

UNCLASSIFIED

Guide to Securing Linux

Prepared by the
Space and Naval Warfare Systems Center, San Diego
for the
Technical Support Working Group

Authors:
Carsten P. Gehrke
Charles N. Long



Revision:
27 Sep 2004

Distribution Statement: This document is available for general release to all interested parties. The software associated with the Fort Knox for Linux (FKL) program along with this documentation is licensed under the terms of the GNU General Public License (GPL) and as such is considered open source. The software and this documentation is free; you can redistributed it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. You should have received a copy of the GNU General Public License along with the FKL program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

UNCLASSIFIED

UNCLASSIFIED

This page intentionally left blank.

UNCLASSIFIED

Revision Information

Date	Comment	Author
2004-09-27	First public release.	CPG

Disclaimer

SOFTWARE AND DOCUMENTATION IS PROVIDED “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE AND DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademark Information

Linux® is a registered trademark of Linus Torvalds.

“Red Hat” is a registered trademark of Red Hat, Inc.

“SuSE” is a registered trademark of SuSE AG, a Novell, Inc. business.

All other trademarks are property of their respective owners.

Abstract

As the popularity of the Linux operating system increases, so does its use in sensitive or critical environments. While most modern distributions of Linux are more secure than previous versions, there are still some additional procedures that can improve these systems. This document describes an installation and configuration procedure which will enhance the security of the Linux operating system, specifically the Red Hat Enterprise Server and Workstation, as well as the SuSE Enterprise Server distributions. Most of the recommendations are general enough that they may also be applied to other Linux systems by an experienced system administrator.

Many books, documents, and Web sites authored by security professionals and organizations have been studied and analyzed to determine industry best practices. Some of the resources dealt with general computer security, while others addressed Linux and/or the above distributions in particular. Furthermore, Department of Defense directives were consulted to determine the Department's specific requirements.

The described system will have minimal functionality, which is in accordance with industry accepted standards for secure operation. In some cases, these guidelines may reduce the ease-of-use of the computer system. This has been done deliberately when the result is a system that is more difficult to compromise. Department of Defense directives have been followed where applicable to ensure that the system meets the most stringent requirements.

Being of a technical nature, this document is directed at system administrators and sophisticated users of computers running the Linux operating system. A degree of familiarity with Linux or a UNIX system is assumed. Knowledge of computer security is not required.

Typographic Conventions

Typeface	Meaning
command	Command line input, the names of files, the contents of text files.
<i>variable</i>	A variable in a command line, to be replaced with an actual value such a real file name.

Table of Contents

Revision Information	i
Disclaimer	ii
Trademark Information	iii
Abstract	iv
Typographic Conventions	v
1 Introduction	1
1.1 Background	1
1.1 .1 Threats	1
1.1 .2 Risk	2
1.1 .3 Controls	3
1.1 .4 Security Policy	3
1.2 How to Use this Document	3
1.2 .1 Audience	3
1.2 .2 Verification of Integrity	4
2 Preparation	5
2.1 Hardware Clock	5
2.2 Partitions	5
2.3 Installation of Linux Distribution	6
2.3 .1 Packages	7
2.4 Apply Patches	12
3 Configuration	13
3.1 System Initialization	13
3.1 .1 BIOS	13
3.1 .2 Boot Loader	13
3.1 .3 Init	14
3.2 Accounts	15
3.2 .1 Superuser Access	15
3.3 Account Passwords	16
3.3 .1 DoD Requirements	16
3.3 .2 Enforcing Strong Passwords	16
3.4 Login Banner	17
3.5 Resource Limits	17
3.6 Auditing	17
3.6 .1 DoD requirements	18
3.7 Filesystem	18
3.7 .1 Set Mount Point Options	18
3.7 .2 Administrative Commands	19
3.7 .3 setuid/setgid Programs	19
3.7 .4 Other File and Directory Permissions	20
3.8 Services	20
3.8 .1 ntpd	20
3.8 .2 sshd	20
3.9 Document the Settings	21

3.10 Restart the System	21
4 Maintenance	23
4.1 Accounts	23
4.2 Filesystem	23
4.2 .1 setuid/setgid Programs	23
4.3 Apply Patches	23
4.4 Backups	24
Appendix A	25
Detailed Installation Instructions	25
Appendix B	29
Sample Login Banners	29
Appendix C	31
License Information	31
References	39

List of Tables

Table 2-1: File system sizes.	6
Table 3-1: Historical users and groups.	13

1 Introduction

As the Linux operating system gains popularity, its use in sensitive or critical environments also increases. The deployment of Linux in these areas requires special attention to the security of the systems, and while most modern distributions are more secure than earlier versions, there are additional procedures and settings that can improve them. The goal of this document is to provide the system administrator or end user with specific guidelines to achieving that improvement. These guidelines were developed from the study of numerous documents, books, and other works on computer security. Where applicable, requirements of the U.S. Department of Defense (DoD) have been observed and implemented.

1.1 Background

Any but the most simple computer program will have a number of undetected errors. Some of these errors do not affect the operation of the system unless an extraordinary combination of factors is presented. In some cases, such a combination may lead to a system compromise, that is a situation where the reliability or integrity is damaged. Throughout the world, there are a number of individuals who are intent on finding the exact combination of factors necessary to effect such a compromise.

Probably the most well known exploit of a programming error is the buffer-overflow attack. In this situation, the attacker provides a program with specially crafted input designed to fill a buffer beyond its bounds. In doing so, the person trying to compromise the system can overwrite locations normally reserved for the control of program execution, substituting values which will coerce the target computer to perform operations of the attacker's choosing. In most cases, the goal of that individual will be to spawn a shell with the privileges of the owner of the original program. For many service daemons, this may be a user with special privileges or even the superuser.

While transport-layer firewalls may offer some protection against malicious intent from a remote user by blocking traffic on unused ports, the greatest threat to a system stems from the exploitation of programming errors in required services listening on well-known ports that the firewall does not block. Nevertheless, a firewall is still useful because it can block traffic on unauthorized ports, which might be used by an intruder for collecting additional resources to further compromise the system, or to launch attacks against other hosts. In addition, the logging facilities of a firewall can be an aid in the forensic analysis of intrusions.

Since firewalls cannot be relied upon to keep unauthorized individuals from accessing a system's resources, additional measures must be applied. In some cases, these measures may mitigate an entire class of attack, in others they may simply discourage all but the most determined attacker.

1.1 .1 Threats

Threats are circumstances that can lead to undesirable results, such as loss of data or denial of service. Computer systems can be threatened physically or logically. The sources of threats can be internal or external to an organization.

Physical threats are perhaps the best understood, since they can affect any type of asset. For instance, like any other item that has a high value, a computer may be stolen. What must be

considered in addition to the loss of the actual hardware is the loss or exposure of the information that may have been stored on it. The pure loss of information can be mitigated by regular back-up of the data onto removable media or onto a different system. The exposure of the information can be prevented by the use of encryption. Other types of physical threats are acts of nature, intentional acts of violence, and hardware failure. Again, traditional methods of protection can be applied in these cases.

Logical threats are particular to computer systems, and more difficult to understand because of their short history. Examples of this kind of threat are the aforementioned buffer overflow as well as Trojan horse programs, back doors, and denial of service. Finally, user errors may also be classified as a logical threat.

In addition to the above examples, social engineering attacks may also be considered a logical threat. In the past, the simplest form of social engineering was peering over an authorized user's shoulder while that person entered authentication information; the older programs would often show the password in cleartext as it was entered. Modern programs generally do not do this anymore; they now either echo each keystroke with the same character (e.g. an asterisk), or show no output at all. Following someone's rapid movements on the keyboard is much more difficult than reading the output on the screen. A more effective method, which also allows the attacker to remain remote and anonymous, is that of manipulation. An example would be a call to a legitimate user from someone claiming to be the system administrator or network technician. The caller may persuade the user to divulge authentication information such as the password by suggesting that there is a problem with the user's account or connection, and that the caller can fix this quickly by logging in as that user. Recent examples of this kind of attack are e-mail messages that claim to come from a legitimate source, but in fact are from a third party ("phishing"). The victims are asked to enter confidential information on a Web site that looks authentic, but is in fact set up by the attacker to collect this information for later abuse.

