



Linux Security Guidelines

Document ID: GUI-0001

Information Technology Protective Security Services Pte. Ltd.

Block K, Information Technology And State Store Building,

Jalan Gadong BE1110,

Negara Brunei Darussalam

Tel: +(673) 245 8001

Fax: +(673) 245 8002

Email: cert@brucert.org.bn

Website: <http://www.brucert.org.bn>

LEGAL NOTICE & DISCLAIMER

Do not attempt to implement any of the settings in this guide without first testing in a non-operational environment.

This document is only a guide containing recommended security settings. It is not meant to replace well-structured policy or sound judgment. Furthermore, this guide does not address site-specific configuration issues. Care must be taken when implementing this guide while using mission specific and mission critical products.

The security changes described in this document only apply to the intended operating system, service or product and should not be applied to any other software.

SOFTWARE 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, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Document Version History

Version	Reason for Issue	Date of Issue
1.00	Draft	10 th January 2005
1.03	First release	26 th January 2005
1.04	Second release	14 th February 2005

Related Documents

Document ID	Document Name

Table of content

LEGAL NOTICE & DISCLAIMER	2
DOCUMENT VERSION HISTORY	3
RELATED DOCUMENTS	3
LIST OF TABLES	5
SUMMARY	6
1. INTRODUCTION	7
1.1. PURPOSE AND SCOPE	7
1.2. ASSUMPTIONS	7
1.3. INTENDED AUDIENCE	7
2. PHYSICAL SECURITY	8
2.1. BIOS PASSWORD	8
2.2. PLACE SERVERS IN A CONTROLLED AREA	8
2.3. PREVENT SERVERS FROM BEING BOOTED THROUGH OTHER MEDIUM	8
2.4. SERVERS ARE TO BE PLACED IN RACKS WITH LOCKING MECHANISMS	9
2.5. CONCEAL CABLING AND POWER OUTLETS	9
3. INSTALLATION AND CONFIGURATION	10
3.1. INSTALL FROM A CLEAN FORMATTED DRIVE	10
3.2. PARTITIONS	10
3.3. CUSTOM INSTALLATION	10
3.4. PATCHES	11
3.5. INSTALLING PATCHES	11
4. LINUX OPERATING SYSTEM HARDENING	12
4.1. ACCOUNTS	12
4.2. ACCOUNTS POLICY	12
4.3. REMOVING UNNECESSARY ACCOUNTS	13
4.4. ROOT ACCOUNT	13
4.5. SERVICES AND PORTS	14
4.6. SECURING XINETD	15
4.7. SECURING "/ETC/SERVICES" FILE	16
4.8. DISALLOW ROOT LOGIN FROM DIFFERENT CONSOLES	17
4.9. BLOCKING SU TO ROOT	17

4.10.	TCP WRAPPERS.....	17
4.11.	IPTABLES	18
4.12.	DETECTING SUID/SGID PROGRAMS.....	18
4.13.	HIDING SYSTEM INFORMATION	19
4.14.	OTHER UTILITIES	19
4.15.	TRIPWIRE	19
4.16.	SENTRY TOOLS.....	20
4.17.	BASTILLE	20
5.	CONCLUSION	21
	REFERENCES.....	22

List of tables

TABLE 1:	RECOMMENDED PARTITION.....	10
TABLE 2:	DISABLE UNWANTED SERVICES	15

Summary

Linux has gained its popularity as an open source operating system and has been used for workstations and servers. Due to the growth of its usage in application and production line, its vulnerabilities have increased from time to time. Exploits are written based on these vulnerabilities that enable systems to be controlled over or simply knock out of its purpose.

Having Linux as a server or workstation using default configurations are potentially dangerous and should be taken into consideration. Out of the box configurations are usually configured to its generic use. Initial planning on deciding for the system intended purpose is thus crucial. These purposes are service providing system such as a web server, file server, DNS server, mail server and more.

This document guides users to enforce basic security measures to be implemented on their system. This ensures the systems are not vulnerable to attacks and to improve its ability responding to such kinds of attacks.

