowed - the default list will allow you to ping and traceroute as well as receive 'destination-unreachable' messages if there is a problem when you attempt to connect to a remote host.)
A: destination-unreachable echo-reply time-exceeded, next

Q: Enable source address verification?
(This asks if we would like to take measures to stop traffic likely to have spoofed (fake) IP addresses - a common trick of attackers.)
A: yes, next

Q: Reject method
(This question asks how we should handle traffic we wish to block - do we 'reject' it and let the remote host know that we are blocking their attempt, or do we 'deny' it and simply ignore the connection attempt and let the remote host give up their connection attempt after hearing no response for some time. Either way we are not invisible to the network, since we are offering up public services on various ports, and have decided to be responsive to ping requests. Everything else being equal, we will take the more polite option of rejecting traffic, since most of the time it will be non-malicious traffic which simply hit us by accident - we are on an otherwise trusted network after all, not the internet.)
A: REJECT, next

Q: Interfaces for DHCP queries
(This host has a static IP, not one assigned via DHCP - however, even if we were using DHCP this question does not apply since iptables would recognize DHCP responses which are associated with legitimate requests.)
A: <none>, next

Q: NTP servers to query
(We should use remote Network Time Protocol servers to synchronize our local clock, however because of iptables we do not need to specifically list those NTP servers we are going to use.)
A: <none>, next

Q:ICMP types to disallow outbound
(The suggested course of action is to disallow those ICMP types used to check the route to your host from other hosts on the network.)
A: destination-unreachable time-exceeded, next

Q: Should Bastille run the firewall and enable it at boot time?
(The concern raised here is that you should not enable the firewall if you are logged in remotely if you are not absolutely sure that it will not lock you out (due to errors you may have made in the configuration). It also mentions the configuration file for the firewall (/etc/Bastille/bastille-firewall.cfg) and the location of the firewall control file (/etc/rc.d/init.d/bastille-firewall). Since we are logged in locally, and we are sure of the choices I have tested for this document, we will enable it now.)
A: yes, next

File Permissions
This section deals with disabling the "SUID root" status for various programs that would otherwise allow non-root users to use them without needing the help of an administrator. This is very important in a multi-user system, but in a case such as ours, where the only people with access to the system will be administrators, this is simply an extra layer of protection against the possibility of an attacker who has gained access to a non-privileged account on the system using an as-yet-undiscovered exploit to elevate their access to that of root. We will go through a list of commands that will be disallowed to regular users without the help of an administrator.
(On the first screen an explanation of SUID root status appears.)
A: next

Q: Would you like to disable SUID status for mount/umount?
(This will prevent users from mounting and unmounting devices such as CDROMs and floppies.)
A: yes, next

Q: Would you like to disable SUID status for ping?
(This will prevent users from using ping.)
A: yes, next

Q: Would you like to disable SUID status for at?
(This will prevent users from scheduling one-time tasks.)
A: yes, next

Q: Would you like to disable SUID status for the r-tools?
(This will disable the SUID status for the r-tools - which is a good first step, but these tools are so full of security weaknesses that we will be completely disabling them later in this script. A more detailed explanation will follow in the next section.)
A: yes, next

Q: Would you like to disable SUID status for usernetctl?
(This will prevent users from controlling the network interfaces.)
A: yes, next

Q: Would you like to disable SUID status for traceroute?
(This will prevent users from using traceroute.)
A: yes, next

Account security
Q: May we take strong steps to disallow the dangerous r-protocols?
(This question deals with the remote access services such as rsh, rlogin, and rcp - all of which use IP-based authentication to perform local actions from a remote host. The problem with IP-based authentication is that you trust the incoming request simply because of the IP address the request comes from - and since IP addresses are ridiculously easy to spoof, this is a really bad thing to rely on. Because of this, we will take various methods to ensure these are not accidentally used.)
A: yes, next

Q: Would you like to enforce password aging?
(The longest time you should keep a password without changing it is about half the time it would take to crack it with a password-cracking program (such as John the ripper). This will mean that if someone manages to get a hold of your encrypted password, they will probably not have enough time to crack your current password before you change it to something else. Because of the strong MD5 password encryption scheme in use on our server, the default implemented by Bastille is 180 days, but we can change this in /etc/login.defs if we wish to be even more cautious.)
A: yes, next

Q: Would you like to restrict the use of cron to administrative accounts?
(Since we expect that anyone who will have an account on this machine will have administrative privilege, this is simply another defense in depth measure that will disallow user accounts from scheduling recurring tasks. If we later change our mind, we can add specific users in /etc/cron.allow.)
A: yes, next

Q: Should we allow root to login on tty's 1-6?
(Since there will probably be more than one administrator on this machine, we would like to force them to log in under their own account first, and then su to become root. This will allow us to track who logged in and when, and will stop a potential attacker who has obtained (only) the root password from logging in via the ttys.)
A: no, next

Boot Security
Q: Would you like to password-protect the LILO prompt?
(This will help prevent an attacker with access to the keyboard from being able to alter the booting of the machine if rebooted - while still allowing an operator in the server

room to bring the machine up from a reboot using the normal boot process without knowing any special password.)
A: yes, next

Q: Enter LILO password, please.
(Since this password will unfortunately be stored in clear text on the local disk of the machine, do not use a password you use anywhere else since others may at some time be in a position to see this.)
A: <enter a unique (non-root) LILO password>, next

Q: Would you like to reduce the LILO delay time to zero?
(Since we may need to pass parameters to LILO at boot time at some point, we do not wish to do this.)
A: no, next

