# 20 Linux Server Hardening Security Tips

*by* VIVEK GITE *on* OCTOBER 30, 2009 *LAST UPDATED* JANUARY 31, 2016
*in* DEBIAN LINUX, FEDORA LINUX, GENTOO LINUX, GNU/OPEN SOURCE, HOWTO, LINUX, MONITORING, NETWORKING, PACKAGE MANAGEMENT, REDHAT/FEDORA LINUX, SECURITY, SUSE LINUX, SYS ADMIN, TIPS, UBUNTU LINUX

Securing your Linux server is important to protect your data, intellectual property, and time, from the hands of crackers (hackers). The system administrator is responsible for security Linux box. In this first part of a Linux server security series, I will provide 20 hardening tips for default installation of Linux system.

## Linux Server Hardening Checklist and Tips

The following instructions assume that you are using CentOS/RHEL or Ubuntu/Debian based Linux distribution.

## #1: Encrypt Data Communication

All data transmitted over a network is open to monitoring. **Encrypt transmitted data whenever possible** with password or using keys / certificates.

1. Use scp, ssh, rsync, or sftp for file transfer. You can also mount remote server file system or your own home directory using special sshfs and fuse tools.
2. GnuPG allows to encrypt and sign your data and communication, features a versatile key managment system as well as access modules for all kind of public key directories.
3. Fugu is a graphical frontend to the commandline Secure File Transfer application (SFTP). SFTP is similar to FTP, but unlike FTP, the entire session is encrypted, meaning no passwords are sent in cleartext form, and is thus much less vulnerable to third-party interception. Another option is FileZilla – a cross-platform client that supports FTP, FTP over SSL/TLS (FTPS), and SSH File Transfer Protocol (SFTP).

4. OpenVPN is a cost-effective, lightweight SSL VPN.
5. Lighttpd SSL (Secure Server Layer) Https Configuration And Installation
6. Apache SSL (Secure Server Layer) Https (mod_ssl) Configuration And Installation

**#1.1: Avoid Using FTP, Telnet, And Rlogin / Rsh Services**
Under most network configurations, user names, passwords, FTP / telnet / rsh commands and transferred files can be captured by anyone on the same network using a packet sniffer. The common solution to this problem is to use either OpenSSH , SFTP, or FTPS(FTP over SSL), which adds SSL or TLS encryption to FTP. Type the following command to delete NIS, rsh and other outdated service:

```
# yum erase inetd xinetd ypserv tftp-server telnet-server rsh-serve
```

## #2: Minimize Software to Minimize Vulnerability

Do you really need all sort of web services installed? Avoid installing unnecessary software to avoid vulnerabilities in software. Use the RPM package manager such as yum or apt-get and/or dpkg to review all installed set of software packages on a system. Delete all unwanted packages.

```
# yum list installed
```

```
# yum list packageName
```

```
# yum remove packageName
```

OR

```
# dpkg --list
```

```
# dpkg --info packageName
```

```
# apt-get remove packageName
```

## #3: One Network Service Per System or VM Instance

Run different network services on separate servers or VM instance. This limits the number of other services that can be compromised. For example, if an attacker able to successfully exploit a software such as Apache flow, he / she will get an access to entire server including other services such as MySQL, e-mail server and so on. See how to install Virtualization software:

- Install and Setup XEN Virtualization Software on CentOS Linux 5
- How To Setup OpenVZ under RHEL / CentOS Linux

## #4: Keep Linux Kernel and Software Up to Date

Applying security patches is an important part of maintaining Linux server. Linux provides all necessary tools to keep your system updated, and also allows for easy upgrades between versions. All security update should be reviewed and applied as soon as possible. Again, use the RPM package manager such as yum and/or apt-get and/or dpkg to apply all security updates.

```
# yum update
```

OR

```
# apt-get update && apt-get upgrade
```

You can configure Red hat / CentOS / Fedora Linux to send yum package update notification via email. Another option is to apply all security updates via a cron job. Under Debian / Ubuntu Linux you can use apticron to send security notifications.

## #5: Use Linux Security Extensions

Linux comes with various security patches which can be used to guard against misconfigured or compromised programs. If possible use SELinux and other Linux securityextensions to enforce limitations on network and other programs. For example, SELinux provides a variety of security policies for Linux kernel.