1. Introduction

1.1. Purpose and Scope

This guideline provides information and tips on securing Linux based operating systems. Provided users have knowledge in Linux environment, this paper covers basic guidelines in securing Linux. By practicing security measures in this guideline, users should be able to secure their system as a first step to further hardening the operating system down to its kernel.

1.2. Assumptions

This document is aimed at users that have basic knowledge in Linux console environment. Examples throughout this guideline is based and tested on Redhat 9.0. It is also preferred that users are familiar with basic console commands. This is to ensure every step taken doesn't produce unnecessary results.

1.3. Intended Audience

This guideline is intended for users who wish to implement a basic security hardening baseline for their Linux systems. This includes the 'do's and don'ts' on how to configure their Linux systems. This document is not only limited to systems administrators but also for home users that uses Linux as their daily workstations.

2. Physical Security

Physical security should not be overlooked as an optional factor. Maintaining the physical control of Linux servers/workstations are crucial to data integrity, confidentiality and availability. Unauthorized users are not allowed to power-off servers intentionally or unintentionally that may cause unnecessary downtime, thus disrupting services.

Physical control for Linux servers:

- BIOS password.
- Place servers in a controlled area.
- Prevent servers from being booted through other medium.
- Servers are to be placed in racks with locking mechanisms.
- Conceal cabling and power outlets.

2.1. BIOS Password

Setting up BIOS password protects the system configuration from being reset or altered by intruders. By placing controls to the system BIOS itself, administrators can limit the use of devices such as CD-ROMs from being use as a boot-up sequence; bootable operating system on a CD can be used to access partitions on the system.

2.2. Place servers in a controlled area

The first and most important line of defense is to place servers in a controlled environment. This includes a well ventilated room to dissipate heat generated by the servers. Room should be constantly monitored with locking access control to allow only authorized users to the area. Following steps should be followed to improvised security measures:

- Server rooms should always be locked.
- Monitoring should be both controlled via cameras and human.
- Implement access controls such as biometric or other means of logging entries.
- Servers should be visible from outside the room for operators to notice any potential threats or hazards.
- Fire suppression system must be available to control fire or electrical hazards.

2.3. Prevent servers from being booted through other medium

Servers can be booted to a different operating system through access media thus defeating security controls of that server's operating system. For example, an attacker can boot to an alternate 'live' operating system through the CD/DVD drive and access the server's partitions with write and read capabilities, bypassing the existing permissions.

Security measures can be taken as follows:

- Remove floppy or CD/DVD drives from production servers.
- Disable floppy and CD/DVD drives from BIOS settings.
- If these media are required, enforce locking mechanisms on each device.
- Disable the Linux boot loader or set the timeout to 0.

2.4. Servers are to be placed in racks with locking mechanisms

Servers should be locked firmly to its designated racks to prevent attacker to simply lift and carry the server out. By placing it firmly this minimizes the time an attacker to dismantle the racks.

Choosing suitable racks are as follows:

- Racks are to be made of heavy and durable material
- Individual locks are required for each servers in the rack
- Implement logging controls on each locks

2.5. Conceal cabling and power outlets

Cabling and power outlets are necessary to be taken into account as it is a main source of data flow and operation. Unprotected cablings may result in an attacker, who has physical access to the premises, to locate the servers' power supply or network cable and unplugging it causing downtime. Furthermore, concealing these cables minimizes the threat of being tripped over by operators causing injury or unnecessary downtime.

3. Installation and configuration

Linux installation should be planned out initially to achieve the best quality performance. Therefore, the purpose of usage is crucial to determine the necessity of packages or services to be installed. Installation of servers differs completely from a normal daily workstation.

3.1. Install from a clean formatted drive

Installation should be run on a clean formatted drive and are tested for errors. Run disk diagnostics/utilities to detect any bad sectors that may have arise. In the case of such problems arising, consider replacing the drive and run diagnostics again.