Q: Do you ever boot Linux from the hard drive?
(Since we boot from hard drive (not floppy), we need to commit our previous two changes to disk.)
A: yes, next

Q: Would you like to write the LILO changes to a boot floppy?
(Putting a password on a boot floppy which will probably only be used in a future emergency recovery situation seems like a bad idea, especially since the only security risk this introduces is the possibility of an attacker getting a copy of your recovery diskette and using it in the comfort of your server room to gain access to your system.)
A: no, next

Q: Would you like to disable CTRL-ALT-DELETE rebooting?
(The Bastille explanation for this is that an attacker with access to your keyboard might reboot your machine with this key-sequence - the reality for me however is that if you use Windows even part of the time, you are probably trained to use CTRL-ALT-DELETE frequently and reflexively. Turning this off is probably a good security precaution against you accidentally bouncing your own server while at the console.)
A: yes, next

Q: Would you like to password protect single-user mode?
(Since single-user mode grants root privilege, but does not prompt for a password by default, we will prompt the user for the root password when entering this runlevel.)
A: yes, next

Secure Inetd
Would you like to set a default-deny on TCP Wrappers and xinetd?
(These services allow you to filter incoming connection attempts based on network address and other criteria - and while your firewall is already capable of doing this, this adds another layer of defense (belt and suspenders approach) to your network accessible resources. Note that this step will take the stance that all traffic should be denied, so we

will later have to edit /etc/hosts.allow to explicitly allow traffic from hosts we wish to accept connections from.)
A: yes, next

Q: May we deactivate telnet?
(Since telnet is a clear text protocol, your username, password and all the information you type or receive during a telnet session are sent across the network unencrypted. This makes it very easy for an attacker to capture an enormous amount of sensitive information if you use this protocol - for example, if you log in as a regular user and then later su to root they will be able to capture your username and password, and the root password, as well as see every other command you type. Don't be fooled into thinking that a switched network is a reliable defense against this, since there are many ways to defeat the 'privacy' provided by switches. Secure protocols such as ssh should be used in place of telnet.)
A: yes, next

Q: May we deactivate ftp?
(Again, ftp is a clear text protocol, with all the problems associated with telnet - in addition, there is a long history of vulnerabilities with the software. As previously mentioned, good replacements such as sftp and http downloads should be used instead.)
A: yes, next

Configure Misc PAM
Q: Would you like to put limits on system resource usage?
(In this step we modify /etc/security/limits.conf to limit user accounts from hogging too many resources (for example, in a situation where a daemon running as a user is subjected to some type of denial of service attack). By limiting the core dump size, number of processes, and individual file size we mitigate the risk posed by potential future denial of service exploits by limiting our exposure to them.)
A: yes, next
(We can later modify quotas and so on in /etc/security/limits.conf.)
A: next

Q: Should we restrict console access to a small group of user accounts?
(Since only administrators will have access to this machine, explicitly listing those user accounts allowed to log in on the console is simply another defense in depth step we take.)
A: yes, next

Q: Which accounts should be able to login at console?
(We've already configured the machine to not allow root to login on any of the ttys, but we will once again exclude root (forcing administrators to log in as themselves first and then su to root) and only list the accounts of those administrators who will be allowed to login at the console.)
A: bill bob fred, next

Logging
Q: Would you like to add additional logging?
(As the saying goes: "Preventing security breaches is ideal, but detecting them is critical". The more intelligent logging you do in your environment, the better. In the next step we will add logging of kernel 'warning' and 'error' messages, and we will add 'live' logging to your virtual terminals on tty 7 and 8 (Ctrl-Alt-F7 and Ctrl-Alt-F8) - when you are using X, this will move your display :0 to Ctrl-Alt-F9, but your text consoles will still be on Ctrl-Alt F1 -> F6. These logs to your virtual terminals allow you to instantly see anything of note that has recently happened on your system.)
A: yes, next
A: next

Q: Do you have a remote logging host?
(Remote logging hosts are a good idea since savvy attackers who wish to cover their tracks can tamper with local copies of logs. Forwarding copies of the logs to a remote machine means that they will have to compromise the remote logging host as well if they wish to destroy the evidence. In our case, the ESM software we will be running actually serves as a super loghost for other machines on your network and provides the ability to consolidate, correlate, and apply security logic to data as it flows into the system in real time. Once the data is processed, it is passed along to another machine and inserted into a relational database both as an archive and for use in reporting. This being the case, we will forward the logs for our system back into itself as a loghost so that it can be processed by ESM and then be inserted in a remote database machine.)
A: yes, next

Q: What is the IP address of the machine you want to log to?
(We want to forward the logs to ESM which is listening on 514, so we simply point it at the local loopback address.)
A: 127.0.0.1, next

Q: Would you like to set up process accounting?
(Process accounting allows you to log which commands are run when and by whom. Unfortunately these logs grow large quickly, consume a lot of CPU time, and they do not include important information such as the parameters of the commands. For these reasons we will chose not to enable process accounting.)
A: no, next

Miscellaneous Daemons
(The difference between a hardened system and one that is merely patched (with the latest fixes from the vendor) is that hardened systems will be immune to many vulnerabilities before they are discovered by attackers, vendors, and before patches even exist. One facet of hardening a system is to turn off, disable, or uninstall all unnecessary services, features and applications - in this way you will be immune to any exploits which may be discovered in those services at some point in the future.)
A: next

Q: Would you like to disable apmd?
(We do not need the Advanced Power Management Daemon, since this server is not a laptop and does not have batteries – note that this differs from UPS management software such as NUT.)
A: yes, next