### #5.1: SELinux

I strongly recommend using SELinux which provides a flexible Mandatory Access Control (MAC). Under standard Linux Discretionary Access Control (DAC), an application or process running as a user (UID or SUID) has the user's permissions to objects such as files, sockets, and other processes. Running a MAC kernel protects the system from malicious or flawed applications that can damage or destroy the system. See the official Redhat documentation which explains SELinux configuration.

## #6: User Accounts and Strong Password Policy

Use the useradd / usermod commands to create and maintain user accounts. Make sure you have a good and strong password policy. For example, a good password includes at least 8 characters long and mixture of alphabets, number, special character, upper & lower alphabets etc. Most important pick a password you can remember. Use tools such as "John the ripper" to find out weak users passwords on your server. Configure pam_cracklib.so to enforce the password policy.

### #6.1: Password Aging

The chage command changes the number of days between password changes and the date of the last password change. This information is used by the system to determine when a user must change his/her password. The /etc/login.defs file defines the site-specific configuration for the shadow password suite including password aging configuration. To disable password aging, enter:

```
chage -M 99999 userName
```

To get password expiration information, enter:

```
chage -l userName
```

Finally, you can also edit the /etc/shadow file in the following fields:

```
{userName}:{password}:{lastpasswdchanged}:{Minimum_days}:{Maximum_days}:{Warn}:{Inact
ive}:{Expire}:
```

Where,

1. **Minimum_days**: The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password.
2. **Maximum_days**: The maximum number of days the password is valid (after that user is forced to change his/her password).
3. **Warn** : The number of days before password is to expire that user is warned that his/her password must be changed.
4. **Expire** : Days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

I recommend chage command instead of editing the /etc/shadow by hand:

```
# chage -M 60 -m 7 -W 7 userName
```

Recommend readings:

- Linux: Force Users To Change Their Passwords Upon First Login
- Linux turn On / Off password expiration / aging
- Lock the user password

- Search for all account without password and lock them
- Use Linux groups to enhance security

### #6.2: Restricting Use of Previous Passwords

You can prevent all users from using or reuse same old passwords under Linux. The pam_unix module parameter remember can be used to configure the number of previous passwords that cannot be reused.

### #6.3: Locking User Accounts After Login Failures

Under Linux you can use the faillog command to display faillog records or to set login failure limits. faillog formats the contents of the failure log from /var/log/faillog database / log file. It also can be used for maintains failure counters and limits.To see failed login attempts, enter:

```
faillog
```

To unlock an account after login failures, run:

```
faillog -r -u userName
```

Note you can use passwd command to lock and unlock accounts:

```
# lock account
```

```
passwd -l userName
```

```
# unlocak account
```

```
passwd -u userName
```

### #6.4: How Do I Verify No Accounts Have Empty Passwords?

Type the following command

```
# awk -F: '($2 == "") {print}' /etc/shadow
```

Lock all empty password accounts:

```
# passwd -l accountName
```

### #6.5: Make Sure No Non-Root Accounts Have UID Set To 0

Only root account have UID 0 with full permissions to access the system. Type the following command to display all accounts with UID set to 0:

```
# awk -F: '($3 == "0") {print}' /etc/passwd
```

You should only see one line as follows:

```
root:x:0:0:root:/root:/bin/bash
```

If you see other lines, delete them or make sure other accounts are authorized by you to use UID 0.

## #7: Disable root Login

Never ever login as root user. You should use sudo to execute root level commands as and when required. sudo does greatly enhances the security of the system without sharing root password with other users and admins. sudo provides simple auditing and trackingfeatures too.

## #8: Physical Server Security

You must protect Linux servers physical console access. Configure the BIOS and disable the booting from external devices such as DVDs / CDs / USB pen. Set BIOS and grub boot loader password to protect these settings. All production boxes must be locked in IDCs (Internet Data Center) and all persons must pass some sort of security checks before accessing your server. See also:

- 9 Tips To Protect Linux Servers Physical Console Access.

