To search, type & hit enter...

40 Linux Server Hardening Security Tips [2024 edition]

Author: Vivek Gite • Last updated: March 13, 2024 • 157 comments

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 of the Linux box. In this first part of a Linux server security series, I will provide 40 Linux server hardening tips for default installation of Linux system.

Linux Server Hardening Security Tips and Checklist

Космос довжиною в життя
Почни життя на самотній планеті
Xcraft 21+

1. Encrypt Data Communication For Linux Server

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 management system as well as access modules for all kind of public key directories.
3. OpenVPN is a cost-effective, lightweight SSL VPN. Another option is to try out tinc that uses tunneling and encryption to create a secure private network between hosts on the Internet or private insecure LAN.
4. Lighttpd SSL (Secure Server Layer) Https Configuration And Installation
5. Apache SSL (Secure Server Layer) Https (mod_ssl) Configuration And Installation
6. How to configure Nginx with free Let’s Encrypt SSL certificate on Debian or Ubuntu Linux

2. Avoid Using FTP, Telnet, And Rlogin / Rsh Services on Linux

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 Type the following yum command to delete NIS, rsh and other outdated service:

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

If you are using a Debian/Ubuntu Linux based server, try apt-get command/apt command to remove insecure services:

```
$ sudo apt-get --purge remove xinetd nis yp-tools tftpd atftpd
tftpd-hpa telnetd rsh-server rsh-redone-server
```

3. Minimize Software to Minimize Vulnerability in Linux

Do you really need all sort of web services installed? Avoid installing unnecessary software to avoid vulnerabilities in software. Use the RPM package

```
# yum list packageName
# yum remove packageName
```

OR

```
# dpkg --list
# dpkg --info packageName
# apt-get remove packageName
```

4. 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 or she will get an access to entire server including other services such as MySQL/MariaDB/PGSql, e-mail server and so on. See how to install Virtualization software for more info:

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

5. 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. It is also possible to configure unattended upgrades for your Debian/Ubuntu Linux server using apt-get command/apt command:

```
$ sudo apt-get install unattended-upgrades apt-listchanges
bsd-mailx
```

misconfigured or compromised programs. If possible use SELinux and other Linux security extensions to enforce limitations on network and other programs. For example, SELinux provides a variety of security policies for Linux kernel.

7. 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.

8. Linux 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.

9. Set Up Password Aging For Linux Users For Better Security

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:

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 file 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

10. Restricting Use of Previous Passwords on Linux

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.

11. 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:

```
# lock Linux account
passwd -l userName
# unlock Linux account
passwd -u userName
```

12. 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
```

13. 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.

14. 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 tracking features too.

15. Physical Server Security

some sort of security checks before accessing your server. See also:

- 9 Tips To Protect Linux Servers Physical Console Access.

16. Disable Unwanted Linux 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
```

A note about systemd based Linux distro and services

Modern Linux distros with systemd use the systemctl command for the same purpose.

Print a list of services that lists which runlevels each is configured on or off

```
# systemctl list-unit-files --type=service
# systemctl list-dependencies graphical.target
```

Turn off service at boot time

```
# systemctl disable service
# systemctl disable httpd.service
```

Start/stop/restart service

```
# systemctl disable service
# systemctl disable httpd.service
```

Get status of service

```
# systemctl status service
# journalctl
# journalctl -u network.service
# journalctl -u ssh.service
# journalctl -f
# journalctl -k
```

17. Find Listening Network Ports

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

```
netstat -tulpn
```

OR use the ss command as follows:

```
$ ss -tulpn
```

OR

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

- Top 32 Nmap Command Examples For Sys/Network Admins for more info. Use iptables to close open ports or stop all unwanted network services using above service and chkconfig commands.
- 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.

18. Delete X Window Systems (X11)

X Window systems on server is not required. There is no reason to run X11 on your dedicated Linux based mail and Apache/Nginx 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"
```

On CentOS 7/RHEL 7 server use the following commands:

```
# yum group remove "GNOME Desktop"
# yum group remove "KDE Plasma Workspaces"
```

19. Configure Iptables and TCPWrappers based Firewall on Linux

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 only necessary 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:

- How to setup a UFW firewall on Ubuntu 16.04 LTS server
- How to set up a firewall using FirewallD on RHEL 8
- Linux: 20 Iptables Examples For New SysAdmins
- 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.

Read all our iptables command and ufw command help pages.

20: 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
```

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
```

22. 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.

23. Turn Off IPv6 only if you are NOT using it on Linux

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

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

24. 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.

25: World-Writable Files on Linux Server

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.

26. 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.

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.

28. 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.

29. 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.

30. Monitor Suspicious Log Messages With Logwatch / Logcheck

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

```
                             ( 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(
   general: info: zone XXXXXX.com/IN: refresh: retry limit for m
   general: info: zone XXXXXX.com/IN: Transfer started.: 4 Time(
   general: info: zone XXXXXX.com/IN: refresh: retry limit for m

---------------------- 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 #########################
```

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.

32. 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.
- Secure your Linux desktop and SSH login using two factor Google authenticator.

33. 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

34. Disable USB/firewire/thunderbolt devices

Type the following command to disable USB devices on Linux system:

```
# echo 'install usb-storage /bin/true' >>
/etc/modprobe.d/disable-usb-storage.conf
```

You can use same method to disable firewire and thunderbolt modules:

```
# echo "blacklist firewire-core" >>
/etc/modprobe.d/firewire.conf
# echo "blacklist thunderbolt" >>
/etc/modprobe.d/thunderbolt.conf
```

Once done, users can not quickly copy sensitive data to USB devices or install malware/viruses or backdoor on your Linux based system.

35. Disable unused services

You can disable unused services using the service command/systemctl command:

```
$ sudo systemctl stop service
$ sudo systemctl disable service
```

For example, if you are not going to use Nginx service for some time disable it:

```
$ sudo systemctl stop nginx
$ sudo systemctl disable nginx
```

36. Use fail2ban/denyhost as IDS (Install an Intrusion Detection System)

Fail2ban or denyhost scans the log files for too many failed login attempts and blocks the IP address which is showing malicious signs. See how to install and use denyhost for Linux. One can install fail2ban easily:

```
$ sudo apt-get install fail2ban
```

OR

```
$ sudo yum install fail2ban
```

Restart the service:

```
$ sudo systemctl restart fail2ban.service
```

- Debian / Ubuntu Linux Install Advanced Intrusion Detection Environment (AIDE) Software
- psad: Linux Detect And Block Port Scan Attacks In Real Time

37. Secure Apache/PHP/Nginx server

Edit httpd.conf file and add the following:

```
ServerTokens Prod
ServerSignature Off
TraceEnable Off
Options all -Indexes
Header always unset X-Powered-By
```

Restart the httpd/apache2 server on Linux, run:

```
$ sudo systemctl restart apache2.service
```

OR

```
$ sudo systemctl restart httpd.service
```

You must install and enable mod_security on RHEL/CentOS server. It is recommended that you edit php.ini and secure it too.

- Top 25 Nginx Web Server Best Security Practices
- How to analyze Nginx configuration files for security misconfiguration on Linux or Unix

38. 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

- Linux or UNIX password protect files with openssl and other tools.
- Full disk encryption is a must for securing data, and is supported by most Linux distributions. See how to encrypting harddisk using LUKS on Linux. Make sure swap is also encrypted. Require a password to edit bootloader.
- Make sure root mail is forwarded to an account you check.
- Howto: Disk and partition encryption in Linux for mobile devices.
- 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.

39. 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. You must set up encrypted backups to external storage such as NAS server or FreeNAS server or use cloud computing service such as AWS:

- Debian / Ubuntu Linux Install and Configure Remote Filesystem Snapshot with rsnapshot Incremental Backup Utility
- How To Set Red hat / CentOS Linux Remote Backup / Snapshot Server
- How To Back Up a Web Server
- How To Use rsync Command To Backup Directory Under Linux

40. Other Recommendation and conlcusion

This page explained Linux server hardening security tips. Please see the following pages for more info:

- How to look for Rootkits on Linux based server.
- How to Enable ExecShield Buffer Overflows Protection on Linux based server.
- EUD Security Guidance: Ubuntu 16.04 LTS
- A Guide For Securing RHEL 7
- Basic and advanced config OF SELINUX

Hi! 🤠

I'm Vivek Gite, and I write about Linux, macOS, Unix, IT, programming, infosec, and open source. Subscribe to my RSS feed or email newsletter for updates.

🔎 To search, type & hit enter...

Related Posts

- Book review: Linux Server Security, 2nd Edition
- Coming Soon: Dell Server Preinstalled with Ubuntu…
- Security: Shell script optimization and security tips
- QD#5: Linux Networx Pipes in $37M, Firefox Extending…
- Download of the day: Linux or Solaris Java Standard…
- Install Flash 10 Under Ubuntu Linux 64 bit Edition
- Windows 7 Starter Edition Only Runs 3 Applications At Once
- Linux Advanced Hardening With the Capability Bounding Set

Category	List of Unix and Linux commands
Ansible	Check version • Fedora • FreeBSD • Linux • Ubuntu 18.04 • Ubuntu • macOS
Archiving	z commands
Backup Management	Debian/Ubuntu • FreeBSD • RHEL
Database Server	Backup MySQL server • MariaDB Galera cluster • MariaDB TLS/SSL • MariaDB replication • MySQL Server • MySQL remote access
Download managers	wget
Driver Management	Linux Nvidia driver • lsmod

File Management	cat • cp • less • mkdir • more • tree
Firewall	Alpine Awall • CentOS 8 • OpenSUSE • RHEL 8 • Ubuntu 16.04 • Ubuntu 18.04 • Ubuntu 20.04 • Ubuntu 24.04
KVM Virtualization	CentOS/RHEL 7 • CentOS/RHEL 8 • Debian 9/10/11 • Ubuntu 20.04
Linux Desktop apps	Chrome • Chromium • GIMP • Skype • Spotify • VLC 3
LXD	Backups • CentOS/RHEL • Debian 11 • Fedora • Mount dir • Ubuntu 20.04 • Ubuntu 22.04
Modern utilities	bat • exa
Network Management	Monitoring tools • Network services • RHEL static IP • Restart network interface • nmcli
Network Utilities	NetHogs • dig • host • ip • nmap • ping
OpenVPN	CentOS 7 • CentOS 8 • Debian 10 • Debian 11 • Debian 8/9 • Ubuntu 18.04 • Ubuntu 20.04
Power Management	upower
Package Manager	apk • apt-get • apt • yum
Processes Management	bg • chroot • cron • disown • fg • glances • gtop • iotop • jobs • killall • kill • pidof • pstree • pwdx • time • vtop
Searching	ag • egrep • grep • whereis • which
Shell builtins	compgen • echo • printf
System Management	reboot • shutdown
Terminal/ssh	sshpass • tty
Text processing	cut • rev
Text Editor	6 Text editors • Save and exit vim
User Environment	exit • who
User Information	groups • id • lastcomm • last • lid/libuser-lid • logname • members • users • whoami • w
User Management	/etc/group • /etc/passwd • /etc/shadow • chsh
Web Server	Apache • Let's Encrypt certificate • Lighttpd • Nginx Security • Nginx
WireGuard VPN	Alpine • Amazon Linux • CentOS 8 • Debian 10 • Firewall • Ubuntu 20.04 • qrencode

157 comments… add one ↓

Lego • Oct 30, 2009 @ 10:36

Excellent article! Thanks for posting this.

↩ ∞

veeru • Jan 8, 2012 @ 14:17

sir,
how to configure LDAP server(server side, client side) in UBUNTU linux plese tell me step by step

↩ ∞

Google is your friend. I found the above link in less than 30 seconds. We Linux geeks like to be helpful. Most will tell you how to hunt, but most won't hunt for you, cook for you, and feed you too. 🙂

↩ ∞

Alan • Oct 28, 2014 @ 18:49

Robert, Can you confirm which one of the 2 is best for users authentication? LDAP or Active Directory? Let me know..
-Alan.

↩ ∞

Unop • Jan 15, 2016 @ 10:07

LDAP is just a data store for users or groups – you usually need Kerberos or something similar to authenticate a user against entities in LDAP. Active directory does both of these in a arguably nicely integrated way – you could have Linux servers/workstations be enrolled into AD but it's a case of your mileage may vary .. typically you'd stand up LDAP, Kerberos, etc services yourself.

↩ ∞

AJ • Jan 19, 2016 @ 17:56

This is a good 3 part series for ldap, kerberos, and nfs to get you started.

↩ ∞

Bishwajit • May 9, 2016 @ 3:16

Hi, can you explain a bit, how the mileage would get affected, i mean symptoms where from i can identify lagging issues. Also if i would configure samba 4 as a domain controller with active directory admin pack installed for a single domain. is it worth it??

↩ ∞

Liju • Oct 30, 2009 @ 11:21

Great article.

Really wroth……

↩ ∞

Data7 • Oct 30, 2009 @ 12:16

Very useful indeed. Thanks a lot!

great post. One for the bookmarks….:)

↩ ∞

Suresh • Oct 30, 2009 @ 13:31

Great one sir! Thanks a lot

↩ ∞

surendra kumar anne • Oct 30, 2009 @ 13:38

Though i am an active user in your forum, i never posted a comment on your blog.. but this post really tempted me to comment.
the post really rocks man.. Most of the things new to me..
Thanks for sharing.

↩ ∞

Ben • Oct 30, 2009 @ 14:31

#10 – Disable X-Windows. I think you meant to say edit /etc/inittab and set to run level 3 not 5.

↩ ∞

Andrew Ensley • Oct 30, 2009 @ 15:03

Great article! Thanks for sharing! Bookmarked and Dugg.

↩ ∞

nixCraft • Oct 30, 2009 @ 15:12

@Ben,

It was a typo on my part. Runlevel 5 is for X and 3 is text based full network mode under CentOS / RHEL / Fedora etc.

↩ ∞

Ivan Nemeth • Oct 30, 2009 @ 16:47

bookmarked immediately, thank you

↩ ∞

Toby • Oct 30, 2009 @ 17:25

I'm not surprised that SSH is #1, but I am a little puzzled that there's no mention of key-only authentication… or denyhosts, if password access is a requirement.

nixCraft • Oct 30, 2009 @ 17:39

Please see (#18 SSH) – a direct link Top 20 OpenSSH Server Best Security Practices.

↩ ∞

Cody F • Nov 11, 2022 @ 13:20

Toby's right. Password ageing only encourages people to write their passwords down and no matter how hard you try and prevent it it's going to happen. Better to use a complex password and make sure it's secure. Granted some might still be enticed to write it down but they're certainly going to if they have to constantly change their password. It's a bad idea to enforce changing passwords. You can change passwords if you fear it's been compromised but forcing the user to repeatedly change it is a very bad idea indeed.

Add in key based login as Toby suggested and it's even better.

There were a few others I saw that are suspect but it's a good list overall.