3.2. Partitions

Linux offers partitioning for its directories to protect against data loss due to corrupted partitions. Example, `/usr` directory on a different partition, `hda3`, is not affected if a partition fails or corrupts in `'hda1'`.

It is recommended to make separate partitions as follows:

Name	Recommended size	Usage
<code>/boot</code>	20-50 MB	This is where the kernel image is stored; also used to overcome bios limitations on large drives.
<code>/</code>	250MB – 3+ Gig	Root file system, all libraries, programs, and configuration files
<code>/usr</code>	500 MB +	Most applications are installed in this directory.
<code>/var</code>	250 MB – 1 Gig	System log files are kept
<code>/home</code>	250 MB – 1+ Gig	Users directories, depends on their usage

Table 1: Recommended Partition

3.3. Custom installation

Installation must be done with custom or minimal packages as possible. This prevents unnecessary services to be running on either workstations or servers. Additional packages can be installed later depending on the purpose of usage. Example, running Linux for a web server only needs packages such as Apache, PHP, OpenSSL, etc, as required. Having other services such as Sendmail (mail server) may jeopardize the web server's security.

3.4. Patches

Installing Linux should be done in an isolated network, meaning no network connection is to be active for the installation. This prevents the Linux computer of being attacked as soon as installation is finished. Updates and patches has to be downloaded from a previously hardened machine on the network which later is to be burn to CD/DVD for transferring patches to the fresh installed Linux. Verifications of patches should be taken to ensure the integrity from the original vendors. Most vendors provide MD5 checksum for their patches.

To verify integrity of a patch:

```
# md5sum <filename>
```

Output of the above command can be used to compare with the md5sum provided by the vendor. If the output matches, integrity is preserved, otherwise patch might be corrupted or tampered.

3.5. Installing Patches

Patches that are acquired should be tested on a test system before implementing it on production level. This is to ensure patches don't crash the production system resulting unnecessary downtime.

Update and patches sites differ from each Linux distributions or packages. Here are list of major packages sites.

Redhat Linux

<http://www.redhat.com/support/errata>

Mandrake Linux

<http://www.mandrakesoft.com/security/>

4. Linux Operating System Hardening

After installing and configuring, further steps have to be taken to ensure operating system security. Linux by default will execute useful services upon boot up, however most of these services are not necessary and pose a potential security risk. These steps include in removing any unnecessary services, adding rules, permissions, etc. This process is called operating system hardening.

4.1. Accounts

Linux store its user accounts information in */etc/passwd* file. Most Linux nowadays have shadow passwords enabled by default. This invokes an encryption scheme on all user passwords. In addition, another file is created to store these encrypted passwords for user accounts which can be located at */etc/shadow*.

In case shadow is not enabled, the command *pwconv* will create the shadow file based on */etc/passwd* file.

4.2. Accounts policy

User accounts needs to be enforced with strict policies to avoid user based errors. Such policies are defined as follows:

- Limit ability to access areas the system by using “groups” to categorize users
 - Use *groupadd* <groupname> command to create a group
 - Use *useradd -g* <groupname> <username> to add username to groupname or *usermod -g* <groupname> <username>
- Enforce password aging that forces users to change their passwords from time to time
 - *Chage* command is used to enforce password aging.
- Default password length allowable in Linux is 5. Change it to enforce users to choose passwords more than 8 characters for better security, takes longer time to crack.
 - # *vi /etc/login.defs*
 - Change the value of PASS_MIN_LEN 5 to PASS_MIN_LEN 8
- Enforce complex password policy like Cz!q#n.*8fF. This makes passwords difficult to be cracked that might discourage attackers.