## #9: Disable Unwanted Services

Disable all unnecessary services and daemons (services that runs in the background). You need to remove all unwanted services from the system start-up. Type the following command to list all services which are started at boot time in run level # 3:

```
# chkconfig --list | grep '3:on'
```

To disable service, enter:

```
# service serviceName stop
```

```
# chkconfig serviceName off
```

### #9.1: Find Listening Network Ports

Use the following command to list all open ports and associated programs:

```
netstat -tulpn
```

OR

```
nmap -sT -O localhost
```

```
nmap -sT -O server.example.com
```

Use iptables to close open ports or stop all unwanted network services using above service and chkconfig commands.

#9.2: See Also

- update-rc.d like command on Redhat Enterprise / CentOS Linux.
- Ubuntu / Debian Linux: Services Configuration Tool to Start / Stop System Services.
- Get Detailed Information About Particular IP address Connections Using netstat Command.

## #10: Delete X Windows

X Windows on server is not required. There is no reason to run X Windows on your dedicated mail and Apache web server. You can disable and remove X Windows to improve server security and performance. Edit /etc/inittab and set run level to 3. Finally, remove X Windows system, enter:

```
# yum groupremove "X Window System"
```

## #11: Configure Iptables and TCPWrappers

Iptables is a user space application program that allows you to configure the firewall (Netfilter) provided by the Linux kernel. Use firewall to filter out traffic and allow onlynecessary traffic. Also use the TCPWrappers a host-based networking ACL system to filter network access to Internet. You can prevent many denial of service attacks with the help of Iptables:

- CentOS / Redhat Iptables Firewall Configuration Tutorial
- Lighttpd Traffic Shaping: Throttle Connections Per Single IP (Rate Limit).
- How to: Linux Iptables block common attack.
- psad: Linux Detect And Block Port Scan Attacks In Real Time.
- Use shorewall on CentOS/RHEL or Ubuntu/Debian Linux based server to secure your system.

## #12: Linux Kernel /etc/sysctl.conf Hardening

/etc/sysctl.conf file is used to configure kernel parameters at runtime. Linux reads and applies settings from /etc/sysctl.conf at boot time. Sample /etc/sysctl.conf:

```
# Turn on execshield

kernel.exec-shield=1

kernel.randomize_va_space=1

# Enable IP spoofing protection

net.ipv4.conf.all.rp_filter=1

# Disable IP source routing

net.ipv4.conf.all.accept_source_route=0

# Ignoring broadcasts request

net.ipv4.icmp_echo_ignore_broadcasts=1

net.ipv4.icmp_ignore_bogus_error_messages=1

# Make sure spoofed packets get logged

net.ipv4.conf.all.log_martians = 1
```

## #13: Separate Disk Partitions

Separation of the operating system files from user files may result into a better and secure system. Make sure the following filesystems are mounted on separate partitions:

- /usr
- /home
- /var and /var/tmp
- /tmp

Create separate partitions for Apache and FTP server roots. Edit /etc/fstab file and make sure you add the following configuration options:

1. **noexec** – Do not set execution of any binaries on this partition (prevents execution of binaries but allows scripts).
2. **nodev** – Do not allow character or special devices on this partition (prevents use of device files such as zero, sda etc).
3. **nosuid** – Do not set SUID/SGID access on this partition (prevent the setuid bit).

Sample /etc/fstab entry to to limit user access on /dev/sda5 (ftp server root directory):

```
/dev/sda5  /ftpdata          ext3    defaults,nosuid,nodev,noexec 1 2
```

### #13.1: Disk Quotas

Make sure disk quota is enabled for all users. To implement disk quotas, use the following steps:

1. Enable quotas per file system by modifying the /etc/fstab file.
2. Remount the file system(s).
3. Create the quota database files and generate the disk usage table.
4. Assign quota policies.
5. See implementing disk quotas tutorial for further details.

## #14: Turn Off IPv6

Internet Protocol version 6 (IPv6) provides a new Internet layer of the TCP/IP protocol suite that replaces Internet Protocol version 4 (IPv4) and provides many benefits. Currently there are no good tools out which are able to check a system over network for IPv6 security issues. Most Linux distro began enabling IPv6 protocol by default. Crackers can send bad traffic via IPv6 as most admins are not monitoring it. Unless network configuration requires it, disable IPv6 or configure Linux IPv6 firewall:

- RedHat / Centos Disable IPv6 Networking.
- Debian / Ubuntu And Other Linux Distros Disable IPv6 Networking.
- Linux IPv6 Howto – Chapter 19. Security.
- Linux IPv6 Firewall configuration and scripts are available here.