↩ ∞

JIS • Oct 30, 2009 @ 19:04

I usually don't comment on blogs, but this post deserves it…great article! Thank you for sharing….

↩ ∞

robo • Oct 30, 2009 @ 19:10

this is life saver for sysadmins 😉 thanks for sharing,

↩ ∞

Theodoros Goumenidis • Oct 30, 2009 @ 19:56

Your articles always have something special to read. Thanks for sharing.

↩ ∞

luc_rom • Oct 30, 2009 @ 20:12

Great work as always Vivek.

↩ ∞

dave • Oct 30, 2009 @ 22:05

a password for 30 days. Then the user is forced to learn a new password. After another 30 days they are forced to change but by this time the user is starting to forget the passwords because they are changing and can not reuse an old one.

So, Mr User writes it on a sticky note and puts it where he can read it, right on his monitor.

See where I'm going with this? This will happen time and time again which creates more of a compromise to security and defeats the purpose.

↩ ∞

Adam • Sep 11, 2015 @ 18:19

Actually,

There could just be an amendment to those sections advising admins to hold regular security meetings and actively, physically walk around and check for this sort of thing. If a user gets to keep his/her same password for as long as they want, they are going to use that password on each and every site/mail account/etc they have.

Once the "bad guy" has that password, first name dot last name or first initial dot last name isn't too hard to figure out.

Another note here is to use the AllowUser directive in the sshd_config file.

↩ ∞

Cody F • Nov 11, 2022 @ 13:24

'If a user gets to keep his/her same ..'

Rubbish. Maybe some do but not all. They will do that whether they have password ageing or not! It's unrelated.

Besides that they will start writing passwords down if they're forced to frequently change them. Some people still do that! Password ageing is a terrible idea.

Remember that security and convenience are mutually exclusive – and people do not like to be inconvenienced. The more inconvenient something becomes the more people will try coming up with alternatives (like writing the passwords down) and this breaks security. It does not enhance it.

↩ ∞

neolix • Oct 30, 2009 @ 23:45

d0wnund3r01 • Oct 31, 2009 @ 0:45

This is awesome, thanks for posting this for us newbies.
Sorry for my stupid question in advance:
Q: if I remove Xwindows. can I still VNC and get an Xwindows display ?

↩ ∞

Alfa • Oct 31, 2009 @ 3:48

Sir, how to remove / disable "Linux Single" ?

nice post (articles)….

↩ ∞

P Saint Amour • Oct 31, 2009 @ 3:56

I mention so many times to clients that they should set up and use SELinux in mission critical secure situations and they constantly ignore it.
It should be used without question in installations where you want and need an extremely hardened system.

↩ ∞

Antiks • Oct 31, 2009 @ 4:37

Wait….I thought Linux was secure by default?

↩ ∞

Antiks • Oct 31, 2009 @ 9:11

Wait….I thought Linux was secure by default?
Oops…forgot to say great post! Looking forward to your next one.

↩ ∞

Solaris • Oct 31, 2009 @ 13:54

I don't agree with disabling ipv6. The switch must be done and ipv6 has been pretty well
tested until now, chances that some bad traffic will cause a buffer overflow is very low.

↩ ∞

Someone in time • Oct 31, 2009 @ 16:47

Excelent post…
Thanks for share your knowledge…

I see someone's trying to be smart again. Not so much.
#1.1 Removing xinetd would disable my git:// offering.
#3 Hilarious amount of work that only makes sense if you run a corp with load
#10 Almost impossible with many distros due to interdependencies (dbus-1-glib, anyone!?)
#12 Do not forget to set vm.vdso_enabled=1 (some distros still have it at 2, which is only the compat mode)
#13 And leads to "oops, now your partition is full". Been there done that, threw it out. Only /home remains separate.
#14 PEBKAC is not a justification to turn it off.
#20 Truecrypt is a joke (has its own crypto implemention, its own VFAT implementation, and is limited to VFAT even) when you have dm-crypt at hand which has: a well-tested-and-known crypto impl, can use all the well-tested filesystems Linux offers, etc.

↩ ∞

sreeraj.K.G • Nov 2, 2009 @ 9:55

Hello,
This article great one and very useful for all sysadmins.One again gr8 article.

Thanks
Sreeraj.K.G

↩ ∞

John • Nov 2, 2009 @ 14:09

>#1.1 Removing xinetd would disable my git:// offering.
Use your common sense and keep required services.
>#3 Hilarious amount of work that only makes sense if you run a corp with load
Not really, how hard is to run xen under Linux?
>#10 Almost impossible with many distros due to interdependencies (dbus-1-glib, anyone!?)
Really? You run X windows on all servers? You are just wasting your resources.
>#12 Do not forget to set vm.vdso_enabled=1 (some distros still have it at 2, which is only the compat mode)
I do not see vm.vdso_enabled under CentOS, may be it is part of latest kernel or 3rd party.
>#13 And leads to â€œoops, now your partition is fullâ€�. Been there done that, threw it out. Only /home remains separate.
You need to use LVM2.
#20 Truecrypt is a joke (has its own crypto implemention, its own VFAT implementation, and is limited to VFAT even) when you have dm-crypt at hand which has: a well-tested-and-known crypto impl, can use all the well-tested filesystems Linux offers, etc.

chris j • Nov 2, 2009 @ 14:15

I disagree with the #7 disable root login. I agree that root logins should be disabled for things like ssh, forcing users to login using their credentials. Howerver I think sudo makes a box less secure. If an account gets compromised and they have sudo access for root level work, all the attacker has to do is type sudo whatever and away they go.

With having requiring them to su to root, you're adding defense in depth. They might compromise bob's account, but now they have to work harder to get into root.

I think sudo is great for 1 off commands but as a hardening system it leaves a lot to be desired.

↩ ∞

Unop • Jan 15, 2016 @ 10:24

That's based on a limited understanding of sudoku .. Sudo requires you set it up properly to make security matter while also delegating privileges in a controlled fashion – you don't share your root password amongst all the non-sysadmins who require elevation, do you? if you set sudo up so that users are only allowed to invoke a subset of commands as root then an attacker can't just "sudo" and "away they go" .. for e.g. we have developers who push out changes to code and require services to be restarted – in that case, the only command they can run under sudo is '/sbin/service' (and we do have sudo locked down further so they can only restart specific whitelisted services) – every other use of sudo is prohibited and logged (and the latter is how you monitor attempts).

↩ ∞

TransTux • Nov 2, 2009 @ 15:16

Lots of good information on hardening Linux. How about /etc/security/limits.conf and friends to control other security aspects of the Linux?

↩ ∞

chris j • Nov 2, 2009 @ 15:51

to clarify sudo is great for one off commands on personal computers, but not that great for production servers.

↩ ∞

hideaki • Nov 2, 2009 @ 17:45

↩ ∞

Ricardo • Nov 3, 2009 @ 0:11

Perfect! Congratulations,

Friend, you always give greats articles to all we! Your article, it has been very important to i can build a more secure system!

I am from Brazil, and i am student in the Science Computer! But, your level of knowledge is very high!

Good luck with your site!

Bye!

Ricardo Costa

↩ ∞

Jose • Nov 3, 2009 @ 14:51

Lots of good stuff, Thank you so much!

↩ ∞

Gokul • Nov 3, 2009 @ 19:43

Thank you for your tips 🙂
I made a script to harden server and install all necessary things using all of you good guys advise.