Q: Would you like to disable GPM?
(We do not require General Purpose Mouse support for the console mode - and differs from the regular mouse support provided under X.)
A: yes, next

Q: Would you like to deactivate the routing daemons?
(This machine will not be acting as a router.)
A: yes, next

Sendmail
Q: Do you want to leave sendmail running in daemon mode?
(We do not want sendmail running in daemon mode, since ESM will be running on port 25 instead so that it can gather data from security devices that send alerts via SMTP.)
A: no, next

Q: Would you like to run sendmail via cron to process the queue?
(We do not require sendmail to process the outgoing mail queue every 15 minutes since ESM has it's own internal SMTP delivery mechanism to send out email and beeper pages as required.)
A: no, next

Q: Would you like to disable the VRFY and EXPN sendmail commands?
(This question is irrelevant to us, since we are listening with ESM instead of sendmail, but we will turn them off as a precautionary measure.)
A: yes, next

Printing
Q: Would you like to disable printing?
(Historically, there have been a number of security problems with the standard printing mechanisms of lpr and lpd, so we will turn them off. If printing later becomes necessary, a reasonable alternative is a program like xpdq which does not use lpr or lpd.)
A: yes, next

tmpdir
Q: Would you like to install TMPDIR/TMP scripts?
(Bastille: "Many programs use the /tmp directory in ways that are dangerous on multiuser systems. Many of those programs will use an alternate directory if one is specified with the TMPDIR or TMP environment variables. We can install scripts that will be run when users log in that safely create suitable temporary directories and set the TMPDIR and TMP environment variables". An example of a vulnerability possible with an unsafe /tmp

directory is exposure of application data to unauthorized reads or writes from an attacker.)
A: yes, next

End screen
Q: Are you finished answering the questions, i.e. may we make the changes?
A: yes, next
Q: Finishing Up
A: Apply Config to System

Now that we've run Bastille, exactly what changes did we make to our system?
A full log of all the implementation details of our changes is written to an 'action-log' file – I've attached a copy of the file generated by running Bastille as described on my system in Appendix A at the end of the document. It's important that you look through the changes made to files on your system by the hardening process so that you can learn how to make the required changes yourself – after all, as you continue to update the software on your system to keep up with security patches and bug fixes you may unintentionally undo some of the hardening that Bastille did for you! Rerunning Bastille may be an option, but you'll still have to verify that it works properly against the modified system without breaking the patches you just installed. The only real answer is for you to become familiar with the changes that need to be made and monitor them yourself.
One thing that can help you to keep on top of all this is tripwire. If we run `tripwire --check` right now we will get a list of all the files that have been modified by Bastille. I will not go through this process line by line here, but if you are using this document to build a system, you should take the time to read through the action-log and the tripwire check report and verify the changes yourself. You will need this knowledge to properly maintain your system over time so that it will stay secure.

While we are at it, let's make a few minor changes that were not covered by Bastille:
Edit /etc/ssh/sshd_config and change root logins from yes to
```
PermitRootLogin no
```
Edit hosts.allow and add rules at the top to allow ssh access by those machines you wish to have it – remember this file is 'first match and exit' so you need to be careful about order. I added
```
sshd : 192.168.0.0/255.255.255.0
```
Add whatever is appropriate for you - in my case this would allow hosts on the local LAN connect via ssh, but not hosts connecting over the VPN (or anywhere else such as the internet or the DMZ)
`killall -HUP sshd` will make these last couple changes take effect immediately.
We don't require sendmail for anything, since ESM will be replacing it from a network perspective and there will be no local user accounts that maintain mailboxes on this machine. Let's turn it right off.
```
chkconfig sendmail off
/etc/rc.d/init.d/sendmail stop
```

Patching the OS

Congratulations – now that you've finished a clean install of the OS and run the latest hardening scripts, you're officially out of date. As of the time that this document was written there were already a list of over a dozen patches available for the packages installed, including a number of security vulnerabilities, and a new release of the operating system. The good news is that it's easy to catch up, and stay up to date.

Before we begin, I find it useful to take a new snapshot of my system with a `tripwire --init` before making a number of major changes, just so I can keep track of what takes place.

At the prompt:
`:rhn_register`
This program will prompt you to enter some information about yourself and your system - be aware that this *may* expose some of your system info such as your IP and the packages you have installed, however Redhat has a fairly enlightened privacy policy, which is well detailed during the registration process. If this doesn't sit well with you, or your organizational security policies prohibit such actions, you can use a third party update manager such as the fantastic AutoUpdate perl script available at http://www.mat.univie.ac.at/~gerald/ftp/autoupdate/index.html . This script allows you to download patches for all your systems to a central location on your network and then update each machine as required from there.
For this document however, we will take the easy path and assume that you have "entitled" your machine by registering it with RedHat.

Now that our machine is entitled we can get updates as follows:
`:up2date`
This will check the versions of packages you have installed against their database to see where you are out of date and what the 'advisories' are that you should be aware of (if a recommended update is a bug fix or a vulnerability fix or just an enhancement).
You will be prompted to accept Redhat's key into your keyring so you can validate the digital signatures on all the packages you get from them. The program will prompt you to do this as follows:
`/usr/bin/gpg --import /usr/share/rhn/RPM-GPG-KEY`

Among other things, up2date will prompt you to update your kernel – this sounds scary, but it's actually painless and unless you are locked into a 'stock' kernel because of proprietary hardware drivers or something, it's a good idea to stay up to date on this as well. You will be prompted to reboot once the kernel update is complete for the change to take effect.