4.3. Removing unnecessary accounts

Default system accounts which are not required should be removed immediately from the system. Example is an *ftp account* which grants user to ftp anonymously. Unless the purpose is an FTP server, this account should be removed. There are 2 ways can be used to accomplish this:

- `userdel` command is used to delete user accounts .i.e **`userdel -r ftp`** ; this will remove user account 'ftp' , home directory and files residing in it.
- Other way is by manually removing entries from **`/etc/passwd`** and **`/etc/shadow`** related to the user account.

```
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin - remove in /etc/passwd
ftp:!:12329:0:99999:7::: - remove in /etc/shadow
```

To protect against brute-force entries, root account should be renamed and replaced by a dummy account named root to mislead the attacker. Example of a **`/etc/passwd`** file that is stripped to its minimum depending on circumstances.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
mail:x:8:12:mail:/var/spool/mail:
uucp:x:10:14:uucp:/var/spool/uucp:
nobody:x:99:99:Nobody:/
```

System accounts such as news, ftp, games, etc are not required unless those services are intended.

4.4. Root Account

The root account is the most privileged account on a UNIX system. When the administrator forgot to logout from the system root prompt before leaving the system then the system should automatically logout from the shell. A special variable in Linux, 'TMOUT', must be set in `/etc/profile` to use the feature.

Edit the `/etc/profile` file:

```
# vi /etc/profile
```

Add the following lines:

```
"HISTFILESIZE="
"TMOUT=3600"
```

The value entered for the variable "TMOUT=" is in second and represent 1 hours (60 * 60 = 3600 seconds). By adding the line in */etc/profile*, automatic logout after one hour of inactivity will apply for all users on the system. Setting this variable in user's individual ".bashrc " file will automatically logout them after a certain time.

After this parameter has been set on the system, re-login is required for the changes to take effect.

4.5. Services and ports

Services are background programs that serve as a utility function without being called by a user. This utility may range from maintenance utility or to provide an interface upon request. Most of these services are not useful depending on the Linux usage purposes. On the other hand, ports are designated to provide a gateway to the services. These ports can be numbered from 1 to 65535.

Commands such as *netstat* and *ps* can provide information on running services and ports. Example, *netstat -taun* will provide information on running network services and the ports it listen to.

The most informative command, *ps*, can be issued by '*ps aux*' without quotes, will provide information on every running services on the system.

After determining which services should be disabled based on the information, the command *service* can be used to stop those services.

Example, to stop sendmail:

```
# service sendmail stop
```

And the command below can be issued to stop the service completely upon startup.

```
# chkconfig sendmail off
```

These general services are recommended to be disabled:

<i>apmd</i>	Required only in laptops to monitor battery information
<i>portmap</i>	Only if rpc services is running (which is dangerous) i.e NFS, NIS
<i>pcmcia</i>	Required only in laptops
<i>telnet</i>	Use Secure Shell (SSH)
<i>finger</i>	Used to query account information
<i>samba</i>	Used to share volumes with Windows clients
<i>sendmail</i>	Mail server, depends on purpose
<i>httpd</i>	Apache web server, depends on purpose
<i>mysql</i>	Database server
<i>vnc</i>	Remote desktop administration
<i>nfs</i>	Network File Server
<i>xfs</i>	X Font server

Table 2: Disable unwanted services

4.6. Securing xinetd

Xinetd is a secure replacement for **inetd** and it also known as the internet service daemon. It is started on bootup and listens on all service ports for the services listed in its configuration file, `/etc/xinetd.conf`. When connection request is made, xinetd starts up the corresponding network service. Xinetd is available in newer versions of Linux e.g. Redhat 7.x and later. For older versions of Linux refer to inetd instead of xinetd.

- Ensure xinetd configuration is own by root

```
[root@asydz etc]# ls -l xinetd.conf
-rw-r--r-- 1 root root 289 Feb 18 02:59 xinetd.conf
```

- Change the permissions on `/etc/xinetd.conf` file to 600, so that only root can read or write to it

```
# chmod 600 /etc/xinetd.conf
```

- Set `/etc/xinetd.conf` file immutable, using the **chattr** command so that nobody can modify that file

```
# chattr +i /etc/xinetd.conf
```

This will prevent any changes (accidental or otherwise) to the "xinetd.conf" file. The only person that can set or clear this attribute is the super-user root. To modify back the xinetd.conf file, unset the immutable flag.