Grateful 🙂

Thanks,
Gokul.

↩ ∞

bcwoods1 • Nov 4, 2009 @ 19:06

I've heard both sides of the root login/su debate. Personally I don't like using sudo. I generally use set up a rather long root password and change it every other month or so. I agree with chris j that it adds another layer especially if you set up ssh etc correctly to disable root logins and such. Of course, I don't run any large servers so my experience most likely isn't as large as some of the posters here.

↩ ∞

>> Not really, how hard is to run xen under Linux?
>For real? Itâ€™s harder than running vmware, vbox, qemu/kvm. Because for a start you need >an appropriate xen kernel.
Oh, come on. With Debian or CentOS you need max 5 minutes to have Dom0 + DomU functional (and you don't even have to know what you are doing, there is a zillion howto's on the web)

↩ ∞

Anshell • Nov 7, 2009 @ 4:53

Great Article!! 🙂

↩ ∞

snappy • Nov 7, 2009 @ 22:40

Most of these tips are pretty much ubiquitous. Secure passwords (e.g. those found outside of hacker dictionaries), and mod_security or something similar for your webserver are truly key. When confronted with a linux/UNIX machine, hackers will first try to penetrate among common username/passwords and scan for vulnerabilities in common web applications. Prevent it before it occurs. If you can, setup public-key auth for all SSH related crap. If you're using lighttpd, look for mod_security like rules.

Anyways, one cannot implement all since each environment is different. Also surprised to not see a file intrusion detection system up. Also, securing your machine isn't enough, you want to keep at least daily backups. If you host your server and become a victim of being hacked. Don't expect it to stop there, they will use your machine as a zombie/bot to attack other machines. The ISP will shut your machine down, and you will have even a difficult time getting back to your data. Make backups frequently and off-site. Data is truly of value, the machine it runs on isn't.

Just my 2c.

↩ ∞

Espen • Nov 9, 2009 @ 11:37

Thanks for a great post.

↩ ∞

Denis • Nov 19, 2009 @ 12:27

OVZkernel RHEL

[root@server etc]# sysctl -p
…..

↩ ∞

nixCraft • Nov 19, 2009 @ 14:46

OVZkernel share kernel with its host and other vps operating systems. So you will not able to use all MIBs or iptables features.

↩ ∞

Stefano • Nov 22, 2009 @ 16:10

As usual, thanks!!

↩ ∞

K.K. • Nov 29, 2009 @ 7:50

It's gr8..
Thx 4 sharing..

↩ ∞

Praful • Dec 11, 2009 @ 5:29

Thanks for sharing tips for linux ……… Thanks Mr. Vivek Gite

↩ ∞

prabaas • Dec 14, 2009 @ 4:14

thanks a lot linux guru …………………..great info…………….thanks guru………….

↩ ∞

tux4fun • Dec 20, 2009 @ 12:40

Thanks for the mass of information!

↩ ∞

Vincent • Jan 15, 2010 @ 10:03

Thanks alot for UBER tips…. Thanks Mr. Vivek, from Nixcraft to Cyberciti you keep them coming.

↩ ∞

Vincent • Jan 15, 2010 @ 10:04

Any tips on FAQs on SNMP. Baby steps please..

↩ ∞

network (that includes vpn access), not from the internet side. No need to eat your brain thinking and thinking about sudo, passwords, blah blah. Ah, btw… automatic updates can only break your working system 🙂 The rest, is just common sense. You can't learn linux only by applying rules you read on a web page… you learn linux after years, and maybe only then.

↩ ∞

Ahmed hassan elzebair • Jan 25, 2010 @ 9:39

I do appreciate the effort that has been done to present this informative topic please do inform me via e-mail regardig such security issues.
Many thanks
Eng. ahmed

↩ ∞

Abdul • Feb 7, 2010 @ 11:35

gr8 job yaar

↩ ∞

mrf • Feb 10, 2010 @ 12:24

wow this is heaven for me he3x 🙂 thx mr vivek

↩ ∞

thyag • Feb 19, 2010 @ 19:05

fantastic work!…maximum info with minimum words…great!!

↩ ∞

cyvan • Feb 26, 2010 @ 6:29

Whatever happened to Bastille Linux. Doesn't seem to be maintained anymore.

↩ ∞

Andre • Apr 25, 2010 @ 22:12

Wow! Awesome website!
Keep the tips coming, I am learning lots of good sys admin here.

↩ ∞

Pradeep Singh • Apr 30, 2010 @ 3:48

Could we have a post here for step by step configuration of LDAP (Centralized Authentication Service). And the usage.

really gud info…..Thanxz to the postings……

↩ ∞

vanni linux • Jun 9, 2010 @ 10:11

Thanks for giving this …
TuxRacer

↩ ∞

Abdullah • Jun 16, 2010 @ 7:21

I would choose to install grsecurity:http://grsecurity.net/download.php linux kernel patch anytime over "SELinux"
because it have much more paranoid-security options that would make SElinux look like a baby toy,

↩ ∞

a_m_y • Jun 17, 2010 @ 14:50

Cool! It will help a lot, especially to novice linux users that will make them look expert, as well as for newbies. Thanks so much!! More power!

↩ ∞

edvard • Jun 21, 2010 @ 15:54

Excellent article, however with the need for IPv6 fast approaching, telling users to disable it is like telling us to bury our heads in the sand.
I've seen this advice all over the internet, and it will very soon be not such a good idea.
I would suggest that instead of telling users to disable IPv6, let's start learning about it, creating tools to deal with it and get our hands dirty using it.

↩ ∞

Dave • Jun 30, 2010 @ 23:05

@Ruben. Even if you only can access SSH from your lan, you should still disable root login. Just login using your own SSH key and become root (su). Also limit the users that can become root (wheel users). So before someone can login root, he (or she) first have to crack two user accounts. But disable root login helps also with the physical security.

About some other points. Passwords should not expire if you enforce strong passwords. The trouble is that users can only remember only so many passwords, so if thay have to change password frequently, they're gonna use the same password at other places.

that shows you the passwords. However, a comprised database is dangerous. If I wanted it to, I could have read a lot of emails and collect even more sensitive data like registration mails from websites that show you your password..

SE-Linux should be a standard installed with every Linux distribution. It makes it a bit harder to exploits bugs in code. That's also valuable on workstations. Most companies only secure the front door. If you break a window, you can go anywhere in the building. Hack a workstation and often you can access everything within the LAN.

IPv6 should be disabled if you don't have an IPv6 IP or services. If you have, you have to secure just like you secure an IPv4 network. I already use IPv6 within every LAN I install. The main router (gateway) has an IPv6 bridge to my data center (which is IPv6 enabled) and from there they can connect to both IPv6 networks or IPv4 networks.

↩ ∞

norg • Jul 20, 2010 @ 10:34

There is a slight wording mistake in #1: Encrypt Data Communication, section 3 "Fugu is a graphical frontend to the commandline Secure File Transfer application (SFTP)". The acronym SFTP is misleading. SFTP is the "SSH file transfer protocol", "Secure FTP" is something very different (http://en.wikipedia.org/wiki/FTP_over_SSH#FTP_over_SSH_.28not_SFTP.29). Secure FTP encrypts only the control channel , the data channel stays unencrypted.

↩ ∞

Charlie Brown • Jul 23, 2010 @ 2:55

SFTP is not the SSH file transfer… Whuuat??

SFTP is a UTILITY that RUNS on SSH…

Two different animals dude.. Authur had it right..

It kills me how many people get their info "facts" from wiki…
Man.. doesn't anyone watch CNN? wiki is poo.. not accurate.. it is user-defined.. users make mistakes… SFTP is NOT SSH… Agghhh!! (Charlie Brown Scream…)

↩ ∞

Unop • Jan 15, 2016 @ 10:48

Perhaps you are referring to FTP/S instead? That is not SFTP.

↩ ∞