1.1 .2 Risk

Risk is the possibility of a successful attack. There are several classes of risk: 1) Loss of confidentiality; 2) Loss of integrity; and 3) Loss of availability. Risk analysis is the process of determining the cost of the risk to the organization. Obviously, the higher the cost, the more effort an organization would expend to avoid a particular risk.

An example of loss of confidentiality would be if an attacker was able to compromise the database servers of a company conducting sales over its Web site, and managed to obtain a list of the customer's credit card information. The cost of this risk would be difficult to ascertain, since it might comprise loss of sales due to lack of customer confidence, or even the cost of litigation by customers or credit card companies.

Loss of integrity concerns the validity of stored data. Instead of copying data as in the previous example, an attacker may choose to modify it by changing, adding, or deleting fields, records, or even entire files.

Finally, loss of availability refers to the authorized users' access to the resources. In this case, the attacker may disable the system entirely, or cause it to process so much maliciously provided data that normal operations cannot occur. The denial-of-service attacks against certain Web sites are good examples of this risk. Here the cost might be the loss of sales due to customers not being able to use the site to place orders.

1.1 .3 Controls

Controls are tools and procedures used to avert threats and mitigate risks to a computing system. They may come in a number of forms, and be of a physical or logical nature. Examples of physical controls may be guards at the entrances to a building or locks on the doors to a computing facility. These types of controls are fairly well understood, because like the physical threats described above, they apply not only to computers, and have been used for a long time in other environments.

Logical controls may be a bit more difficult to understand. The simplest example is probably the nearly ubiquitous user ID and password, which users must enter to access their accounts on certain Web sites or on their office computers. Another example of a logical control is the UNIX system file permissions, which govern the ability to read, modify, or execute a file. Perhaps most difficult to understand are logical controls which are embedded in a program's code, such as those which determine under what circumstances a server program relinquishes elevated privileges.

1.1 .4 Security Policy

An organization's security policy is the cornerstone of its information processing capability. It provides rules which govern the management of the computing environment. The policy should specify how user accounts are handled, e.g., who is allowed access to what resources and when. It should define the procedure for installing new programs or program updates. It should specify what to do in the event of a system compromise or intrusion.

These guidelines have been developed after careful study of industry best practices and DoD directives. Before you implement them, you must ensure that they do not conflict with your organization's existing security policy.

1.2 How to Use this Document

Use these guidelines when installing Linux for the first time on a particular machine. You may wish to read the entire document before starting the installation process to get an idea of what must be done. Since security is a continuous process, guidelines for maintaining the systems are also part of this work.

The specific instructions given are designed for the following distributions of Linux: Red Hat Linux Enterprise Server v3.0 (RHES3); SuSE Linux Enterprise Server v9 (SLES9); and Red Hat Enterprise Workstation v3.0 (RHEW3). It is possible to use the concepts presented in this document and adapt the instructions to other distributions. In any case, it is assumed that any commands given are executed by a fully privileged user, i.e. root.

In addition to this document, the authors provide an extension to the Bastille tool which automates many of the changes to the newly installed system as described herein. Another function is available which supports maintaining the system securely by validating the configuration with respect to these guidelines. Visit the SourceForge site at the URL <http://fortknox.sourceforge.net/> for details.

1.2 .1 Audience

This document is directed at system administrators and end-users of the Linux operating system wishing to limit the susceptibility of their computers to attacks. While intended primarily

to guide individuals responsible for systems used in critical infrastructure environments, this work can benefit other users as well.

As a reader of this document you are assumed to have some experience or familiarity with the installation of a Linux distribution as well as its use and administration. You should be familiar with the package installation tool used by your distribution. It is also expected that you understand the UNIX-system file permissions.

1.2 .2 Verification of Integrity

Since Linux is an open-source project, many distributions, patches, and additional software are available on public servers for download by anyone around the world. These servers are just as vulnerable to compromise as any other system. Therefore, verifying the integrity of the downloaded files is extremely important.

Perhaps the best method for performing such a test is through the use of Pretty Good Privacy (PGP) signatures. An open-source implementation of PGP is the GNU Project's Privacy Guard (GPG), which is included in most distributions of Linux. The use of GPG requires that the downloaded file has been signed by the author or vendor's private key. You will need to obtain the corresponding public key. For some distributions, the key may be stored on the media. In other cases, it must be retrieved from another source. This can easily be done by using one of the many public key servers such as `keyserver.pgp.com`, `wwwkeys.pgp.net`, or `pgp.mit.edu`, e.g.:

```
/usr/bin/gpg --keyserver pgp.mit.edu --recv-keys key-id
```

Where *key-id* is the hexadecimal identification number of the public key, such as `0xE0002FC4`. You may also find keys on Web sites, as e-mail signatures, and a variety of other places. To test the signature of a file, issue the following command:

```
/usr/bin/gpg --verify file.sig signed.file.name
```

In the above example *file.sig* is the detached signature file, usually ending in `.sig` or `.asc`, and *signed.file.name* is the downloaded file that is to be checked for integrity. For RPM files, this command should be used:

```
/bin/rpm --checksig -v file.name
```

These are only a few examples of using PGP signatures; refer to `man gpg` or the many books and Web sites devoted to this topic for more information.

If a PGP signature is not available for the file in question, try to locate an MD5 fingerprint of the file that is cryptographically signed or from a different source. Do not use an unsigned MD5 fingerprint from the same server from which you obtained the file itself; if the file was replaced by maliciously modified version, the attacker most likely replaced the MD5 fingerprint as well (this occurred in the fall of 2002 for the source code package of the `tcpdump` program, available at <http://www.tcpdump.org/>). Other sources may be a mirror site (provided it hasn't automatically mirrored the altered file and fingerprint) or an e-mail from the author or vendor directly.

2 Preparation

As stated earlier, this guide assumes you are starting with a new system, or are completely overwriting an existing installation. While it is possible to apply some of the practices to an existing system, the most benefit will be realized when you start with a clean slate. Depending on the security policies of your organization, it may be necessary to completely obliterate all previous data on all of the system's disks.

Before starting, verify the integrity of the installation media. This is particularly important if the source was a publicly available server, as in the case of a downloaded ISO image. It is also recommended that you disconnect the computer from any public networks during the installation and configuration of the operating system. This may mean that a second system be available to retrieve and verify any required patches or additional software. The transfer of these files to the target system should be done via removable media.

2.1 Hardware Clock

Keeping accurate time is important for the proper recording of events in the system log. To ensure that the system time is as accurate as possible, the installation and usage of the Network Time Protocol (NTP) daemon will be discussed later in this document. Modern PCs also have a hardware clock, which is sometimes known as the realtime clock, which will keep time while the operating system is not running or while the computer is turned off. It is this time that is used initially when the system is booted. If the hardware clock is not correct by a considerable amount, the NTP daemon will not adjust the time automatically. Therefore, it is important that the time of the hardware clock be set correctly before the NTP daemon is started. A good way to set this clock is from the BIOS set up routine. Do this before you start the installation of your Linux distribution.

The hardware clocks on UNIX systems are normally set to Universal Time Coordinated (UTC), which was previously referred to as Greenwich Mean Time (GMT). The operating system then performs the conversion to the local time based on the selected time zone. Consider this when setting the hardware clock through the BIOS set up program.

2.2 Partitions

Separate partitions for certain file systems may benefit the security of the system. Before installing a distribution, consider defining individual partitions for the following file systems and determine the amount of disk space to allocate to each:

Mount Point	Minimum Size [MB]			Remarks
	RHES3	SLES9	RHEW3	
/	50	79	200	Files required for system start-up and operation before the other file systems are mounted.
/boot	11	6	11	Kernel and other files related to the boot process.
/home	0	0	0	Files for individual users.

/opt	0	0	0	Add-on application packages.
/svr	n.a.	0	n.a.	Data used by servers running on the system (SuSE only).
/tmp	0	0	0	Temporary files, available to all users.
/usr	425	311	1,946	Most system binaries that are shared among users.
/var	13	20	38	Files that are of variable or dynamic nature, such as logs, etc.