## #15: Disable Unwanted SUID and SGID Binaries

All SUID/SGID bits enabled file can be misused when the SUID/SGID executable has a security problem or bug. All local or remote user can use such file. It is a good idea to find all such files. Use the find command as follows:

```
#See all set user id files:
```

```
find / -perm +4000
```

```
# See all group id files
```

```
find / -perm +2000
```

```
# Or combine both in a single command
```

```
find / \( -perm -4000 -o -perm -2000 \) -print
```

```
find / -path -prune -o -type f -perm +6000 -ls
```

You need to investigate each reported file. See reported file man page for further details.

### #15.1: World-Writable Files

Anyone can modify world-writable file resulting into a security issue. Use the following command to find all world writable and sticky bits set files:

```
find /dir -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print
```

You need to investigate each reported file and either set correct user and group permission or remove it.

### #15.2: Noowner Files

Files not owned by any user or group can pose a security problem. Just find them with the following command which do not belong to a valid user and a valid group

```
find /dir -xdev \( -nouser -o -nogroup \) -print
```

You need to investigate each reported file and either assign it to an appropriate user and group or remove it.

## #16: Use A Centralized Authentication Service

Without a centralized authentication system, user auth data becomes inconsistent, which may lead into out-of-date credentials and forgotten accounts which should have been deleted in first place. A centralized authentication service allows you maintaining central control over Linux / UNIX account and authentication data. You can keep auth data synchronized between servers. Do not use the NIS service for centralized authentication. Use OpenLDAP for clients and servers.

### #16.1: Kerberos

Kerberos performs authentication as a trusted third party authentication service by using cryptographic shared secret under the assumption that packets traveling along the insecure network can be read, modified, and inserted. Kerberos builds on symmetric-key cryptography and requires a key distribution center. You can make remote login, remote copy, secure inter-system file copying and other high-risk tasks safer and more controllable using Kerberos. So, when users authenticate to network services using Kerberos, unauthorized users attempting to gather passwords by monitoring network traffic are effectively thwarted. See how to setup and use Kerberos.

## #17: Logging and Auditing

You need to configure logging and auditing to collect all hacking and cracking attempts. By default syslog stores data in /var/log/ directory. This is also useful to find out software misconfiguration which may open your system to various attacks. See the following logging related articles:

1. Linux log file locations.
2. How to send logs to a remote loghost.
3. How do I rotate log files?.
4. man pages syslogd, syslog.conf and logrotate.

### #17.1: Monitor Suspicious Log Messages With Logwatch / Logcheck

Read your logs using logwatch or logcheck. These tools make your log reading life easier. You get detailed reporting on **unusual items** in syslog via email. A sample syslog report:

```
 ################### Logwatch 7.3 (03/24/06) ####################


        Processing Initiated: Fri Oct 30 04:02:03 2009


        Date Range Processed: yesterday

```

```
                              ( 2009-Oct-29 )

                              Period is day.

      Detail Level of Output: 0

              Type of Output: unformatted

           Logfiles for Host: www-52.nixcraft.net.in

  ##################################################################



 --------------------- Named Begin ------------------------



 **Unmatched Entries**

    general: info: zone XXXXXX.com/IN: Transfer started.: 3 Time(s)

    general: info: zone XXXXXX.com/IN: refresh: retry limit for master
ttttttttttttttttttt#53 exceeded (source ::#0): 3 Time(s)

    general: info: zone XXXXXX.com/IN: Transfer started.: 4 Time(s)

    general: info: zone XXXXXX.com/IN: refresh: retry limit for master
ttttttttttttttttttt#53 exceeded (source ::#0): 4 Time(s)



 ---------------------- Named End -------------------------



  --------------------- iptables firewall Begin ------------------------



 Logged 87 packets on interface eth0
```

```
  From 58.y.xxx.ww - 1 packet to tcp(8080)

  From 59.www.zzz.yyy - 1 packet to tcp(22)

  From 60.32.nnn.yyy - 2 packets to tcp(45633)

  From 222.xxx.ttt.zz - 5 packets to tcp(8000,8080,8800)


---------------------- iptables firewall End -------------------------


--------------------- SSHD Begin ------------------------


Users logging in through sshd:

   root:

      123.xxx.ttt.zzz: 6 times


---------------------- SSHD End -------------------------


--------------------- Disk Space Begin ------------------------


Filesystem            Size  Used Avail Use% Mounted on

/dev/sda3             450G  185G  241G  44% /

/dev/sda1              99M   35M   60M  37% /boot
```

```
---------------------- Disk Space End -------------------------



###################### Logwatch End #########################
```