Best practice is 60 or 90 day, 14 characters minimum, and complexity requiring minimum of – 1 upper, 1 lower, 1 alpha, 1 symbol, 1 numeric.
Remember password history..

Is it convienent? No… DO passwords get weaker with time? YEs..

Why because exploits move forward every day as do caps.. Each day a password remains static, is one more oppertunity given to comprimise your system security and capture user information…
The problem w/ user passwords is that SO many users, use bank info, pins, etc…

Its a best practice… As yourself this.. If you are sued.. yes.. lawsuit.. What will you tell the prosecuting atty. when he asks if you used complexity requirements and changes on passwords?

All the attorney of the guy suing you has to prove is negligence.. Because so many passwords have been compromised.. you not enforcing it could be cionsidered negligence and could be a fatal loss to the suit..

Not saying it is right or easy.. But it's best practice and it will help keep you and your company (did I mention you) out of a bind if legal issues arise…

↩ ∞

Navneet Gaur • Aug 13, 2010 @ 11:46

Really a very good and concise article that is informative and addresses various security issues.
Very well written.
Thank you for writing and posting this article.

↩ ∞

JohnnyO • Aug 27, 2010 @ 21:18

Well written! Wow. Great great great article!

↩ ∞

jeffatrackaid • Sep 8, 2010 @ 2:26

Nice round up of some common server hardening techniques. While not specific to the server, I would add having a web application firewall, e.g. mod security or something similar. According to SANS, most exploits these days happen via web applications. Even with these tips (SELinux excepted), attackers can often setup shell kits, spam bots or similar tools.

Also, never just rely on the hardening. Using something like Nessus to audit the server. With a professional feed, you can actually audit against a variety of

Liya Comp • Sep 16, 2010 @ 15:17

Good article

↩ ∞

jef • Oct 14, 2010 @ 0:28

just what i was looking for. thanks for the info.
treat gout

↩ ∞

Eric Gillette • Oct 15, 2010 @ 20:27

Wow! This is an amazing article. Lots of things about securing a server that I either overlooked, or simply forgot about! You rock! =0)

↩ ∞

Hello • Nov 8, 2010 @ 11:29

@A G33k
If you get rid of the end user who cannot remember password, you will fire 99% of people in your company. Not a very good idea? Everybody are using yellow stickers, excel files etc. There is so many passwords to rember, most of for absolutely pointless accounts, which nobody cares.

↩ ∞

Abhijit • Nov 24, 2010 @ 15:45

Really nice article. Also, i really the comments too.
Good luck for your future.

↩ ∞

JR • Dec 26, 2010 @ 15:08

Hi,
Tried #12 Kernel/sysctl hardening, but 'sysctl -p' comes up with "error: 'kernel.exec-shield' is unknown key" on Ubuntu 10.04.1 LTS as well as Mint 9 KDE. Any ideas?
TIA

↩ ∞

Francisco • Jan 21, 2011 @ 14:30

There are several things that should be added:

* Limit the maximum number of connections with a firewall, using iptables and ip6tables.

↩ ∞

Satish • Jan 27, 2011 @ 11:55

Great Article very help full for Unix admins..

↩ ∞

Hasib • Feb 7, 2011 @ 19:54

Great site. Always find it useful in times of need.

↩ ∞

Juan • Feb 11, 2011 @ 21:17

I reviewed the comments and nobody seems to be bothered by one little fact… Hackers are not Crackers… It’s kinda disappointing to read such a “confusion” on a Unix dedicated site. Not only it is not a confusion, but it is “clarified”, openly associating and presenting the word “cracker” as a synonym for “Hacker”.

Please educate yourself: http://www.catb.org/~esr/faqs/hacker-howto.html

↩ ∞

Shadus • Feb 22, 2011 @ 20:19

Sudo is crap for security period except leaving an audit trail… which any user with sudo access can get rid of trivially. Lets say you have 5 admins each who needs root level access. With sudo that means each user’s password is another potential compromise of root level privileges. There are things you can do to help with that like using rootpw or disabling the ability to get a true shell with sudo but this breaks much of sudos functionality. Sudo is very good at offering a false sense of security and accountability of LEGITIMATE users. It does very little for non-legitimate users.

↩ ∞

Unop • Jan 15, 2016 @ 10:46

Admins with passwords ? Get them to use SSH keys and do away with passwords completely – we’re in which century now?. Then set up 2 factor auth and only allow SSH from client trusted machines/networks.

The argument that limiting sudo to a subset of commands offers a false sense of security is ridiculous – it’s exactly the point. if the number of commands that are available under sudo is low – yes, functionality takes a really need addressing).

↩ ∞

Christopher Quinn • Mar 5, 2011 @ 4:47

perfect. I was searching how to disable the root access. I love this site. I can't believe I didn't find it sooner. I switched from shared web hosting to vps web hosting and I love it.

Thanks!

↩ ∞

DSpider • Mar 24, 2011 @ 0:53

Well, Christopher… I think if, God forbid, the user account is compromised then you can simply login as root and delete it, along with it's ~/ directory. But if you disable root access… I guess you'd have to reinstall the OS.

Also, setting the "noexec" flag in fstab is a very smart move. Especially for data partitions (why would you wanna run binaries from a data partition anyway ? Programs should have no business there). I thought this flag also applied for scripts. Hmmm….

↩ ∞

Ramakrishna- krrish • Apr 29, 2011 @ 12:39

Hi Sir, Am fan to your article.. Really these are very excellent sessions.. we never get this from any other books.. Really Am so happy and we are improving our confidential levels by following your articles.. One small request, Why dont you keep an article on Solaris server issues.. Because now a days, both unix and linux are growing popular across the world.. And so many administrators are working in dual modes (LINUX and UNIX) . So, if the send an article based on linux and unix(solaris) then, so many administrators feel much better..

Thanks

↩ ∞

Ramakrishna - Krrish • Apr 30, 2011 @ 5:03

Hi Sir,

I have been trying to implement OpenLDAP server in CentOS5.4 for the past 10 months. But, till i haven't implemented. I studied and gathered so many books and articles.. even though am not succeeded. So, could you send openldap server configuration article in CentOS5. Then i can follow your help to complete the task..And i need exactly what is ldap ? why for Ldap? where to Implement with best regards..

thanks,

Ramakrishna – krrish

↩ ∞

d0rk-E • May 28, 2011 @ 20:56

I have heard the arguments for and against #7, disable root login, and am for it… But you never tell me HOW to. 😛

↩ ∞

ckdie92hc8899s9 • Jul 20, 2011 @ 19:31

WARNING to fellow DEBIAN users:

debian apt-get may break system if cannot use /tmp. Tmp may be set noexec, nosuid, etc.
To harden, may need to write pre-process script and post-process scriipt after apt-get upgrade.

alert: re”: Also, setting the â€œnoexecâ€� flag in fstab
not confirmed and demonstrated and fully tested. sorry.

Linux hostnamm 2.6.39-3.slh.xxx-aptosid-xxx64 #1 SMP PREEMPT Sat Jul xxx 2011 x86_64 GNU/Linux

Great article. Advanced persistent threats and rootkits. Kernel is the last line of defense.

obviously, strategy involves both HARDENING and SOFTENING. example of softening
is honeypot and other ‘trap doors.’ Basic – set your firefox or google chrome to send browser message as IE Internet Explorer.

Excellent Article. Intermediate. Highest return on value is getting to known how to tune the KERNEL. Second highest is learning how to compress data and ‘backup it up’ across the wide spread NET. as well as separate physical devices –