Table 2-1: Filesystem sizes.

The size information given above represents minimum values only, and does not reflect additional space required for filesystem maintenance. For example, ReiserFS requires approximately 30 MB extra. Also, the `/home`, `/opt`, `/svr`, `/tmp`, and `/var` filesystems size requirements vary greatly, depending on the number of users and/or the services provided.

Changing the partitions after the operating system has been installed may require considerable effort, therefore it is important to define a partition scheme before the actual installation is performed.

Rationale:

Having separate partitions can help mitigate some forms of attack. Some denial of service attacks attempt to use all available disk space in a filesystem; having separate partitions for certain file systems limits the extent of such an attack. Using separate partitions also allows more fine-grained control of mount options.

Depending on the application of the system and the amount of RAM available, a swap partition may be required. The heuristic is to provide swap space that is twice the size of RAM, e.g. a computer with 256 MB of RAM should have a swap partition of approximately 512 MB. For systems with large amounts of RAM, i.e. in the GB range, the amount of swap space can be reduced. If the amount of RAM installed is very small, consider adding more memory first.

2.3 Installation of Linux Distribution

After the partitions and their sizes have been determined, you are ready to start the installation of the Linux distribution. Depending on your hardware configuration and the particular distribution, this may mean booting the computer from a floppy disk or a CD-ROM. You may also consider installation from a central server over a trusted network. Do not install the operating system from a public FTP server, this might lead to a system compromise before you have a chance to secure it. Whatever the case, familiarize yourself with the procedure required by the distribution you are using before starting.

Since this document requires certain changes to the partitioning of the disks and to the selection of the packages, the “Custom” installation option should be selected. This provides the greatest level of control over the installation process, including the ability to override the default partitioning scheme. Step-by-step instructions for the supported distributions are given in Appendix A at the end of this document.

It should be noted that no other bootable operating system may be installed on the target, i.e. you cannot have a multi-boot configuration. The foreign operating system could possibly not have the same security settings, and thus may allow an unauthorized access to critical files. That unauthorized access could then change the settings arbitrarily, defeating all efforts to harden the primary system.

2.3 .1 Packages

All modern Linux distributions include a myriad of program packages. While each may provide a feature that appears useful, it may also introduce an additional vulnerability. When selecting packages to install, you should be guided by the axiom: Install only what is required to fulfill the mission. The sections below provide some guidance with respect to package selection, grouped by the criteria required, prohibited, discouraged, and optional. Specific instructions that may be followed during the installation of each particular distribution are in Appendix A at the end of this document. The instructions given here for each package reflect what would need to be done if the installation were performed in a manner different from the one suggested. Except as noted, all packages should be contained on the distribution media.

Both Red Hat and SuSE distributions use the Red Hat Package Manager (RPM) to install, update, or remove packages. The rpm program is fairly easy to use. Its main use here is to install or remove a package. To install, the general form of the command is

```
/bin/rpm -ivh full_package_name
```

The *i* option instructs the program to install, the *v* option causes verbose output (i.e. The package name), and the *h* option produces a series of hash marks as progress indicators during the installation. The *full_package_name* includes the version and architecture information. It must also include the full path to the package file, unless the file is in the present working directory. This is assumed in the sections below.

To remove a package, the general form of the command is

```
/bin/rpm -e short_package_name
```

The *e* option removes (erases) the package. The *short_package_name* does not require any version or architecture information, as the rpm program can retrieve this from its database. No output is generated on successful operation. If removal of the package would cause problems with other packages due to dependencies, the named package will not be removed and a message regarding the dependencies will be displayed.

If a required package is already installed, or a prohibited package is not present, the respective command will display a message to that effect. To check for the existence of a particular package before issuing an install or delete command, the rpm program can be used to query the rpm database:

```
/bin/rpm -q short_package_name
```

In case the package with *short_package_name* has already been installed, the response will be the short name with version information (but no architecture information). If the package is not installed, the response will be a message to that effect.

2.3 .1 .1 Required

Several packages are absolutely necessary to afford the level of security that this document intends to provide. They are related to authentication and auditing.

Linux Audit Subsystem (LAuS)

The Linux Audit Subsystem consists of kernel patches and several user space tools to facilitate the tracking and analysis of every system call. It was developed by IBM and SuSE to certify the SuSE Enterprise Server 8 at Common Criteria evaluation level EAL3, but can be used in other distributions as well. Make sure this package has been installed.

For Red Hat Enterprise Server 3 and Workstation 3:

Red Hat does not ship this package with the installation media, but it can be installed through the Red Hat Network. As of this writing, the file is `laus-0.1-65RHEL3.i386.rpm`.

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh laus-0.2-14.17.i586.rpm
```

Network Time Protocol (NTP)

Keeping the system time accurate is important for log file analysis, especially when a central log server collects the information for several hosts. Make sure this package has been installed.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh ntp-4.1.2-4.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh xntp-4.2.0a-23.1.i586.rpm
```

A documentation package exists also. Its installation is optional.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh ntp-doc-4.1.2-4.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh xntp-doc-4.2.0a-23.1.i586.rpm
```

Rationale:

A system compromise is an undesirable event by itself. Not being able to determine how the compromise was executed is even worse, because this may mean that it can happen again. Forensic analysis of the system logs depends on accurate time information, which makes it possible to correlate events among different hosts, perhaps some of which are part of another organization.

Pluggable Authentication Modules (PAM)

PAM allows very fine-grained control of user authentication. Make sure this package has been installed.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh pam-0.75-51.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh pam-0.77-221.1.i586.rpm
```

In order to work properly with the auditing system installed above, the LAuS-enabled version of the PAM libraries must be installed.

For Red Hat Enterprise Server 3 and Workstation 3:

As with the LAuS package described earlier, Red Hat does not ship this package with the installation media, but it can be installed through the Red Hat Network. As of this writing, the file is `laus-libs-0.1-65RHEL3.i386.rpm`.

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh pam-laus-0.77-4.3.i586.rpm
```

This will overwrite the original PAM package, but the RPM database will still list it to keep certain dependencies satisfied. Do not later remove, reinstall or update the plain PAM package.

GNU Privacy Guard (GPG)

Applying patches to fix program vulnerabilities will only enhance the security of your system if the integrity of the patches can be verified. The best way to do this is by checking cryptographic signatures of these packages with GPG (an open-source implementation of Pretty Good Privacy, PGP). Make sure this package has been installed.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh gnupg-1.2.1-4.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh gnupg-1.2.4-68.1.i586.rpm
```

2.3 .1 .2 Prohibited

DoD Instruction 8500.2 requires that passwords may not be transmitted or stored in clear-text. This prohibits the use of protocols which do not encrypt such sensitive information, e.g. TELNET, FTP or the Berkeley r-commands (`rlogin`, `rsh`, etc.). It is best to not install or to remove these packages altogether. Strict adherence to DoD Instruction 8500.2 requires that neither client nor server programs be used. Many of the newer distributions do not install these packages by default. If this is the case on your system, you may safely ignore any error message regarding that fact.

TELNET

The TELNET protocol transmits all data, including the authentication information, without encryption. Neither the client nor the server programs may be used.

```
/bin/rpm -e telnet telnet-server
```

File Transfer Protocol (ftp)

FTP also transmits all data without encryption. Neither the client nor the server programs may be used.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -e ftp
```

For SuSE Enterprise Server 9:

```
/bin/rpm -e lukemftp ncftp tftp
```

```
/bin/rom -e pure-ftpd
```

Berkeley r-commands (rsh)

These commands transmit or store passwords without being encrypted, and have a long history of security abuses. They are best removed completely.