```
# chattr -i /etc/xinetd.conf
```

4.7. Securing "/etc/services" file

Securing the "/etc/services" file prevents unauthorized deletion or addition of services. This involves in adding an immutable bit to the file. To secure the "/etc/services" file, use the command:

```
# chattr +i /etc/services
```

Securing root programs

Ensure **/sbin** and **/etc** folders are owned by root. By default, normal users can reboot the system by issuing 'reboot' command or by pressing Ctrl-Alt-Del combo keys.

To disable the reboot command to users, ensure **/sbin/halt** is owned by root:

```
# chmod 700 /sbin/halt
```

To disable Ctrl-Alt-Del, edit **/etc/inittab** :

```
# vi /etc/inittab
```

Add a comment to the line stating, **ca::ctrlaltdel:/sbin/shutdown -t3 -r now**, so it reads

```
# ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

After making changes issue the command to take effect :

```
# /sbin/init q
```

By commenting out the line, restarting using Ctrl-Alt-Del is useless even to root. To shutdown, login as root and use the proper shutdown command :

```
# /sbin/shutdown -r now
```

Replace 'r' with 'h' for powering off the system.

4.8. Disallow root login from different consoles

The `/etc/securetty` file specifies which TTY devices the “root” user is allowed to log in. Edit the `/etc/securetty` file to disable any tty that are not needed by commenting them out (# at the beginning of the line).

4.9. Blocking su to root

The `su` (Substitute User) command allows a user to become other existing users on the system. To prevent users from `su` to root or restrict `su` command to certain users then add the following two lines to the top of `su` configuration in the `/etc/pam.d` directory.

Edit the `su` file (`vi /etc/pam.d/su`) and add the following two lines to the top of the file:

```
auth sufficient /lib/security/pam_rootok.so debug
auth required /lib/security/Pam_wheel.so group=wheel
```

This example provides that only members of the ‘wheel’ group can `su` to root, which also includes logging.

4.10. TCP Wrappers

TCP wrapper is used to provide additional security against intrusion by controlling connections to defined services. TCP wrappers are controlled from two files.

- `/etc/hosts.allow`
- `/etc/hosts.deny`

The best policy is to deny all hosts by putting "ALL: ALL@ALL, PARANOID" in the `/etc/hosts.deny` file and then explicitly list trusted hosts who are allowed to connect to the machine in the `/etc/hosts.allow` file.

Edit the `hosts.deny` file (`vi /etc/hosts.deny`) and add the following lines:

```
# Deny access to everyone.
```

```
ALL: ALL@ALL, PARANOID
```

Which means all services, all locations is blocked, unless they are permitted access by entries in the allow file.

Edit the `hosts.allow` file (`vi /etc/hosts.allow`) and add for example, the following line:

As an example:

```
ftp : blahtest.com
```

In the above example, ftp service is only allowed to any domain from blahtest.com. However, advance filtering can be achieved using a built-in utility called iptables.

4.11. IPTables

IPTables provide customization of rules depending on the user needs. Here are some recommended IPTables configurations. First general rule is to block everything, and from there rules are added accordingly. An allowed rule, ACCEPT, will bypass a blocking rule, e.g DROP, REJECT.

IPTables consists of chains that control the packet flow. These chains are INPUT, OUTPUT and FORWARD. Here are some basic configurations:

Rules should be cleared from the beginning.

```
# iptables -F; iptables -t nat -F; iptables -t mangle -F
```

To deny everything:

```
# iptables -A INPUT -j DROP  
# iptables -A OUTPUT -j DROP  
# iptables -A FORWARD -j DROP
```

These sample rules make a secure connection by enabling inspection against flowing packets. Only packets with established sessions are allowed through. 'eth0' is the interface number of a network card, changes should be applied accordingly.

```
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  
# iptables -A INPUT -m state --state NEW -i ! eth0 -j ACCEPT  
# iptables -P INPUT DROP  
# iptables -A FORWARD -i eth0 -o eth0 -j REJECT
```

The use of IPTables is beyond the scope of this guideline and further reading on the topic is recommended.