SSD preferred.

↩ ∞

Ashok • Jul 24, 2011 @ 5:57

justme19 • Aug 6, 2011 @ 15:51

Just another one of those valuable well written article. Thank you vivek for sharing this with the rest of us.

↩ ∞

michael anderson • Aug 20, 2011 @ 1:53

Is this hardening checklist good for ALL Linux distributions, such as CentOS, Fedora, Debian, Ubuntu, etc………

thanks,

↩ ∞

venkat • Sep 6, 2011 @ 12:54

Great work Vivek sir ji…

Venkat

↩ ∞

iasava • Sep 12, 2011 @ 10:43

thank for sharing. it the best best practice for me. thank you very much Vivek

↩ ∞

Rajasekhar • Oct 14, 2011 @ 4:44

ThanQ

↩ ∞

renjith • Oct 19, 2011 @ 5:54

Thanks for the gr8 info.

need to know which file we need to edit or how we can set password rules in redhat such as “password should include alphanumeric,special characters,numbers etc.

Thanks
Renjith

↩ ∞

Sankar M • Nov 12, 2011 @ 5:32

Good work!! Thanks a million for all useful tips.. 🙂

I want to show appreciation to this writer just for bailing me out of this type of issue. Right after searching throughout the world wide web and finding ways which were not helpful, I believed my life was gone. Living without the approaches to the difficulties you have fixed by means of your entire blog post is a crucial case, and those that would have in a negative way damaged my career if I hadn't encountered your web blog. Your ability and kindness in maneuvering all the details was crucial. I'm not sure what I would have done if I hadn't come across such a subject like this. It's possible to at this time relish my future. Thank you very much for the reliable and amazing guide. I won't be reluctant to refer your web blog to anyone who needs guidelines about this topic.

↩ ∞

Lamont Granquist • Dec 2, 2011 @ 21:21

You need to triage your recommendations for how much they cost to do (in terms of time):

Sites with thousands of servers and understaffed admins can't possibly do all of this, and even on smaller sites with only a few dozen boxes, there needs to be some focus on which of these offer the best bang for the amount of time spent.

You must do these:

#1: Encryption – This is good, but the suggestion to remove xinetd wholesale is generally bad, ideally use chef to only enable xinetd where needed.
#3: One service one box – This is a good goal, much more achievable in the virtualization era. Exceptions can be made, particularly with lightweight internal services.
#6: Password policy – Largely you have to do this, auditors expect it. I share the concerns about rotation leading to sickies on monitors, but I know I won't win that argument with auditors.
#7: Disable root login – Yes, remote root needs to be disabled to prevent non-reputability, I actually agree here.
#9: Disable services – Very good. Do this. Highly likely that unneeded and unmaintained services lead to actual security compromise.
#10: Disable X11 – Yep, unneeded on servers generally, don't install. Some software installation requires it, which is annoying and you'll need to make exceptions for on limited case-by-case basis.
#11: Sysctl hardening – Good and reasonably cheap. Use chef.
#15: Disable unwanted SUIDs and SGIDs – I agree, time well spent, reduces attack surface.
#17: Logging and Auditing – Past some point this just becomes using a loghost with enough disk to retain logs, and the noise level becomes insane. I wouldn't spend too much time watching all the logs all the time, although its nice if you've got a junior admin with enough free time to watch for events. In PCI situations you have to not only watch this, but respond and it becomes mandatory.

corporate border firewall, this is not necessary and can lead to headaches. This is almost in my “do not bother” list, but if you *dont* have a firewall and you’ve just got servers hanging out in the breeze on EC2 this becomes more necessary.
#16: Centralized Auth – I actually like spending the time to do Kerberos

Do not bother with these, your energy is best spent elsewhere:

#2: Removing/auditing RPMs – This became laughable to me a decade ago, nearly a complete waste of time.
#5: SElinux – Also largely a waste of time, and ongoing maintenance nightmare, most actual intrusions would be prevented by getting easier stuff right
#8: Locking down BIOS and Grub – Servers should be secure in datacenters, physical access means a compromise anyway and grub passwords get in the way of administration
#13: Seperate Partitions for Everything – Oh, FFS, I have a job to do. Complete waste of my time.
#14: Turn off IPv6 – this is laughable and becoming more indefensible now
#19: IDS – Also mostly a source of noise. I suggest using fail2ban to automate iptables blocking in response to attacks, which does something useful (e.g. ssh attacks actually chew up your cpu, and fail2ban gets that back).
#20: Encryption of files – largely a waste of time within the enterprise, other than *very* targetted systems that are high-value targets. Just get your account management right.

Most important completely missed aspect:

USE CHEF, PUPPET OR SOME OTHER CONFIG MANAGEMENT ENGINE TO ENFORCE POLICY

And yes, I wrote that in all CAPS for a reason. That should be policy #0 that comes before all else.

↩ ∞

Kishor • Dec 9, 2011 @ 19:18

Excellent article!

↩ ∞

Matteo "roghan" Cappelli • Dec 21, 2011 @ 15:10

Very good article!! 😛

↩ ∞

nbasileu • Jan 11, 2012 @ 13:38

Thx to add this 🙂

↩ ∞

bash_coder • Jan 22, 2012 @ 20:49

Well , one forgot about 8080 , port needed in some apps like ISPConfig or whatever.
Having ssh server enabled , we can disable 8080 via port forwarding in router, but use a " backdoor " aka tunnelling needed ports through ssh :
ssh -D localhost:8080 user@domain.com.
Put firefox using socksV5 127.0.0.1 and voila ! , of course ,port number can vary !
Let Mysql as default to listen only 127.0.0.1 ,enforce apache with mod_security and mod_evasive,check website folders not to be 777,and if using wordpress look for a good firewall or go write yourself a decent one to prevent sql injection.
And keep it in mind ,everything made by humans will be cracked by humans , it is just a matter of time !

Sincerly , Gabriel

↩ ∞

Arun • Mar 13, 2012 @ 14:43

Great thanks a lot…

↩ ∞

adhishesh • May 17, 2012 @ 13:01

One more thing we need to consider as a security treat, some softwares have default UserID and Password like phpmyadmin and other softwares, after installation of this kind of softwareâ€™s we need to take care of userID and Password.

↩ ∞

saroj kumar sahu • May 18, 2012 @ 18:16

Hello Dear,

Thanks a lot for your work and information to all of us…..
Thanks u boss………

↩ ∞

kc • May 27, 2012 @ 0:12

Don't forget fail2ban

↩ ∞

leon • Aug 5, 2012 @ 9:25

thanks for your valueable comments

↩ ∞

hhalat • Aug 10, 2012 @ 2:40

Very very very very usefull info. It help me a lot. Many thanks to you

↩ ∞

Shyam yeduru • Aug 28, 2012 @ 14:51

Really nice glance on linux securities..

↩ ∞

Remesh • Sep 5, 2012 @ 14:35

Thanks a lot. Its very useful.

↩ ∞

John Airey • Sep 21, 2012 @ 9:04

What about setting up a catch-all mailbox for all the root email on your servers? root's email does not normally get read on a lot of sites. Reading one mailbox is better than logging into every server to check status.

↩ ∞

CounterSpace • Sep 21, 2012 @ 17:54

I love you, Vivek. You save me everytime I have issues or questions. You make me look like an elite linux user and server admin. Thank you so much for your hard work and please do keep on keeping on.

All the best!

↩ ∞

Mickael Monsieur • Oct 11, 2012 @ 12:27

Donâ€™t forget GRSec patch for Kernel, mod_security for Apache and suhosin patch for PHP.