```
/bin/rpm -e rsh rsh-server
```

Rationale:

Passwords that are transmitted in cleartext may be intercepted by sniffer programs. Even within a trusted network it is possible that one or more hosts have been compromised, and a sniffer installed. In addition, the r-commands provide a feature called “trusted hosts” which allows user authentication without a password, based only on the IP address of the client system. Since IP addresses can be spoofed, such authentication should not be allowed.

2.3 .1 .3 Discouraged

Certain packages may provide useful features, but may also introduce vulnerabilities. Thus it is imperative to carefully evaluate the need for these features, and only make them available if they are a mission-critical necessity. The following packages should be removed if they fail this test:

X Window System

This is actually a collection of packages. The foundation is the X server, which consists of several packages that have a common base name. A desktop such as Gnome or KDE is also usually included. Removing all of these packages may prove difficult, since there are many dependencies between them. It is best to make sure that none of these packages is selected during installation of the operating system. You can effectively disable the X server by removing the base package, forcing rpm to ignore any dependencies.

```
/bin/rpm -e --nodeps XFree86
```

Dynamic Host Configuration Protocol (dhcp)

If possible, avoid using this protocol. It requires no authentication, neither on the client nor on the server side.

```
/bin/rpm -e dhcp dhcpcd dhcp-server
```

Rationale:

The lack of authentication allows an attacker with physical access to the network to easily add a client or server machine. A rogue server could be used to harvest authentication information from legitimate users.

2.3 .1 .4 Optional

There are a few packages which provide secure operation of features that may not be required on every system

General Purpose Mouse (gpm)

Install this package if you wish to use the mouse with text-only systems, i.e. when the X Window System is not installed. The package has had no significant security issues since the end of 2001.

For Red Hat Enterprise Server 3:

```
/bin/rpm -ivh gpm-1.19.3-27.2.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh gpm-1.20.1-301.1.i586.rpm
```

Superuser Do (sudo)

This package allows unprivileged users to execute certain privileged commands. The system administrator can narrowly define which users can perform which specific commands. By using sudo, some administrative tasks can be delegated to different individuals without giving them access to the root account.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh sudo-1.6.7p5-1.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh sudo-1.6.7p5-117.1.i586.rpm
```

Secure Shell (SSH)

To replace TELNET with a more secure means of accessing a host remotely, the Secure Shell protocol and programs were developed. These programs communicate between the client and host computers using encryption, offering an eavesdropper no useful information. The package also contains a program for the transfer of files, which can be used as a replacement for the FTP programs. If remote access to the system is a mission-critical requirement, you may choose to install the Secure Shell server package.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh openssh-server-3.6.1p2-18.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh openssh-server-3.8p1-37.9.i586.rpm
```

Accessing other systems securely is possible with the Secure Shell client package.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh openssh-3.6.1p2-18.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh openssh-3.8p1-37.9.i586.rpm
```

Expanded Internet Daemon (xinetd)

This “superserver” has been traditionally used to enhance the security of some of the older Internet protocols such as finger or rlogin. Since all of these services should be turned off, inetd or the newer xinetd may not be needed. On the other hand, it could be used to allow certain services that require access to privileged ports (those in the range below 1024) to run as a non-privileged user. Be aware that this usage may not work with all services, and should be considered experimental.

For Red Hat Enterprise Server 3 and Workstation 3:

```
/bin/rpm -ivh xinetd-2.3.12-2.3E.i386.rpm
```

For SuSE Enterprise Server 9:

```
/bin/rpm -ivh xinetd-2.3.13-39.3.i586.rpm
```

2.4 Apply Patches

After the operating system has been installed from the original media, any available security-related patches should be applied. These are normally available on the vendor’s Web site. Since the configuration of the system has not been completed, it is necessary to retrieve these patches using a second system that can be connected to the public network. As with any other software obtained in this manner, the integrity of the patches must be verified as described above before they are applied or even transferred to the target system.

3 Configuration

Once the operating system has been installed, additional steps must be performed which will further enhance its security. These steps typically affect the system configuration files stored in the `/etc` subtree, but may also change the permission settings on critical files stored in various locations.

3.1 System Initialization

During system initialization a computer is quite vulnerable to attack since none of the safeguards provided by the operating system can have any effect. Typically, a personal computer will start by executing code in its Basic Input/Output System (BIOS), which has very limited functionality. Its job is to read the boot loader from a mass storage device, which is normally a hard disk but may also be removable media such as a floppy disk or CD-ROM. Some of the behavior of the BIOS can be customized through settings stored in nonvolatile RAM (NV-RAM). The boot loader in turn will read the actual kernel from the same or a different mass storage device, and transfer control to it.

3.1 .1 BIOS

To prevent unauthorized individuals from booting the computer with an arbitrary operating system on removable media, the appropriate option, if available, should be set in the computer's BIOS configuration. You must consult the manufacturer's documentation for the procedure, as this may be different from one system type to the next.

While you are setting up the BIOS, you may consider disabling any unused serial or parallel ports. This reduces the chance of someone attaching an external modem to the serial line or connecting certain external mass storage devices to the parallel port.

Since anyone who is able to insert a floppy disk or a CD-ROM would be physically present at the computer and therefore also have the ability to change the BIOS configuration, it is necessary to password-protect this facility. Choose a good password that is different from other passwords used on the machine. Ideally, the password should be unique. How much effort you expend on this is related to the level of physical security provided for the system. Also, be aware that some manufacturers have configured an additional default password to facilitate customer service in the event the custom password is forgotten. These passwords may be known to attackers.

Keep in mind that an isolated system is most vulnerable, since an attacker might be able to open the computer's case and reset the BIOS configuration by using a switch or by temporarily removing the CMOS battery. In such situations, where there are no other individuals in the vicinity to discourage physical tampering, it may be required to provide a case lock or even an alarm.

3.1 .2 Boot Loader

There are two boot loaders in common use on Linux systems: The older Linux Loader (LILO) and the more advanced Grand Unified Boot-Loader (GRUB). Both allow the user to select different operating systems or kernel versions to boot, if they are available. In addition, both allow customized parameters to be passed to the kernel, such as booting into the single-user run-level, which would give superuser access without requiring a password. To prevent

unauthorized users from changing the parameters, the boot loaders can be configured to require a password in that case.

LILO can be configured to require a password if parameters are to be passed to the kernel. The configuration file `/etc/lilo.conf` must be modified to include these two lines in the global section:

```
restricted
password=cleartext_password
```

Note that the password is stored in cleartext, thus it is important to restrict access to the file with proper permissions:

```
/bin/chmod 600 /etc/lilo.conf
```

GRUB can store the password in its configuration file as an MD5 hash. Perform the following steps to do this:

1. Convert the plain-text password to an MD5 hash by issuing this command and entering the password at the prompt. The output of the command is the MD5 hash.

```
/sbin/grub-md5-crypt
```

2. Edit the configuration file `/etc/grub.conf` by adding or replacing an existing password line with this line:

```
password -md5 hashed-password
```

Replace *hashed-password* with the output of the first command. If you have `gpm` installed, you can cut-and-paste to minimize the chance of making a mistake while transcribing the hashed password.

3.1 .3 Init

The init task is the first process (PID 1) started by the kernel after booting. It reads its configuration from the file `/etc/inittab`. There are a few security-related changes that need to be done to this file.

By default, the single-user mode requires no password. This must be changed by adding the following line to the configuration file:

```
su:S:wait:/sbin/sulogin
```

Now the root password is required when entering single-user mode. Of course, you must be careful not to forget that password, since you no longer have any way of changing it without knowing it.

Rationale:

Allowing access to a system without a password is a serious breach of security.

A change that many sources recommend, but which will not be applied here, is the disabling of `ctrl-alt-delete` to reboot.

Rationale:

An individual who has the ability to enter `ctrl-alt-delete` would necessarily have physical access to the target computer. Thus this person could effect a reboot by cycling power to the machine, perhaps even as crudely as by removing the power cord from the mains

socket and then replacing it. This kind of power cycling could have disastrous effects on the data stored on the disks, and perhaps even to the hardware itself. A controlled reboot, such as initiated by ctrl-alt-delete, is preferable.

To make init reload its configuration file and thus have the above changes take effect immediately, enter the following command:

```
/sbin/init q
```

3.2 Accounts

Several user accounts and groups are created during installation that are not needed and have been included for historical reasons. The following unused users and groups must be removed:

Distribution	Users	Groups
SLES9	games uucp	dialout games modem uucp xok
RHES3	games gopher	
RHEW3	games gopher	

Table 3-1: Historical users and groups.