Now if we took a tripwire snapshot before we began the update, we can check the state of our system with a `tripwire --check` and easily determine which critical files were affected. We can quickly reference this against our Bastille action-log, and see if any of our hardening steps were undone, then fix them.

Another thing we may wish to do at this time is configure our preferences in X. You will occasionally need to use the ESM visual rules studio under X, but we recommend continuing to run the system in runlevel 3 (as per your installed default in /etc/inittab) and only using the GUI portion of the operating system with `startx` when required. You may wish to once again run a `tripwire --check` at this time to see what files X modifies each time it starts.

Once we've made any necessary fixes, and we are satisfied with the state of the system, we can once again start with a clean slate by doing another `tripwire --init`.

Installing ESM in a chroot jail

The chief goal of this document is to build a suitable server for a Java-based Enterprise Security Management application. Among other things it will be listening on ports 25 (SMTP), 162 (SNMP traps), 514 (syslog) and 13337 (inter-ESM communication) to consolidate and correlate the information from your security devices and critical servers. For this reason, it is vital that the server be as secure as possible, so to increase the layers of difficulty that an intruder must go through to compromise the box, we will run ESM in a chroot jail.

The name 'chroot' comes from "change root" – as in the root of the file system, not the "administrative" account - the function of the program is to change the directory root of the application in question to a specially prepared 'jail' inside the file hierarchy which contains copies of only that portion of the system that the application requires to function. This would therefore limit an attacker who was able to exploit the application to a very limited set of tools from which to compromise the host operating system (in our case, preventing them from going on to compromise the security data warehouse).

While Java has historically been a relatively secure application platform with comparatively few vulnerabilities, it makes sense to apply this extra measure in accordance with the principle of defense in depth.

Install the JVM

```
/mnt/cdrom/j2sdk-1_3_1_01-linux-i386-rpm.bin
<agree to the license with "yes">
rpm -ivv jdk-1.3.1_01.i386.rpm
```

Create the ESM User

```
groupadd -g 150 esm
useradd -u 150 -g 150 -M esm
```

Create the chroot jail

Setup the file structure

```
mkdir -m 700 /home/esm
cd /home/esm
mkdir -p bin dev lib/i686 usr/java/jdk1.3.1_01 usr/bin usr/lib
mknod -m 666 dev/null c 1 3
```

copy in the JVM

```
cp -R /usr/java/jdk1.3.1_01/* usr/java/jdk1.3.1_01
```

copy in the required libraries

```
cp /lib/i686/libpthread.so.0 lib/i686
cp /lib/i686/libc.so.6 lib/i686
cp /lib/libdl.so.2 lib
cp /lib/ld-linux.so.2 lib
cp /lib/libtermcap.so.2 lib
cp /lib/libnsl.so.1 lib
cp /lib/libm.so.6 lib
cp /usr/lib/libstdc++-libc6.1-1.so.2 usr/lib
```

Copy in the required system binaries

```
cp /bin/bash bin
cp /bin/basename bin
cp /bin/uname bin
cp /bin/ls bin
cp /usr/bin/expr usr/bin
cp /usr/bin/dirname usr/bin
cp /bin/grep bin
cp /usr/bin/head usr/bin
cp /bin/cut bin
```

create symbolic links for java and sh (as you would have in the host OS)

```
ln -s /usr/java/jdk1.3.1_01/bin/java bin/java
ln -s bash bin/sh
```

Crafting your own chroot

So, how did I come up with this rather odd set of libraries, binaries, and devices to copy into my chroot structure you ask? Well, unfortunately the answer is not pretty. The first place to start is by running ldd against the programs you want to put in the chroot jail to find out what libraries they require:

```
[root@esm-cs lib]# ldd /bin/bash
        libtermcap.so.2 => /lib/libtermcap.so.2 (0x40020000)
        libdl.so.2 => /lib/libdl.so.2 (0x40024000)
        libc.so.6 => /lib/i686/libc.so.6 (0x40028000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

This works well for binaries such as bash, but not so for the chameleon-like java which has a lot of wrappers to handle environmental issues. Obviously binaries with statically linked libraries are the easiest to handle – and you can quickly solve many issues by including busybox (which is a single binary which has most of the functionality of many common UNIX utilities – in fact it claims to be a fairly complete POSIX environment). The downside of busybox is that you might not **want** all that functionality in your chroot jail, since you are trying to limit the scope of what an attacker could do if they were able to use a buffer overflow or similar exploit to gain a shell prompt inside your chroot environment.

I copied in everything that ldd suggested, copied in java, chrooted into the environment and continued to try to instantiate the jvm:

```
[root@esm-cs lib]# chroot /home/esm
bash-2.04# java -version
/usr/java/jdk1.3.1_01/bin/i386/native_threads/java: error while
loading shared libraries: libm.so.6: cannot load shared object
file: No such file or directory
```

Each failure would produce an error message and I would copy in the appropriate library. If your program is still failing and you can see why, try running truss or trace and see what files or devices it is trying to open. You can create new devices in your chroot /dev with mknod.

Test the JVM in the chroot environment

```
chroot /home/esm
java -version
```

Expected Result:

java version "1.3.1_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1_01)
Java HotSpot(TM) Client VM (build 1.3.1_01, mixed mode)