↩ ∞

Santosh Bhabal • Jan 16, 2013 @ 7:24

Rahul krishnan • Feb 18, 2013 @ 12:08

Thanks for the mass of information!

↩ ∞

Jason • Mar 25, 2013 @ 4:36

Great read! Thanks for taking the time to put this out there.

↩ ∞

Ravi • Jul 18, 2013 @ 11:27

Everything in one place and so neat…Thanks for sharing such a useful info… Thanks in tons….

↩ ∞

Ozjon • Jul 20, 2013 @ 6:45

Hey thanks for writing up an article on securing server. Today I had a lot of hacking on my vps server and I couldn't access any of the sites. Anyway, I had to go in and kill apache via ssh and had to switch it off for 12 hours until the hacking went away. I later realised that my wordpress sites were getting a whacked via the login path.

Your article is great – thanks for sharing.
Oz

↩ ∞

Mohammad Forhad Iftekher • Aug 1, 2013 @ 18:19

+1, very handy

Systems Administrator
Disney Interactive

↩ ∞

suresh • Sep 16, 2013 @ 9:18

Hi,

Great Article… 🙄

Thanks

↩ ∞

Great post.

↩ ∞

Sepahrad Salour • Mar 18, 2014 @ 5:53

Thanks for your great article 🙄
I really love your website…

↩ ∞

Muhasa Ivans Enock • Apr 28, 2014 @ 13:34

Great Info, I will now apply it on my new project file Server.

↩ ∞

Cody • Jul 26, 2014 @ 13:14

#13 is especially important when you consider the flaws of chroot (and any error that allows a user to chroot that is not root).

I seem to remember that /var (which yes, /var should be its own volume) and /var/tmp should be separate. More specifically, /tmp should be its own volume and /var/tmp should be a symbolic link to /tmp

But I'll leave that to each administrator … (I know there is something about this subject though but I cannot remember exactly what it is about/for. It is a complete manual about security issues, from RedHat …, that has it).

↩ ∞

Steve • Sep 19, 2014 @ 21:53

#1: the root vs sudo debate is entirely based on ignorance. the idea that "if the user is compromised, all they have to do is sudo" is simply wrong. the exact same thing applys to the root user, if they are compromised, yet minus the sudo. what sudo offers is the ability to resrict said user (with proper confuration), to specific subsets of functionality within the server. moreover, the administrative user should have a complex user name, along side a password. this means that the would-be attacker needs to brute force both a username, and a password. this decreases the likelyhood for success exponentially. finally, the sudo user should be combined with something like Two-Factor Authentication. this makes said user incredibly difficult to succumb to an attack.

#2. remote logging is NOT for constantly monitoring. it IS something all distributed networks should employ. and it DOES serve a purpose. purpose number one is the forensic logging. in the event of an intrusion, this provides an off site server where log files have been untouched by any attacker. this may be the only way to figure out what has happenend to the system, and aids in

#3 Intrusion Detection or Prevention Software is of CRITICAL importance. to claim that these things add to the “noise” is just an excuse, and lazyness, on the side of the system administrator. IDS software essentially takes the place of all those people who used to monitor forensic logging components. the idea is to create an automous system and security blanket that detects emerging threats, responds to events in real time, and alerts system administrators based on policy and threshold. combined with remote logging, this can be done with fairly low over head, and can be maintained with fairly low overhead. the ideal IDS is a combination of a generic firewall policy, file integrity checksum database software, brute force detection software, web and application firewall software, and automatic log file analysis software. this system should be able to manipulate the firewall to respond to immediate threats. and once this system is tuned for a specific use case scenario, it should be generate almost NO “noise” for the system administrator. in fact, it should lessen any noise generated by a constant barrage of botnets and rouge hosts (that which constantly probe any system).

one must make note: fail2ban is NOT intrusion detection or prevention software. it may be used as part of the over all security CHAIN… but does not cover all the essential bases. furthermore, it’s used mostly as a set-it and forget-it tool. and in this state, is only useful for brute force attacks. nothing more. and only reacts against a small number of predefined patterns.

#4 Firewall Rulesets are another CRITICAL component of any security audit. its inherently unethical for any system administrator to ignore this. after your system wide policy is defined, a generic rule set can be created to defend against generic attacks. this rule set should use split horizon like topology to ensure a back door is always available to the system administrator, and to ensure that server-to-server channels are only accessable to desirable system. a basic incoming connection ruleset helps protect against malicious malware from listening for connections in the user-space high port range. and each user should be restricted using the “owner” module available in linux, so that they are only allowed to connect out to a predefined set of servers, and on a predefined set of ports. another great feature is to ratelimit or set quotas for SYN packets going out per-user. all this helps deter malicious scripts from connecting back to a command and control center, from downloading counterparts to malware, and helps prevents the machine from participating in denial of service attacks.

5#. Auditing the software on your distributed network is essential. we are after all depending on a open source network of programmers, and security is intended… but often times realized as an afterthought. its not all that difficult to purge packages not in use. anybody who thinks this is irrelevant negates the understanding of just how a compromise is usually acheived.

where this becomes much more relevant however, is when you are activley running server software or services that have not been compiled with the latest security wise, or up to date repository. sometimes it means recompiling the software on your own.

6# its STILL important to have data on seperate partitions. however, current technology allows us to make this much easier. why define seperate partitons for everything when you can remount specific areas of your system with size allocation restrictions. again, choosing NOT to implement safe guards is just plain laziness. this is often accomplished with a one liner in your FStab

7# encryption of files IS important. however, this is usually over-thought. typically, it would make the most sense to encrypt things like: back up partitions. off-site storage. physical back up devices. system administrator /home volumes. anything with SENSITIVE information. just because it is time consuming doesn't mean you should void the process. again, please refrain from laziness. just re-think the process. there is no need to encrypt EVERYTHING, just the IMPORTANT things. moreover, automatic encryped file systems (using tools like encfs) makes this incredibly easy. there is NO excuse.

#8 refrain from laziness. it will be your undoing.

↩ ∞

Steve • Sep 19, 2014 @ 22:13

oh and #9: the MYTH that Chroot is insecure… is just that. a MYTH. the Chroot is only as secure as the system administrator defines it.

there is a reason why it is built in as a core security feature and principal of SSH, Apache, Dovecot, Sendmail, Postfix, Bind, OpenVPN, and just about any other software that allows outside user interaction with internal system functionality. if you think that they have implemented faulty secure mechanisms in the base system of our linux operating systems… you are wrong.