To remove the users, issue the following command for each user to be removed:

```
/sbin/userdel user
```

To remove the groups, issue the following command for each group to be removed:

```
/sbin/groupdel group
```

Rationale:

Purging unused entries in the `/etc/passwd`, `/etc/shadow`, and `/etc/group` files facilitates detection of any unauthorized additions.

If the deleted users and groups owned any files on the system, they should be assigned to root by issuing these commands:

```
/usr/sbin/find / -nouser -exec /bin/chown root {} \;  
/usr/sbin/find / -nogroup -exec /bin/chgrp root {} \;
```

3.2 .1 Superuser Access

Which locations the superuser may log in from are defined in the `/etc/securetty` configuration file. It should contain entries for the console and possibly virtual consoles only. They are identified by `ttyn`, where `n` is usually a number in the range 1-8. If your system uses devfs, the entries are of the form `vc/n`. There should be no entries that look like `pts*`, which would allow root logins over the network. If the file contains such entries, they must be deleted.

It is important to note that if `/etc/securetty` does not exist, then root can log in from anywhere. If it exists, but is empty, then root cannot log in at all. Of course it is still possible to login as an unprivileged user and then use `su` to become root. In some cases this behavior may be required, since using the `su` command will create an entry in the system log. For instance, DoD Instruction 8500.2 specifies the principle of non-repudiation, which is only established by proper audit records. Your organization's security policy may have similar requirements.

3.3 Account Passwords

Use MD5 or bigcrypt password hashes in the `pam_unix` module. The passwords should be set to expire in 60 days, with a 5-day advance warning to the user. If the user does not change the password within a week (7 days) after the password expires, the account should be locked, requiring system administrator intervention before being enabled again.. Use a command similar to the one given below to set the aging requirements in the `/etc/shadow` file.

```
/usr/bin/chage -M 60 -W 5 -I 7
```

Another way to enforce password aging is by setting system-wide values in the `/etc/login.defs` configuration file. These directives would accomplish the same as the `chage` command:

```
PASS_MAX_DAYS      60
PASS_WARN_AGE     5
```

3.3 .1 DoD Requirements

To ensure the confidentiality of sensitive or classified information, DoD Instruction 8500.2 requires that passwords:

- are case sensitive
- are at least 8 character in length
- include at least one upper case letter: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- include at least one lower case letter: abcdefghijklmnopqrstuvwxyz
- include at least one numeral: 0123456789
- include at least one special character: !"#%&'()*+,-./:;<=>?[\]^_`{|}~

In addition, password aging should be implemented, for example as described above, and the new password must differ by at least four characters. Finally, the password must be checked to ensure that it is sufficiently strong to resist attacks.

Another requirement of DoD Instruction 8500.2 addresses default passwords, which must be removed or changed. Since the installation programs of modern Linux distributions do not supply default passwords, this issue is not much of a concern.

3.3 .2 Enforcing Strong Passwords

There are PAMs available which can be applied to ensure that users will select passwords that are not easily cracked. The `pam_passwdqc` module can be used to implement some of the DoD requirements described above. It is stacked on the password changing module in the `/etc/pam.d/passwd` or `/etc/pam.conf` file. The relevant section of the former file should look like this:

```
password required \
  pam_passwdqc.so min=disabled,disabled,disabled,disabled,8,random=0
password sufficient pam_unix.so nullok use_authok md5 shadow
password required pam_deny.so
```

3.4 Login Banner

It is common practice for a system to display a message to a user before asking for authentication information. This login banner often contains information about the system itself, such as what version of the operating system is running on what hardware. It is best to change this message to be terse, and provide no information about the system at all. This will make it more difficult for an attacker to apply methods for known vulnerabilities of the respective operating system.

In addition, the login banner must state that use of the system is only allowed to authorized personnel in pursuance of official duties. Your organization may have other requirements, check your security policy for guidance. Sample login banners are available in Appendix B at the end of this document.

To modify the login banners, two files must be edited. The banner that is displayed when a user logs in at the console or at a terminal connected directly to the machine is stored in the file `/etc/issue`. Remote logins over the network use the file `/etc/issue.net`. Both files contain the banners as plain text and can be changed with any text editing program.

3.5 Resource Limits

The PAM module `pam_limits` enforces specific limits for users and groups as specified in its configuration file `/etc/security/limits.conf`. Some of the resources that can be limited are core file size, resident set size, CPU time in seconds, and number of processes. The limits may be defined as hard or soft, where the former is fixed while the latter is a default lower value. The following example shows how to disable core files, set the resident set size to 10 MB, and limit the number of processes to 50:

```
*          hard    core    0
*          hard    rss     10000
*          hard    nproc   50
```

See `man limits.conf` for more information.

3.6 Auditing

Audit records are an important tool which allow the system administrator to check for potential security breaches. They may also become part of the evidence in a forensic investigation of a system compromise. For these reasons, the system logging facility must be enabled. If possible, have the event log records sent to a central logging host. Thus, if the system is compromised and the intruder attempts to alter the log file contents to obscure his or her activities, a copy is available on another machine.

3.6 .1 DoD requirements

DoD Instruction 8500.2 specifies that audit records include:

- User ID
- successful and unsuccessful attempts to access security files
- data and time of event
- type of event
- success or failure of event
- successful and unsuccessful logons
- denial of access resulting from excessive number of logon attempts
- blocking of blacklisting a user ID, terminal or access port, and the reason for the action
- activities that might modify, bypass, or negate safeguards controlled by the system

For classified systems additionally:

- data required to audit the possible use of covert channel mechanisms
- privileged activities and other system-level activities
- starting and ending time for access to the system
- security relevant actions associated with periods processing or the changing of security labels or categories of information

Configure the audit subsystem by providing the configuration file `/etc/sysconfig/audit` with the following contents:

```
AUDIT_ALLOW_SUSPEND=1
AUDIT_ATTACH_ALL=0
AUDIT_MAX_MESSAGES=1024
AUDIT_PARANOIA=0
```

It is a requirement of DoD Instruction 8500.2 that audit records should be retained for at least one year. The log files must be protected against unauthorized access, modification, or deletion by setting appropriate file access permissions. They should be backed up to a different system or media at least once a week. Be aware that all of these requirements demand a large amount of storage space.

3.7 Filesystem

Protecting the integrity of the stored data is one of the primary objectives of computer security. By adjusting some of the controls provided by the filesystem, it is possible to increase the level of protection by following industry accepted best practices.

3.7 .1 Set Mount Point Options

The partitions created initially allow you to control the mount options more closely. Change the file `/etc/fstab` to reflect the settings below (not all fields are shown):

```
/dev/partition / defaults
/dev/partition /boot nodev,ro
/dev/partition /home nodev,nosuid
/dev/partition /opt nodev,nosuid,ro
/dev/partition /tmp nodev,noexec,nosuid
/dev/partition /usr nodev,ro
/dev/partition /var nodev,noexec,nosuid
```

Replace *partition* in each line above with the relevant device file, e.g. sda1 or hda3. Some systems may use the format LABEL=*label* instead of the device file in the first column. Note that the ro option for the /usr mount point will make system updates a bit difficult, since it will require that the option is changed to rw before the update and back to ro afterward. The same may apply to the other file systems that have the ro option set. If you will be setting up your system to perform automatic updates, then you must not give the ro option.

Also, be aware that the noexec option does not entirely prevent execution of files on that filesystem. A knowledgeable attacker may still run a program on /tmp or /var by using the linking loader ld-linux.so.*, for instance as in this command:

```
/lib/ld-linux.so.1 /tmp/someprog
```

Consider the noexec option to be a control against accidentally executing a rogue program in the file systems so protected.