(Note output is truncated)

**#17.2: System Accounting with auditd**

The auditd is provided for system auditing. It is responsible for writing audit records to the disk. During startup, the rules in /etc/audit.rules are read by this daemon. You can open /etc/audit.rules file and make changes such as setup audit file log location and other option. With auditd you can answers the following questions:

1. System startup and shutdown events (reboot / halt).
2. Date and time of the event.
3. User respoisble for the event (such as trying to access /path/to/topsecret.dat file).
4. Type of event (edit, access, delete, write, update file & commands).
5. Success or failure of the event.
6. Records events that Modify date and time.
7. Find out who made changes to modify the system's network settings.
8. Record events that modify user/group information.
9. See who made changes to a file etc.

See our quick tutorial which explains enabling and using the auditd service.

## #18: Secure OpenSSH Server

The SSH protocol is recommended for remote login and remote file transfer. However, ssh is open to many attacks. See how to secure OpenSSH server:

- Top 20 OpenSSH Server Best Security Practices.

## #19: Install And Use Intrusion Detection System

A network intrusion detection system (NIDS) is an intrusion detection system that tries to detect malicious activity such as denial of service attacks, port scans or even attempts to crack into computers by monitoring network traffic.

It is a good practice to deploy any integrity checking software before system goes online in a production environment. If possible install AIDE software before the system is connected to any network. AIDE is a host-based intrusion detection system (HIDS) it can monitor and analyses the internals of a computing system.
Snort is a software for intrusion detection which is capable of performing packet logging and real-time traffic analysis on IP networks.

## #20: Protecting Files, Directories and Email

Linux offers excellent protections against unauthorized data access. File permissions and MAC prevent unauthorized access from accessing data. However, permissions set by the Linux are irrelevant if an attacker has physical access to a computer and can simply move the computer's hard drive to another system to copy and analyze the sensitive data. You can easily protect files, and partitons under Linux using the following tools:

- To encrypt and decrypt files with a password, use gpg command.
- Linux or UNIX password protect files with openssl and other tools.
- See how to encrypting directories with ecryptfs.
- TrueCrypt is free open-source disk encryption software for Windows 7/Vista/XP, Mac OS X and Linux.
- Howto: Disk and partition encryption in Linux for mobile devices.
- How to setup encrypted Swap on Linux.

### #20.1: Securing Email Servers

You can use SSL certificates and gpg keys to secure email communication on both server and client computers:

- Linux Securing Dovecot IMAPS / POP3S Server with SSL Configuration.
- Linux Postfix SMTP (Mail Server) SSL Certificate Installations and Configuration.
- Courier IMAP SSL Server Certificate Installtion and Configuration.
- Configure Sendmail SSL encryption for sending and receiving email.
- Enigmail: Encrypted mail with Mozilla thunderbird.

## Other Recommendation:

- Backups – It cannot be stressed enough how important it is to make a backup of your Linux system. A proper offsite backup allows you to recover from cracked server i.e. an intrusion. The traditional UNIX backup programs are dump and restore are also recommended.
- How to: Looking for Rootkits on Linux based server.
- Howto: Enable ExecShield Buffer Overflows Protection on Linux based server.
- Subscribe to Redhat or Debian Linux security mailing list or RSS feed.

Recommend readings:

1. Red Hat Enterprise Linux – Security Guide.
2. Linux security cookbook– A good collections of security recipes for new Linux admin.
3. Snort 2.1 Intrusion Detection, Second Edition – Good introduction to Snort and Intrusion detection under Linux.
4. Hardening Linux – Hardening Linux identifies many of the risks of running Linux hosts and applications and provides practical examples and methods to minimize those risks.
5. Linux Security HOWTO.

In the next part of this series I will discuss