the rules are simple: do not run any services in chroot as Root. do not run any services inside the chroot which are running under the same user outside the chroot. if possible, seperate each service into its own chroot. use namespaces to virtualize /tmp and /var/tmp in order to inhibit race conditions. do not mount unessecary devices or filesystems. if you do mount a device or filesystem, ensure its permissions are set to "as restrictive as possible". only include nessecary applications and libraries. find a way to keep these up to date. if you cant keep them up to date easily, then hardlink or bind mount them. audit all setuid/setguid bit applications. clean up dangling symlinks. use a minimal copy of /etc/passwd and /etc/group. and so on an so forth.

why are these rules "simple"? because most of the are the same rules you should be enforcing on the BASE system. your BASE system security is just as to use it. the MYTH that you can easily break out of a chroot is also just that. a MYTH.

security is only effective when used in LAYERS, and file system virtualization of any kind is a very essential layer to any security solution.

YES, chroot was invented for a totally different purpose. but so was a whole wack of things in life. over time it has evolved to suit a plethora of different purposes, including for layering security. in fact, chroot led to namespaces, which led to virtualization. you can think of openvz as Chroot on steroids. this may be over simplifying it, but it does not effect my point.

↩ ∞

Cody • Sep 27, 2014 @ 13:12

It isn't that chroot is insecure per se. It is that it has risks (some of which depend on if the file systems are properly separated i.e., on different partitions, just like your point in regards to #6). And yes, you're right: security is a layered concept (I would rather extend your point and suggest that without layers it isn't security, at all). And yes, chroot has uses, many uses (e.g., building packages, analysis of something that is potentially risky, …, the latter which would be better in a VM like you refer to). But this question is all one needs to think about:

Why is it that the chroot system call (see chroot(2)) will give an unprivileged user the error EPERM (ie permission denied) ? Sort of like why is it that chown has similar restrictions. Of course, there's more than one thing that can prevent chroot from working, but that's not really relevant (if anything it makes the point more relevant, consider that a paradox if you want).

FreeBSD's jail syscall is stronger as is noted in the Linux man page for chroot.

So it isn't a myth any more than being logged in as root for anything beyond what absolutely must be done as root, is a bad idea. Don't have time to read the rest (only by chance saw your response to #6) but you're absolutely correct: technology evolves and that is a good thing indeed. Still, there is a reason chroot is restricted (just like chown).

↩ ∞

Mohammad Hossein • Oct 9, 2014 @ 11:03

Instead of number #2 try jailing it's a more appropriate technique.

↩ ∞

Mohammad Hossein • Oct 9, 2014 @ 11:03

sudheer • Apr 4, 2015 @ 17:26

Thanks a lot for securing my server in simple steps

↩ ∞

Pankaj • Sep 5, 2015 @ 12:14

Nice information sir

↩ ∞

securityinfo • Sep 9, 2015 @ 17:25

#20 talks about TrueCrypt but that software is not supported anymore.

↩ ∞

IRE • Jul 9, 2016 @ 12:22

Will there be an updated one for CentOS 7.x and RHEL 7.x ?

thanks

↩ ∞

Ben Dover • Sep 30, 2016 @ 17:36

For the record,
SSL = Secure Sockets Layer, not Secure Server Layer
but you knew that.

↩ ∞

Smin Rana • Nov 8, 2016 @ 6:17

Great!

↩ ∞

Vadhvi • Mar 29, 2017 @ 19:17

Thanks great tips for my CentOS 6.8 server.

↩ ∞

raju seth • Mar 29, 2017 @ 19:47

I am using to secure my CentOS 6 server. Very good guide.

↩ ∞

Vadhvi • Apr 18, 2017 @ 22:57

LinuxHostSupport.com • May 5, 2017 @ 16:23

Another useful security measure is to protect SSH with two-factor authentication. You can use the Google authenticator. It can be easily installed and configured.

↩ ∞

david • May 27, 2017 @ 6:55

Thanks for all the good stuff you provide us !

I noticed within the sentence “Read your logs using logwatch or logcheck” le link on logwatch keywork redirect to a 404 page. Do you have any updated link for that ?

↩ ∞

Gini • Aug 21, 2017 @ 3:35

Well listed items,
Thanks

↩ ∞

atomic.kidd • Nov 6, 2020 @ 5:06

all very good and excellent descriptions for immediate use – “Awesome Viv’ – one thing that erks me. Is my old addie is not using the internet I disable my Nic – software down my enps0x and re-enable to go back online. One thing I see pretty much from a cold start before clicking F/Fox or browser choice. Browser fingerprinting and inside “goo” pop’s – so if I enable my nic and immediately issue an “sudo ss -t” I see many calls out bound, tracing these using a whois service – shows MCI google amazon and many other ip hosts and yes now Lots of Collection hits MS Servers. As well my browser starts blank page – still see these pops and or more of them! I’ve not sniffed the data – but imagine its still collecting some personal use to sell to “xyz”. Avahi-daemon I believe is also a fair collector which spoofs data packets via DNS – ever sniff a Win10 box starting up you’d freak the thousands of packet sends to the many, long time now switched perma to *Nix. Browser finger printing shows via VPN, that only the IP changes – which means anywhere you pop – your browser finger print is shown immediately as soon as you pop open your browser. Again try attempt to filter on ufw/gufw ip blocks using cidr values – the numbers of collectors now simply is going on to hardening is good for corp internal networks – a single non-Networked Server is suggest to only be used off the corp network or private network to do your internet actions. Tip!. Various other Distros some have nice features keeping down this activity – or another feature be – VM/VB guest loads to do your email and internet works without using your main Mobo Ident system and open to malicious use unknown to the desktop user. Excellent share post! Like 110%

↩ ∞

Zavier Jaber • Jun 5, 2021 @ 12:08

This is a good list. I am now using them with my CentOS Linux 7 server.

شكرا لك حبيبى

↩ ∞

Syakroni • Jun 24, 2022 @ 18:25

thank you this article really helped me

↩ ∞

Ingar • Nov 9, 2022 @ 12:04

You should remove password aging. That is no longer considered a security advice.

↩ ∞

Daga5988 • Jun 9, 2023 @ 1:10

Thank you for your time and your tips. It is really, a good contribution to community. 😳

↩ ∞

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name

Next post: Linux/Unix App For Prevention Of RSI (Repetitive Strain Injury)

Previous post: Download Ubuntu 9.10 (Karmic koala) CD ISO Images

🔥 FEATURED ARTICLES

1 30 Cool Open Source Software I Discovered in 2013

2 30 Handy Bash Shell Aliases For Linux / Unix / Mac OS X

3 Top 32 Nmap Command Examples For Linux Sys/Network Admins

4 25 PHP Security Best Practices For Linux Sys Admins

5 30 Linux System Monitoring Tools Every SysAdmin Should Know

6 40 Linux Server Hardening Security Tips

7 Linux: 25 Iptables Netfilter Firewall Examples For New SysAdmins

8 Top 20 OpenSSH Server Best Security Practices

9 Top 25 Nginx Web Server Best Security Practices

10 My 10 UNIX Command Line Mistakes

👀 /etc

→ Howtos & Tutorials

→ Linux shell scripting tutorial

→ RSS/Feed

→ About nixCraft

©2002-2025 nixCraft • Privacy • ToS • Contact/Email • Corporate patron Cloudflare

Оголошення вилучено. Деталі