3.7 .2 Administrative Commands

Certain programs are reserved for use by root only and are stored in the /sbin and /usr/sbin directories. Unprivileged users should not be allowed to have access to these files. Issue the following command to revoke read, write, and execute permissions for all other users:

```
/bin/chmod -R o-rwx /sbin /usr/sbin
```

3.7 .3 setuid/setgid Programs

Vulnerabilities in programs that are owned by root and have file permissions set that allow ordinary users to run these programs as user or group root are the cause of most escalation of privilege exploits. The smaller the number of programs with such permissions, the smaller the chance of an attacker gaining root privilege. Only the following programs may have the SUID permission bit set:

```
/bin/ping
/bin/su
/usr/bin/at
/usr/bin/chage
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/gpasswd
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
/usr/bin/lpstat
/usr/bin/passwd
```

Remove the SUID and SGID bits on all other files that may have them set. Use the following command to find such files:

```
/usr/bin/find / -perm +6000 -type f -exec ls -la {} \;
```

If a user must run one of these programs as root, consider using the sudo facility for that program.

3.7 .4 Other File and Directory Permissions

Files and directories that can be modified by any user, i.e. that are world writeable, are a security concern. Check which world writeable directories or files exist by issuing the following command:

```
/usr/bin/find / -perm +0002 \( -type d -o -type f \) -exec ls -la {} \;
```

The command should produce a very short list. The only directory which is meant to be world writeable is `/tmp`. If there are any other directories which have the write permission bit set for all users, carefully evaluate if this is necessary.

The Red Hat Package Manager (rpm) is a powerful tool that allows full control over the program packages that are installed on a system. Only the superuser should be able to install packages. Change the permission of rpm program so that only root may execute it.

```
/bin/chmod 700 /bin/rpm
```

3.8 Services

As configured according to these guidelines, you will have a minimal number of services running on your system. This is the best way to provide security. Some of these services must be configured to operate properly.

3.8 .1 ntpd

Configuring the Network Time Protocol daemon is straightforward. It must know the host name or IP address of one or more NTP servers that it can use to synchronize the system time; it is customary to specify three hosts, with the possibility to indicate which one is preferred. The file `/etc/ntp.conf` contains the configuration information, refer to the documentation provided with the NTP package for details. There are several public servers available which can be freely used as upstream references. See <http://www.eecis.udel.edu/~mills/ntp/servers.html> for lists of stratum 1 and stratum 2 servers. The organizations which operate these systems will typically require that you notify them of your connection. Of course, if you have an NTP server available within your own organization or perhaps even a stratum 0 time source, you should use that as your preferred server.

3.8 .2 sshd

If you elected to install the SSH server package, perform the steps described here. The SSH daemon is configured through the settings in the file `/etc/ssh/sshd_config`. Some changes must be made to ensure a high level of security; the most important one is disabling version 1 of the SSH protocol. It has a history of security problems. The configuration file must contain these settings:

```
Protocol 2
PermitRootLogin no
PermitEmptyPasswords no
IgnoreRhosts yes
HostBasedAuthentication no
PasswordAuthentication yes
X11Forwarding no
```

3.9 Document the Settings

After the system has been completely configured, it would be wise to document the critical settings. This will allow you to quickly repair a system that has been damaged by hardware failure, remote compromise, or some other cause.

The BIOS settings are critical to proper startup and operation of the computer. Copying the settings on paper is the best approach, since an improperly configured BIOS may not allow the computer to start up at all. Having a hardcopy will allow you to restore the working configuration easily. You may also want to consider using a utility program which can save the entire CMOS NV-RAM contents to a file. To find such utilities, search the Internet for the terms “Linux”, “CMOS”, and “backup”. It is prudent to test such a program before relying on it in the event of an emergency.

Document the partitioning of the physical disks by issuing the following command for each of the disks in your computer:

```
/sbin/fdisk -l /dev/device-file
```

Replace *device-file* with the name of the disk device file, e.g. hda or sda. You may redirect the output of the above command to a file which you can copy to another medium or print out, or both. Remember that if the partition table is corrupted, you won't be able to access any files on the disk, but if you have a hardcopy, you can recover the table more easily.

Having a list of all installed packages with the associated version information is also important. Issue the following command to create a file containing the required data:

```
/bin/rpm -qa --last > baseline_package_list
```

This will produce the desired list including the installation date of each package. To produce an alphabetized list, pipe the output of the rpm command through /bin/sort before redirecting to the file. You may want to include date information in the name of the file itself to identify when the list was created, and repeat the procedure each time you install a new or updated package.

Future checks for unauthorized setuid/setgid files are simplified if you make a list of these files at this time. Issue the following command to create a file with a list of the files having their SUID or SGID permission bit set:

```
/usr/bin/find / -perm +6000 -type f -exec ls -la {} \; > suid-sgid_file_list
```

For added security, store the *suid-sgid_file_list* on removable media or on a different system. That way an attacker would not be able to modify your original list.

3.10 Restart the System

You are now ready to restart the system. This will stop all unnecessary services, and ensure that the system uses the new settings. Before you connect the system to an untrusted network, you may wish to make a full backup of all files. A simple way of doing this is to use the tar utility:

```
/bin/tar -cpvf backup_file /
```

This command will backup the entire filesystem to *backup_file*, which could be a SCSI tape drive identified by the device file /dev/st0.

UNCLASSIFIED

As a last step, determine which services will be running at the different runlevels by reviewing the output of the following command:

```
/sbin/chkconfig --list
```

You should see something like this (list has been truncated):

```
syslog          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
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
atd             0:off 1:off 2:off 3:on  4:on  5:on  6:off
keytable        0:off 1:on  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
xinetd based services:
  finger:      off
  echo:        off
```

Verify that the list is what you expect. In particular, check that only those services which you wish to provide are marked “on” at any runlevel.

4 Maintenance

After your system has been installed and configured, you must periodically ensure that it is still secure. Basically, this involves confirming that the settings applied in the previous chapter have not been altered. While most of these checks can be automated, a few must be performed manually.

4.1 Accounts

When an individual is no longer authorized to access a particular computer, his or her account should be promptly disabled. Another person within the organization should be determined to whom the files of the former can be assigned. Then the account can be deleted.

Checking for unknown accounts can be useful in the discovery of a possible system compromise. This must be done manually by examining the `/etc/passwd`, `/etc/shadow`, and `/etc/group` files.

4.2 Filesystem

While the configuration section above described how to initially harden a system, users of the system may introduce vulnerabilities, either unintentionally or intentionally, during day-to-day operations.

4.2 .1 setuid/setgid Programs

Check that no new files with the SUID or SGID permission have been added to the system by running this command:

```
/usr/bin/find / -perm +6000 -type f -exec ls -la {} \;
```

Compare the output with that generated by the same command after the initial configuration was complete. The two lists should be the same.

4.3 Apply Patches

Most distributions provide a method of automatically checking for and applying patches to the installed software. While being notified of available updates is wise, having these changes applied without first testing may not be desirable, particularly on a production system. Some updates have been known to render certain systems unusable. It is also possible that an update will introduce a new vulnerability. Finally, there is always the possibility that the vendor's servers have been compromised. An attacker may have replaced important patches with altered versions. For this reason, it is important that you always verify the integrity of a package you are about to install on your system.

In some cases a patch may not be available for a package that has been determined to have a vulnerability. Unless the risk of running an unpatched service is considerably less than not offering the service at all, it would be wise to disable it until the vendor releases a fix.

4.4 Backups

Many resources exist that address the issue of making backup copies of a system's data and programs. It may be helpful to review some of the basic principles here.

The first thing to consider is what kind of backup medium you will use. For smaller operations, where the amount of data to be stored is relatively small and in the range of a few gigabytes, backing up to writeable CD-ROM or DVD may be a solution. As the amount of data increases, magnetic disk or tape will be required. Fixed disks with high capacities, e.g. in the 200 GB range, are available cheaply and may be good insurance against user error or a malicious attack. Their disadvantage is that these media are usually not removable, and even if so, are not very robust. Removable media are required for disaster prevention, since they can be moved off site. The best backup procedure will not be very effective if the backups are lost in a fire along with the production system. The remainder of this section will assume that a tape drive is used.

Two main types of backup can be used. A full backup copies all important files and directories, while an incremental backup copies only those files that have been modified since the last full backup. The former allows quick restoration of the files while requiring the most space and time, while the latter will require more effort to restore certain files but will conserve space and can be performed more quickly. Most backup schemes combine both methods, such as performing an incremental backup each night, and a full backup once a week, perhaps on the weekend. Depending on the budget and space available, tapes may need to be reused. At a minimum, you should use three tapes. This will allow you to make three full backups before recycling the tapes.