```
exit
```

Install ESM

Install the ESM Central Server application from the product CD as per the user documentation, except install to /esm instead of /opt/esm. Install optional security device normalizers from the enhancement pack as required.

The assumption is that you will house the ESM data tables on your relational database on another machine in your network (probably with your database farm, or whatever might be dictated by corporate policy). Configure the ESM Central Server to log to your database. You should also arrange to install the web-based reporting engine for the database on a remote machine – either the database machine itself or another box.

Copy the application into the chroot directory

```
cd /home/esm
mkdir esm
cp -R /esm/* esm
```

Edit /etc/rc.d/init.d/esm script to start in /home/esm under the 'esm' userid/group you created.

Test your change:

```
/etc/rc.d/init.d/esm stop
/etc/rc.d/init.d/esm start
```

check the /home/esm/esm/audit/db.log to see if ESM is running in the jail. (You should see the startup event logged to this file.)

From a remote machine, test to see if messages are passing through your firewall as expected and reaching ESM by generating a bogus security message on the ESM listener port (13337) using netcat:

```
/usr/bin/nc esm-cs 13337
event
  type esm.test
endevent
^c
```

There are many ways to generate traffic to test our SNMP, SMTP, and syslog listeners. A few simple possibilities are listed below, but they are only presented as examples, not necessarily the best or only way to go.
You can 'capture' real events to file using netcat and play them back as required. For example:
Capture an incoming SNMP trap:
`nc –l –u –p 162 > snmptrap.bin`
'Forward' the trap to ESM:
`nc –u esm-cs 162 < snmptrap.bin`

For SMTP use:
`nc esm-cs 25 < smtpmessage.txt`

For syslog use the technique above, or add esm-cs as your remote loghost and use logger to generate an event.

These test events should pass through your system (based on the default event handling rules) and be logged to your remote database machine over the network using the native JDBC driver. Validate that the events have been successfully logged in the remote database by using the web-based reporting engine installed with the database portion of ESM.

You have now setup and configured a hardened EMS Central Server – but before designing your event handling rules, pointing your external security devices and logs at ESM, and so on, there are a few final steps you should take.

Final Steps

Now that all hardening is done, test your functionality. Have you secured yourself out of anything you needed?

- Generate a new Tripwire --init database before putting the machine on the production network – and copy the database to write-once media or a different machine on the network.

- Configure your NetworkTimeProtocol daemon according to your network policies – it is important to make sure your server's clock stays in sync with all the security devices in your network that will be reporting to it.

- Setup appropriate banners in issue, motd and as a popup on your X11 login screen (a good writeup on how to do this can be found at http://www.sans.org/y2k/practical/Paul_Sery_GCUX.rtf)

- Scan it from the network – here is the sample output from an nmap scan:

```
[root@otherhost root]# nmap -v -sS -sU -P0 -O 192.168.0.2

Starting nmap V. 2.54BETA22 ( www.insecure.org/nmap/ )
Host  (192.168.0.2) appears to be up ... good.
Initiating SYN Stealth Scan against  (192.168.0.2)
Adding TCP port 22 (state open).
Adding TCP port 25 (state open).
The SYN Stealth Scan took 670 seconds to scan 1563 ports.
Initiating UDP Scan against  (192.168.0.2)
The UDP Scan took 97 seconds to scan 1563 ports.
(no udp responses received -- assuming all ports filtered)
Warning:  OS detection will be MUCH less reliable because we did
not find at least 1 open and 1 closed TCP port
For OSScan assuming that port 22 is open and port 44261 is closed
and neither are firewalled
For OSScan assuming that port 22 is open and port 33059 is closed
and neither are firewalled
For OSScan assuming that port 22 is open and port 41886 is closed
and neither are firewalled
Interesting ports on  (192.168.0.2):
(The 3124 ports scanned but not shown below are in state:
filtered)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp

No exact OS matches for host (test conditions non-ideal).
TCP/IP fingerprint:
SInfo(V=2.54BETA22%P=i386-redhat-linux-
gnu%D=11/19%Time=3BF9B204%O=22%C=-1)
TSeq(Class=RI%gcd=1%SI=2A3D34%IPID=C%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=2A3CFB%IPID=C%TS=100HZ)
TSeq(Class=RI%gcd=1%SI=2A3E34%IPID=C%TS=100HZ)
T1(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T2(Resp=N)
T3(Resp=Y%DF=Y%W=16A0%ACK=S++%Flags=AS%Ops=MNNTNW)
T4(Resp=Y%DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

Uptime 0.027 days (since Sat Nov 17 19:51:14 2001)
TCP Sequence Prediction: Class=random positive increments
                         Difficulty=2768436 (Good luck!)
IPID Sequence Generation: Duplicated ipid (!)
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 782
seconds
```

- Check to see what ports you are offering up to the rest of the network. Here is the sample output from a `netstat -tuna`

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:13337           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:514             0.0.0.0:*
udp        0      0 0.0.0.0:162             0.0.0.0:*
```

Take a full backup of the system

- Schedule incremental and full backups to stay up to date with system changes
- Start a changelog NOW
- There's no better time than the present - try doing a restore from your backup

Keep informed about the changing face of security

- Once your system is "entitled" on the Redhat network their bot will email you when you need to update and you can schedule your system to auto update if you are comfortable with that
- Subscribe to the SANS news bulletins
- Read information security publications both in print and on the web.

Ensure that anyone with an account on the system understands

- your security policies and procedures
- how to report or handle any security breaches/ odd things they become aware of
- what their responsibilities are in ensuring the ongoing security of the system

How do we rank against the SANS top twenty vulnerabilities?