4.12. Detecting SUID/SGID programs

A regular user will be able to run a program as root if it is set to SUID root. A system administrator should minimize the use of these SUID/SGID programs and disable the programs which are not needed.

To find all SUID file with a 's' bit;

```
# find / -type f \( -perm -04000 -o -perm -02000 \) \! -exec ls -lg {} \;
```

This will produce an output listing all SUID files. Adding '> suidtxt' to the end of the command will output the results to a file called suidtxt.

The following command will disable any SUID file:

```
# chmod a-s <filename>
```

4.13. Hiding System Information

In a default Linux environment, login screen will show important information such as the Linux distribution name, version and kernel information. With this information, potential attacker might have the information he/she need to focus their attack to a specific version or name.

By following these following steps will disable the information and will only show 'login:' at the login menu.

Edit */etc/rc.d/rc.local* and put # to comment out the following lines:

```
# This will overwrite /etc/issue at every boot. So, make any changes you  
# want to make to /etc/issue here or you will lose them when you reboot.  
#echo "" > /etc/issue  
#echo "$R" >> /etc/issue  
#echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue  
#  
#cp -f /etc/issue /etc/issue.net  
#echo >> /etc/issue
```

The system banner information is only applicable for local logins or remote telnet logins. It is a good practice that the use of telnet is replace by ssh (secure shell) which is more secure and provide encryption.

4.14. Other Utilities

Built-in utilities in Linux might not be able to prevent or detect malicious activities fully. There are several tools that add more security enhancement to the system. Such tools provide configuration interface, system files integrity check, intrusion detection system and more. Below are some useful third party programs for Linux.

4.15. Tripwire

Tripwire is a policy driven file system integrity checking tool that allows system administrators to verify the integrity of their data. The program monitors key attributes of files that should not change, including binary signature, size, expected change of size, etc.

Tripwire is originally known as an intrusion detection tool, but can be used for many other purposes such as integrity assurance, change management, policy compliance and more.

<http://www.tripwire.org>

<http://sourceforge.net/projects/tripwire/>

4.16. Sentry Tools

The Sentry tools provide host-level security services for the UNIX platform. PortSentry, Logcheck/LogSentry, and HostSentry protect against portscans, automate log file auditing, and detect suspicious login activity on a continuous basis.

Portsentry provides detection against scanning and response to it. It runs as a daemon and blocks scanning addresses from connecting to the system.

<http://sourceforge.net/projects/sentrytools/>

4.17. Bastille

Bastille is a useful tool that attempts to "harden" or "tighten" UNIX operating systems, by configuring daemons, system settings and firewall. It currently supports the Red Hat, Debian, Mandrake, SuSE and TurboLinux Linux distributions along with HP-UX and Mac OS X.

<http://www.bastille-linux.org>

5. Conclusion

Linux has become a major operating system used worldwide both for workstations and servers. Practicing good security measures prevents undesirable events that may jeopardize the organization.

This paper has provided basic instructions and guides to secure the Linux system. Also note that, this guideline is generic and based on Redhat 9.0. Different setups may require different approach in hardening the system. Following these steps provide improved security on default Linux installation. Users are recommended to harden their system further based on their system configuration.

Even applying latest security patches or updates, there is no 100% security. The best way is to apply security techniques and methodology to make it as difficult as possible for the system from being compromised.

References

Koutras, Chris. "The process of hardening Linux". Sans Institute, January 2001.
http://www.giac.org/practical/gsec/Chris_Koutras_GSEC.pdf

Wirzenius, Lars, Oja, Joanna, Stafford, Stephen, Weeks, Alex. "The Linux System Administrator's Guide". Version 8.0. 03 December 2003.
<http://www.tldp.org/LDP/sag/html/index.html>

Burgiss, Hal. "Security Quick-Start HOWTO for Linux". Version 1.2. 21 July 2002
<http://www.linux.org/docs/ldp/howto/Security-Quickstart-HOWTO>