Simple backups can be effectively performed using the tar utility. For example, to copy all the files in the /etc and /usr file systems to a SCSI tape drive, use this command:

```
/bin/tar -cpvf /dev/st0 /etc /usr
```

The meaning of the options are: c - create the archive; p - preserve all permission information; v - list all files processed (verbose); and f - must precede the name of the file to be created in this case /dev/st0.

There are more sophisticated solutions to the backup problem, some are commercial products while some are freely available. An example of the latter is the Advanced Maryland Automatic Network Disk Archiver (Amanda). It allows scheduled backups of multiple client systems on network backup servers, and supports many different tape drives including tape library systems. More information is available at <http://www.amanda.org/>.

Regardless of the method you use to create backups, it is important that you verify them. Depending on the amount of time required, you may choose to do this every time you back up, or one a periodic basis (e.g. once a month). In addition to verifying, you should also periodically confirm that you can restore data from the backup tapes.

Appendix A

Detailed Installation Instructions

For SuSE Enterprise Server 9:

1. Select the “Installation” option from the initial screen presented when the system is booted from the first CD-ROM.
2. Accept the license agreement.
3. Choose the installation language.
4. If a previous installation of Linux exists, you will be offered a number of options. Choose “New installation” in this case.
5. The following screen will show the installation settings, select the link “Partitioning”.
6. Choose “Create custom partition setup” from the “Suggested Partitioning” screen..
7. Choose “Custom partitioning--for experts” on the screen titled “Preparing Hard Disk--Step 1”.
8. A new screen will show the physical disk(s) present with any existing partitions. Delete any such existing partitions first. Then press the “Create” button, which will start a new dialog for the partition parameters.
9. If you have more than one physical disk installed, choose the one you wish the new partition to be created on.
10. Select “primary” or “extended” partition type. Note that there can be only four primary partitions on a physical disk. If you need more than four partitions, you must create an extended partition, and define logical partitions within it.
11. Enter the relevant mount point and size information. Leave the “Format” radio button selected, and the “ReiserFS” chosen in the drop-down list for “File system”, or select these if they are not so already. You may also select “Ext3”, “JFS”, or “XFS” as a file system. Note that the Reiser file system requires 30 MB overhead.
12. Perform the same operations for each of the partitions defined.
13. Back on the “Installation Settings” screen, select the “Software” link.
14. Choose “Minimum System”.
15. Get a “Package Group” listing by selecting the appropriate filter.
16. Find the group “Hardware/Modem”.
17. Deselect the “providers” package.
18. Find the group “Productivity/Networking/Boot/Clients”.
19. Deselect the “dhcpcd” package.
20. Find the group “Productivity/Networking/Diagnostic”.
21. If desired, select the “snort” package.
22. Find the group “Productivity/Networking/Ftp/Clients”.
23. Deselect the “Lukemftp” package.
24. Find the group “Productivity/Networking/Other”.
25. Mark the “rsh” package taboo.
26. Mark the “rsh-server” package taboo.
27. Mark the “telnet” package taboo.
28. Mark the “telnet-server” package taboo.
29. Select the “xntp” package.

For Red Hat Enterprise Server 3:

1. Read the “Welcome” screen and move on to the next screen to choose the installation language.
2. The following two screens allow you to choose the keyboard layout and the mouse type.
3. Select “Manually partition with Disk Druid” on the “Disk Partitioning Setup” screen.
4. A new screen will show the physical disk(s) present with any existing partitions. Delete any such existing partitions first. Then press the “New” button, which will start a new dialog for the partition parameters.
5. Enter the relevant mount point and size information. If more than one physical disk is present, select the disk or disks on which the new partition may be created. Leave “ext3” chosen in the “File System Type” drop-down list, or select it if it is not so already. Under the heading “Additional Size Options”, ensure that the “Fixed size” option is checked.
6. Perform the same operations for each of the partitions defined.
7. On the “Boot Loader Configuration” screen, select the “Use a boot loader password” option and enter a unique password for the boot loader. This password will be required when options are given to the kernel.
8. The “Network Configuration” screen allows you to enter network parameters. You should not be using DHCP, so you will need to edit the network devices and enter a static IP address and an appropriate netmask. You must also manually define the hostname and specify the gateway and DNS IP addresses.
9. Leave the default “Enable firewall” option checked on the “Firewall” screen. Verify that no services are checked.
10. Select the system language on the “Additional Language Support” screen.
11. The “Time Zone Selection” screen allows you to set your time zone by clicking on a world map. Be sure to check the “System clock uses UTC” option at the bottom of the screen.
12. Enter the root password on the “Set Root Password” screen. Be sure to select a strong password.
13. On the “Package Defaults” screen, select “Customize the set of packages to be installed”.
14. On the “Package Group Selection” screen, locate the “Miscellaneous” heading and select the “Minimal” package group.
15. Complete the installation as instructed, changing installation media as required. When the installation is done, the system will be rebooted.
16. Log in as root and issue the following commands to remove undesirable packages:
 - a. `/bin/rpm -e telnet`
 - b. `/bin/rpm -e ftp`
 - c. `/bin/rpm -e rsh`
17. Reboot the system

For Red Hat Enterprise Workstation 3:

1. Read the “Welcome” screen and move on to the next screen to choose the installation language.

2. The following two screens allow you to choose the keyboard layout and the mouse type.
3. Select “Manually partition with Disk Druid” on the “Disk Partitioning Setup” screen.
4. A new screen will show the physical disk(s) present with any existing partitions. Delete any such existing partitions first. Then press the “New” button, which will start a new dialog for the partition parameters.
5. Enter the relevant mount point and size information. If more than one physical disk is present, select the disk or disks on which the new partition may be created. Leave “ext3” chosen in the “File System Type” drop-down list, or select it if it is not so already. Under the heading “Additional Size Options”, ensure that the “Fixed size” option is checked.
6. Perform the same operations for each of the partitions defined.
7. On the “Boot Loader Configuration” screen, select the “Use a boot loader password” option and enter a unique password for the boot loader. This password will be required when options are given to the kernel.
8. The “Network Configuration” screen allows you to enter network parameters. You should not be using DHCP, so you will need to edit the network devices and enter a static IP address and an appropriate netmask. You must also manually define the hostname and specify the gateway and DNS IP addresses.
9. Leave the default “Enable firewall” option checked on the “Firewall” screen. Verify that no services are checked.
10. Select the system language on the “Additional Language Support” screen.
11. The “Time Zone Selection” screen allows you to set your time zone by clicking on a world map. Be sure to check the “System clock uses UTC” option at the bottom of the screen.
12. Enter the root password on the “Set Root Password” screen. Be sure to select a strong password.
13. On the “Package Defaults” screen, select “Customize the set of packages to be installed”.
14. On the “Package Group Selection” screen, locate the “Applications” heading and select package groups as desired. You can choose packages within a group by following the “details” link.
15. Locate the “Servers” heading, and ensure that none of the package groups is selected.
16. Locate the “Development” heading and select package groups as desired.
17. Complete the installation as instructed, changing installation media as required. When the installation is done, the system will be rebooted.
18. Log in as root, open a terminal window with System Tools|Terminal and issue the following commands to remove undesirable packages:
 - a. `/bin/rpm -e telnet`
 - b. `/bin/rpm -e ftp`
 - c. `/bin/rpm -e rsh`
19. Close the terminal window.
20. Reboot the system.

UNCLASSIFIED

This page intentionally left blank.

Appendix B

Sample Login Banners

U.S. Department of Defense computer system display this banner. While it may not be terse, it does eliminate all references to the deployed hardware and software.