(http://66.129.1.101/top20.htm or browse from SANS.org)

G1 - Default installs of operating systems and applications
We've done a careful custom install and followed up with the latest patches.
G2 - Accounts with No Passwords or Weak Passwords
We have implemented pass phrases and stated password policies. A PAM to exclude poor passwords would help to reinforce this.
G3 - Non-existent or Incomplete Backups
We have scheduled full and incremental backups.
G4 - Large number of open ports
We offer only those 5 ports that are essential to the operation of the server.
G5 – Not filtering packets for correct incoming and outgoing addresses
This machine does not route traffic, however we still filter source routed packets and so on.
G6 - Non-existent or incomplete logging

We have enhanced the default logging functions, and among its other functions, the ESM server is a sophisticated loghost which warehouses logs on a separate machine.

G7 - Vulnerable CGI Programs

We offer up no CGI

Windows vulnerabilities

None are applicable.

U1 - Buffer Overflows in RPC Services

We do not have any open RPC ports.

U2 - Sendmail Vulnerabilities

We have disabled sendmail.

U3 - Bind Weaknesses

We have disabled named.

U4 - R Commands

We have removed the R-commands.

U5 - LPD (remote print protocol daemon)

We have disabled the print daemon.

U6 – sadmind and mountd

The Solstice sadmind doesn't apply to Linux and we aren't mounting or offering NFS drives.

U7 - Default SNMP Strings

The only SNMP (traps) we are using is being handled by the ESM application, which can apply filters or generate alerts based on the SNMP community strings it receives.

Also, you can now download an assessment tool from http://www.cisecurity.org which allows you to scan for the SANS top twenty vulnerabilities from a remote UNIX host. This tool is actually just an updated version of the SARA vulnerability assessment software.

Appendix A – Bastille action-log

Attached is a copy of the action-log file created by Bastille as it made the requested changes in the system according to the choices detailed in this document.

```
placed file /bastille-firewall  as  /etc/rc.d/init.d/bastille-
firewall
# change permissions on /etc/rc.d/init.d/bastille-firewall from
644 to   500
chmod   500,"/etc/rc.d/init.d/bastille-firewall";
placed file /bastille-ipchains  as  /sbin/bastille-ipchains
# change permissions on /sbin/bastille-ipchains from 644 to   500
chmod   500,"/sbin/bastille-ipchains";
placed file /bastille-netfilter  as  /sbin/bastille-netfilter
# change permissions on /sbin/bastille-netfilter from 644 to
500
chmod   500,"/sbin/bastille-netfilter";
placed file /bastille-firewall.cfg  as  /etc/Bastille/bastille-
firewall.cfg
# change permissions on /etc/Bastille/bastille-firewall.cfg from
644 to   600
chmod   600,"/etc/Bastille/bastille-firewall.cfg";
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
TRUSTED_IFACES="lo"                          # MINIMAL/SAFEST
with:
TRUSTED_IFACES="lo"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
PUBLIC_IFACES="eth+ ppp+ slip+"              # SAFEST
with:
PUBLIC_IFACES="eth+ ppp+ slip+"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
INTERNAL_IFACES=""                       # SAFEST
with:
INTERNAL_IFACES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
TCP_AUDIT_SERVICES="telnet ftp imap pop3 finger sunrpc exec login
linuxconf ssh"
with:
TCP_AUDIT_SERVICES="telnet ftp imap pop-3 finger sunrpc exec
login linuxconf ssh"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
TCP_PUBLIC_SERVICES=""                               # MINIMAL/SAFEST
with:
TCP_PUBLIC_SERVICES="22 25 9317"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
```

```
UDP_PUBLIC_SERVICES=""                          # MINIMAL/SAFEST
with:
UDP_PUBLIC_SERVICES="162 514"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
TCP_INTERNAL_SERVICES=""                    # MINIMAL/SAFEST
with:
TCP_INTERNAL_SERVICES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
UDP_INTERNAL_SERVICES=""                    # MINIMAL/SAFEST
with:
UDP_INTERNAL_SERVICES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
FORCE_PASV_FTP="Y"                          # SAFEST
with:
FORCE_PASV_FTP="N"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
TCP_BLOCKED_SERVICES="6000:6020"
with:
TCP_BLOCKED_SERVICES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
UDP_BLOCKED_SERVICES="2049"
with:
UDP_BLOCKED_SERVICES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
ENABLE_SRC_ADDR_VERIFY="Y"                  # SAFEST
with:
ENABLE_SRC_ADDR_VERIFY="Y"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
IP_MASQ_NETWORK=""                          # DISABLE/SAFEST
with:
IP_MASQ_NETWORK=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
IP_MASQ_MODULES="ftp raudio vdolive"            # RECOMMENDED
with:
IP_MASQ_MODULES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
REJECT_METHOD="DENY"
with:
REJECT_METHOD="REJECT"
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
DHCP_IFACES=""                  # DISABLED
with:
```

```
DHCP_IFACES=""
# File modification in /etc/Bastille/bastille-firewall.cfg --
replaced line
NTP_SERVERS=""                  # DISABLE NTP QUERIES / SAFEST
with:
NTP_SERVERS=""
placed file /bastille-firewall-schedule  as  /sbin/bastille-
firewall-schedule
# change permissions on /sbin/bastille-firewall-schedule from 644
to   500
chmod   500,"/sbin/bastille-firewall-schedule";
placed file /bastille-firewall-reset  as  /sbin/bastille-
firewall-reset
# change permissions on /sbin/bastille-firewall-reset from 644 to
500
chmod   500,"/sbin/bastille-firewall-reset";
placed file /ifup-local  as  /sbin/ifup-local
# change permissions on /sbin/ifup-local from 644 to   500
chmod   500,"/sbin/ifup-local";
# Firewall.pm: invoking firewall
# Firewall.pm: enabling firewall with B_chkconfig_on
# chkconfig_on enabling bastille-firewall
# chkconfig_on will use runlevels 2,3,4,5 for "bastille-firewall"
with S order 05 and K order 98
# created a symbolic link called /etc/rc.d/rc0.d/K98bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc0.d/K98bastille-firewall";
# Created link /etc/rc.d/rc0.d/K98bastille-firewall
# created a symbolic link called /etc/rc.d/rc1.d/K98bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc1.d/K98bastille-firewall";
# Created link /etc/rc.d/rc1.d/K98bastille-firewall
# created a symbolic link called /etc/rc.d/rc2.d/S05bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc2.d/S05bastille-firewall";
# Created link /etc/rc.d/rc2.d/S05bastille-firewall
# created a symbolic link called /etc/rc.d/rc3.d/S05bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc3.d/S05bastille-firewall";
# Created link /etc/rc.d/rc3.d/S05bastille-firewall
# created a symbolic link called /etc/rc.d/rc4.d/S05bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc4.d/S05bastille-firewall";
# Created link /etc/rc.d/rc4.d/S05bastille-firewall
# created a symbolic link called /etc/rc.d/rc5.d/S05bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
```

```
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc5.d/S05bastille-firewall";
# Created link /etc/rc.d/rc5.d/S05bastille-firewall
# created a symbolic link called /etc/rc.d/rc6.d/K98bastille-
firewall from /etc/rc.d/init.d/bastille-firewall
symlink "/etc/rc.d/init.d/bastille-
firewall","/etc/rc.d/rc6.d/K98bastille-firewall";
# Created link /etc/rc.d/rc6.d/K98bastille-firewall
# sub GeneralPerms
# sub SUIDAudit
# change permissions on /bin/mount from 755 to   700
chmod   700,"/bin/mount";
# change permissions on /bin/umount from 755 to   700
chmod   700,"/bin/umount";
# change permissions on /bin/ping from 755 to   700
chmod   700,"/bin/ping";
# change permissions on /usr/bin/at from 755 to   700
chmod   700,"/usr/bin/at";
# change permissions on /usr/bin/rcp from 755 to     0
chmod     0,"/usr/bin/rcp";
# change permissions on /usr/bin/rlogin from 755 to     0
chmod     0,"/usr/bin/rlogin";
# change permissions on /usr/bin/rsh from 755 to     0
chmod     0,"/usr/bin/rsh";
# change permissions on /usr/sbin/usernetctl from 755 to   700
chmod   700,"/usr/sbin/usernetctl";
# change permissions on /usr/sbin/traceroute from 755 to   700
chmod   700,"/usr/sbin/traceroute";
# sub ProtectRhosts
# sub Password Aging
adding PASS_MAX_DAYS setting to /etc/login.defs
# File modification in /etc/login.defs -- replaced line
PASS_MAX_DAYS   99999
with:
PASS_MAX_DAYS   180
# sub RestrictCron
# Created file /etc/cron.allow
Appended the following line to /etc/cron.allow:
root
# sub RootTTYLogins
# File modification in /etc/securetty -- replaced line
tty1