THIS IS A DEPARTMENT OF DEFENSE COMPUTER SYSTEM. THIS COMPUTER SYSTEM, INCLUDING ALL RELATED EQUIPMENT, NETWORKS AND NETWORK DEVICES (SPECIFICALLY INCLUDING INTERNET ACCESS), ARE PROVIDED ONLY FOR AUTHORIZED U.S. GOVERNMENT USE. DOD COMPUTER SYSTEMS MAY BE MONITORED FOR ALL LAWFUL PURPOSES, INCLUDING TO ENSURE THAT THEIR USE IS AUTHORIZED, FOR MANAGEMENT OF THE SYSTEM, TO FACILITATE PROTECTION AGAINST UNAUTHORIZED ACCESS, AND TO VERIFY SECURITY PROCEDURES, SURVIVABILITY AND OPERATIONAL SECURITY. MONITORING INCLUDES ACTIVE ATTACKS BY AUTHORIZED DOD ENTITIES TO TEST OR VERIFY THE SECURITY OF THIS SYSTEM. DURING MONITORING, INFORMATION MAY BE EXAMINED, RECORDED, COPIED AND USED FOR AUTHORIZED PURPOSES. ALL INFORMATION, INCLUDING PERSONAL INFORMATION, PLACED ON THIS SYSTEM OR SENT OVER THIS SYSTEM MAY BE MONITORED. USE OF THIS DOD COMPUTER SYSTEM, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES CONSENT TO MONITORING OF THIS SYSTEM. UNAUTHORIZED USE MAY SUBJECT YOU TO CRIMINAL PROSECUTION. EVIDENCE OF UNAUTHORIZED USE COLLECTED DURING MONITORING MAY BE USED FOR ADMINISTRATIVE, CRIMINAL OR ADVERSE ACTION. USE OF THIS SYSTEM CONSTITUTES CONSENT TO MONITORING FOR THESE PURPOSES.

UNCLASSIFIED

This page intentionally left blank.

Appendix C

License Information

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect

UNCLASSIFIED

making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or

else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with

UNCLASSIFIED

the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software

UNCLASSIFIED

distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY

UNCLASSIFIED

MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

UNCLASSIFIED

This is free software, and you are welcome to redistribute it under certain conditions; type ``show c'` for details.

The hypothetical commands ``show w'` and ``show c'` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ``show w'` and ``show c'`; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ``Gnomovision'` (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

UNCLASSIFIED

This page intentionally left blank.

References

- Anonymous. Maximum Linux Security. Indianapolis: Sams Publishing, 2000.
- Bauer, Michael D. Building Secure Servers with Linux: Tools & Best Practices for Bastion Hosts. Sebastopol: O'Reilly & Associates, 2003.
- Bauer, Michael D. "Paranoid Penguin: Seven Top Security Tools." Linux Journal. 13 Apr. 2004
- Coker, Faye. Getting Started with SE Linux HOWTO: the new SE Linux n.p.: n.p., December 06 2003.
- CIS Linux Benchmark vs 1.1.0. n.p.: Center for Internet Security, 2003
- Department of Defense. Directive 5215.1, Computer Security Evaluation Center. n.p.: Department of Defense, 1982.
- Department of Defense. Directive 8500.1, Information Assurance (IA). n.p.: Department of Defense, 2002.
- Department of Defense. Instruction 8500.2, Information Assurance (IA) Implementation. n.p.: Department of Defense, 2003.
- Fraser, b. Request for Comments: 2196 n.p.: SEI/CMU, 1997.
- Grubb, Steven. A Survey of Process Environments. n.p.: n.p., n.d.
- Harris Tony, Koehntopp Kristian. Linux Partition HOWTO 3.4.2 n.p.: n.p., 2001. <http://www.lissot.net/partition/Partition.html>
- Kabir, Mohammed J. Red Hat Linux Security and Optimization. New York: Hungry Minds, Inc., 2002.
- Kevin Fenzi and Dave Wreski. Linux Security HOWTO., 2000.
- Koconis, David, et al. Securing Linux: A Survival Guide for Linux Security. SANS Press, 2003.
- Koziol, Jack, et al. The Shellcoder's Handbook: Discovering and Exploiting Security Holes. Indianapolis: Wiley, 2004.
- ITSS. . 14 Apr. 2004. Stanford University. 14 Apr. 2004
- Beale, Jay. "ANYONE WITH A SCREDRIVER CAN BREAK IN! (PHYSICAL SECURITY AND BOOT SECURITY ISSUES IN LINUX)" n.p.: n.p., 20 Apr. 2000
- Leen Frisch. UNIX System Hardening Checklist. Sebastopol: O'Reilly & Associates and Exponential Consulting, 2002.
- Leen Frisch. "Hardening Linux Systems." Linux Journal. sept. 2002
- Mann, Scott, Ellen L. Mitchell, and Mitchell Krell. Linux System Security: An Administrator's Guide to Open Source Security Tools. 2nd ed. Upper Saddle River: Prentice Hall PTR, 2003.
- Molnar, Ingo. "exec shield." Online Posting. May 2003. Linux kernel mailing list. May 2004 <http://kerneltrap.org/node/view/644>
- Mourani, Gerhard. Securing and Optimizing Linux: RedHat Edition. Version 1.3. n.p.: OpenDocs Publishing, 2000.
- Mourani, Gerhard. Securing and Optimizing Linux: The Hacking Edition. Version 3.0. n.p.: OpenDocs Publishing, 2002.
- National Security Agency. Guide to the Secure Configuration of Solaris 8. Fort Meade: National Security Agency, 2003.

UNCLASSIFIED

Ott, Amon. The Rule Set Based Access Control (RSBAC)Linux Kernel Security Extension n.p.: n.p. 2001. <http://rsbac.org/linux-kongress/index.html>

PaX? Website PaX n.p.: n.p., n.d. <http://pax.grsecurity.net/docs/pax.txt>

Ranch, David A. TrinityOS: A Guide to Configuring Your Linux Server for Performance, Security, and Manageability. n.p.: n.p., 2004.

Red Hat, Inc. Red Hat Linux 9: Red Hat Linux Security Guide. Raleigh: Red Hat, Inc., 2002.

Ross, Ron, et al. NIST Special Publication 800-53, Recommended Security Controls for Federal Information Systems. (Initial public draft) Gaithersburg: National Institute of Standards and Technology, 2003.

Ross, Ron, et al. NIST Special Publication 800-37, Guide for the Security Certification of Federal Information Systems. Gaithersburg: National Institute of Standards and Technology, 2004.

Spengler, Brad. GRSECURITY ACL DOCUMENTATION V1.5 n.p.: n.p., May 2004 <http://www.grsecurity.org/papers.php>

Stoneburner, Gary, Clark Hayden, and Alexis Feringa. NIST Special Publication 800-27, Engineering Principles for Information Technology Security (A Baseline for Achieving Security). (Draft, Rev. A) Washington, D.C.: U.S. Government Printing Office, 2004.

SuSE Linux AG. SLES High Level Design. Version 2.25. n.p.: SuSE Linux AG., 2003.

Thompson Kerry, The UnOfficial SELinux FAQ n.p.: n.p., May 2004

Thorton, James. Red Hat Enterprise Linux 3. n.p.: Redhat Linux, 2003.

Wack, John, Ken Cutler, and Jamie Pole. NIST Special Publication 800-41, Guidelines on Firewalls and Firewall Policy. Washington, D.C.: U.S. Government Printing Office, 2002.

Wack, John, Miles Tracy, and Murugiah Souppaya. NIST Special Publication 800-42, Guideline on Network Security Testing. Washington, D.C.: U.S. Government Printing Office, 2003.

Weidner, Klaus. SLES Security Guide. Version 2.33. n.p.: atsec GmbH, 2003.

Wheeler, David A. Secure Programming for Linux and Unix HOWTO. 2003. <http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/index.html>

Wojtczuk, Rafal. Openwall GNU*/Linux presentation slides. 2004. <http://www.openwall.com/presentations/Owl/index.html>

Wreski, Dave. The Linux Security Administrator's Guide 1998. <http://www.linuxsecurity.com/docs/SecurityAdminGuide/SecurityAdminGuide.html>

Yale Information Security office <http://www.yale.edu/its/security/securing/unix/-workstation/index.htm>

Youman, Yves. "An overview of common programming security vulnerabilities and possible solutions." Thesis Vrije Universiteit Brussel, 2003.