with:
# File modification in /etc/securetty -- replaced line
tty10
with:
# File modification in /etc/securetty -- replaced line
tty11
with:
# File modification in /etc/securetty -- replaced line
tty2
with:
```

```
# File modification in /etc/securetty -- replaced line
tty3
with:
# File modification in /etc/securetty -- replaced line
tty4
with:
# File modification in /etc/securetty -- replaced line
tty5
with:
# File modification in /etc/securetty -- replaced line
tty6
with:
# Created file /etc/bastille-no-login
Appended the following line to /etc/bastille-no-login:
root
Prepended the following line to /etc/pam.d/xdm:
auth  required   /lib/security/pam_listfile.so onerr=succeed
item=user sense=deny file=/etc/bastille-no-loginPrepended the
following line to /etc/pam.d/gdm:
auth  required   /lib/security/pam_listfile.so onerr=succeed
item=user sense=deny file=/etc/bastille-no-loginPrepended the
following line to /etc/pam.d/kde:
auth  required   /lib/security/pam_listfile.so onerr=succeed
item=user sense=deny file=/etc/bastille-no-login# sub ProtectLILO
Prepended the following line to /etc/lilo.conf:
restricted
password=password
# change permissions on /etc/lilo.conf from 644 to   600
chmod   600,"/etc/lilo.conf";
# change ownership on /etc/lilo.conf from 0 to 0
chown 0,0,"/etc/lilo.conf";
# Re-running lilo# sub SecureInittab
# File modification in /etc/inittab -- hash commented line
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
like this:
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
Inserted the following line in /etc/inittab:

~~:S:wait:/sbin/sulogin
# sub SetWrappersDefaultDeny
Appended the following line to /etc/hosts.allow:
# Bastille: default deny
# no safe_finger for in.fingerd (prevent loops)
in.fingerd : ALL : DENY
# but everything else is denied & reported with safe_finger
ALL : ALL : spawn (/usr/sbin/safe_finger -l @%h | /bin/mail -s
"Port Denial noted %d-%h" root) & : DENY
# sub ModifyLimitsconf
Appended the following line to /etc/security/limits.conf:
# prevent core dumps
*     hard  core  0
```

```
Appended the following line to /etc/security/limits.conf:
# limit size of any one of users' files to 100mb
*     hard  fsize 100000

# sub LimitConsoleLogins
Appended the following line to /etc/security/access.conf:
-:ALL EXCEPT bill:LOCAL
Appended the following line to /etc/pam.d/login:
account    required     /lib/security/pam_access.so
Appended the following line to /etc/pam.d/xdm:
account    required     /lib/security/pam_access.so
Appended the following line to /etc/pam.d/gdm:
account    required     /lib/security/pam_access.so
Appended the following line to /etc/pam.d/kde:
account    required     /lib/security/pam_access.so
# sub ConfigureAdditionalLogging
Appended the following line to /etc/syslog.conf:
############ BASTILLE ADDITIONS BELOW : #################
Appended the following line to /etc/syslog.conf:
# Log warning and errors to the new file /var/log/syslog
*.warn;*.err    /var/log/syslog

Appended the following line to /etc/syslog.conf:
# Log all kernel messages to the new file /var/log/kernel
kern.*     /var/log/kernel

Appended the following line to /etc/syslog.conf:
# Log all logins to /var/log/loginlog
auth.*;user.*;daemon.none   /var/log/loginlog

Appended the following line to /etc/syslog.conf:
# Log additional data to the Alt-F7 and Alt-F8 screens (Pseudo
TTY 7 and 8)

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

Appended the following line to /etc/syslog.conf:
*.*   /dev/tty12
Appended the following line to /etc/syslog.conf:
*.warn;*.err    @127.0.0.1
authpriv.*;auth.*     @127.0.0.1
Appended the following line to /etc/syslog.conf:
########## BASTILLE ADDITIONS CONCLUDED : ###############
# Created file /var/log/syslog
# Created file /var/log/kernel
# Created file /var/log/loginlog
Appended the following line to /etc/logrotate.d/syslog:
```

```
/var/log/kernel {
    postrotate
     /usr/bin/killall -HUP syslogd
    endscript
}

/var/log/syslog {
    postrotate
     /usr/bin/killall -HUP syslogd
    endscript
}

/var/log/loginlog {
    postrotate
     /usr/bin/killall -HUP syslogd
    endscript
}
# sub AddProcessAccounting
# sub AddSecurityChecks
# sub DeactivateAPMD
# Removed link /etc/rc.d/rc2.d/S26apmd
# Removed link /etc/rc.d/rc3.d/S26apmd
# Removed link /etc/rc.d/rc4.d/S26apmd
# Removed link /etc/rc.d/rc5.d/S26apmd
# sub DeactivateGPM
# Removed link /etc/rc.d/rc2.d/S85gpm
# Removed link /etc/rc.d/rc3.d/S85gpm
# Removed link /etc/rc.d/rc4.d/S85gpm
# Removed link /etc/rc.d/rc5.d/S85gpm
# sub DeactivateRoutingDaemons
Didn't chkconfig_off gated because we couldn't open
/etc/rc.d/init.d/gated
Didn't chkconfig_off routed because we couldn't open
/etc/rc.d/init.d/routed
# sub ResistUsernameRecon
# File modification in /etc/sendmail.cf -- replaced line
O PrivacyOptions=authwarnings,novrfy,noexpn,restrictqrun
with:
O PrivacyOptions=goaway
# sub RestrictRelaying
# sub DisableLprdrm
# Removed link /etc/rc.d/rc2.d/S60lpd
# Removed link /etc/rc.d/rc3.d/S60lpd
# Removed link /etc/rc.d/rc4.d/S60lpd
# Removed link /etc/rc.d/rc5.d/S60lpd
# change permissions on /usr/bin/lpr from 755 to   500
chmod   500,"/usr/bin/lpr";
# change permissions on /usr/bin/lprm from 755 to   500
chmod   500,"/usr/bin/lprm";
placed file /bastille-tmpdir.sh  as  /etc/profile.d/bastille-
tmpdir.sh
```

```
# change permissions on /etc/profile.d/bastille-tmpdir.sh from
644 to   755
chmod   755,"/etc/profile.d/bastille-tmpdir.sh";
placed file /bastille-tmpdir.csh  as  /etc/profile.d/bastille-
tmpdir.csh
# change permissions on /etc/profile.d/bastille-tmpdir.csh from
644 to   755
chmod   755,"/etc/profile.d/bastille-tmpdir.csh";
placed file /bastille-tmpdir-defense.sh  as  /etc/bastille-
tmpdir-defense.sh
# change permissions on /etc/bastille-tmpdir-defense.sh from 644
to   755
chmod   755,"/etc/bastille-tmpdir-defense.sh";
```

References

Bastille Linux
http://www.bastille-linux.org/ , Oct 17, 2001

Securing Linux Step-by-Step, Version 1.0
The SANS Institute, Guided and Edited by Lee E. Brotzman and David A. Ranch

Redhat Corporate website and **Redhat network**
http://www.redhat.com and http://rhn.redhat.com, Oct 17, 2001

Linux Documentation Project – particularly categories 4.2.6 and 4.4.8
http://www.linuxdoc.org/, Oct 17, 2001

SANS Website
http://www.sans.org, Oct 17, 2001

Internet Engineering Task Force website – particularly Security Area working group
http://www.ietf.org/, Oct 17, 2001

Linux Journal, particularly "Paranoid Penguin" column and September 2001 issue on Linux Security - Published by Specialized Systems Consultants, Inc.

Incidents.org
http://incidents.org/, Oct 17, 2001