nixCraft
Linux Tips, Hacks, Tutorials, And Ideas In Blog Format

40 Linux Server Hardening Security Tips [2017 edition]

last updated August 20, 2017 **in Debian Linux, Howto, Linux, Monitoring, RedHat/Fedora Linux, Security, Sys admin, 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 of the Linux box. In this first part of a Linux server security series, I will provide 40 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 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

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 Type the following yum command to delete NIS, rsh and other outdated service: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

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
```

#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
```

#6: 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 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: 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: 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}:{Ina
```

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

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:

```
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
```

#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

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.

#16: 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
```

#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 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"
```

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

```
# yum group remove "GNOME Desktop"
# yum group remove "KDE Plasma Workspaces"
# yum group remove "Server with GUI"
# yum group remove "MATE Desktop"
```

#19: 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 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
- 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.

#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
```

#21: 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
```

#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

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. If you are NOT using IPv6 disable it:

- 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 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

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.

#27: 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.

#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:

```
################### 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 ttttttttttt
   general: info: zone XXXXXX.com/IN: Transfer started.: 4 Time(s)
   general: info: zone XXXXXX.com/IN: refresh: retry limit for master ttttttttttt

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

See Common Linux log files names and usage for more info.

#31: 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.

#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 computing system. I recommended that you install and use rkhunter root kit detection software too.

#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
```

Edit the config file as per your needs:

```
$ sudo vi /etc/fail2ban/jail.conf
```

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 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.
- 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:

- 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
- Subscribe to Redhat or Debian Linux security mailing list or RSS feed.

Posted by: Vivek Gite

The author is the creator of nixCraft and a seasoned sysadmin, DevOps engineer, and a trainer for the Linux operating system/Unix shell scripting. Get the **latest tutorials on SysAdmin, Linux/Unix and open source topics via RSS/XML feed** or weekly email newsletter.

GOT FEEDBACK? CLICK HERE TO JOIN THE DISCUSSION

150 comment

Lego October 30, 2009 at 10:36 am

Excellent article! Thanks for posting this.

> **veeru** January 8, 2012 at 2:17 pm
>
> sir,
> how to configure LDAP server(server side, client side) in UBUNTU linux plese tell me step by step
>
> > **Robert** January 14, 2012 at 4:21 am
> >
> > https://help.ubuntu.com/community/LDAPClientAuthentication

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 October 28, 2014 at 6:49 pm

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

Unop January 15, 2016 at 10:07 am

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 January 19, 2016 at 5:56 pm

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

Bishwajit May 9, 2016 at 3:16 am

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 October 30, 2009 at 11:21 am

Great article.

Really wroth.....

Data7 October 30, 2009 at 12:16 pm

Very useful indeed. Thanks a lot!

Dapxin October 30, 2009 at 12:26 pm

great post. One for the bookmarks....:)

Suresh October 30, 2009 at 1:31 pm

Great one sir! Thanks a lot

surendra kumar anne October 30, 2009 at 1:38 pm

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 October 30, 2009 at 2:31 pm

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

Andrew Ensley October 30, 2009 at 3:03 pm

Great article! Thanks for sharing! Bookmarked and Dugg.

nixCraft October 30, 2009 at 3:12 pm

@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 October 30, 2009 at 4:47 pm

bookmarked immediately, thank you

Toby October 30, 2009 at 5:25 pm

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.

I'm personally skeptical about password aging – strength requirements are important, but strong passwords don't get weaker over time.

> **nixCraft** October 30, 2009 at 5:39 pm
>
> Please see (#18 SSH) – a direct link Top 20 OpenSSH Server Best Security Practices.

JIS October 30, 2009 at 7:04 pm

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

robo October 30, 2009 at 7:10 pm

this is life saver for sysadmins 😉 thanks for sharing,

Theodoros Goumenidis October 30, 2009 at 7:56 pm

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

luc_rom October 30, 2009 at 8:12 pm

Great work as always Vivek.

dave October 30, 2009 at 10:05 pm

I actually stronglt disagree with 6.1 and 6.2. 6.2 Especially. Here's why (from experience as an IT manager)..

Suppose you put 6.0 (which I agree with), 6.1, and 6.2 in place and set the age of 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** September 11, 2015 at 6:19 pm
>
> 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.

---

neolix October 30, 2009 at 11:45 pm

GR8 keep posting

---

d0wnund3r01 October 31, 2009 at 12:45 am

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 October 31, 2009 at 3:48 am

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

nice post (articles)....

P Saint Amour October 31, 2009 at 3:56 am

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 October 31, 2009 at 4:37 am

Wait....I thought Linux was secure by default?

Antiks October 31, 2009 at 9:11 am

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

Solaris October 31, 2009 at 1:54 pm

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 October 31, 2009 at 4:47 pm

Excelent post...
Thanks for share your knowledge...

hideaki November 1, 2009 at 11:47 pm

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 November 2, 2009 at 9:55 am

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

Thanks
Sreeraj.K.G

John November 2, 2009 at 2:09 pm

>#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.
Agreed. I never used Truecrypt, but Wikipedia pages gives pretty good information about security.

chris j November 2, 2009 at 2:15 pm

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** January 15, 2016 at 10:24 am
>
> 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 November 2, 2009 at 3:16 pm

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 November 2, 2009 at 3:51 pm

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

hideaki November 2, 2009 at 5:45 pm

John wrote:
>Not really, how hard is to run xen under Linux?
For real? It's harder than running vmware, vbox, qemu/kvm. Because for a start you need an appropriate xen kernel.

Ricardo November 3, 2009 at 12:11 am

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 November 3, 2009 at 2:51 pm

Lots of good stuff, Thank you so much!

Gokul November 3, 2009 at 7:43 pm

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 November 4, 2009 at 7:06 pm

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.

grimr34per November 6, 2009 at 12:07 am

hideaki wrote:
>John wrote:
>>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 November 7, 2009 at 4:53 am

Great Article!! ☺

snappy November 7, 2009 at 10:40 pm

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 November 9, 2009 at 11:37 am

Thanks for a great post.

Denis November 19, 2009 at 12:27 pm

OVZkernel RHEL

[root@server etc]# sysctl -p
.....
error: “net.ipv4.icmp_ignore_bogus_error_messages” is an unknown key
_____

Why unknown key?
Thx.

> **nixCraft** November 19, 2009 at 2:46 pm
>
> OVZkernel share kernel with its host and other vps operating systems. So you will not able to use all MIBs or iptables features.

Stefano November 22, 2009 at 4:10 pm

As usual, thanks!!

K.K. November 29, 2009 at 7:50 am

It’s gr8..
Thx 4 sharing..

Praful December 11, 2009 at 5:29 am

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

prabaas December 14, 2009 at 4:14 am

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

tux4fun December 20, 2009 at 12:40 pm

Thanks for the mass of information!

Vincent January 15, 2010 at 10:03 am

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

Vincent January 15, 2010 at 10:04 am

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

Ruben January 17, 2010 at 5:25 pm

Not very useful for real production servers. Real servers (like the dozens I work with) are administered by 1-2 people accessing directly as root from local 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 January 25, 2010 at 9:39 am

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 February 7, 2010 at 11:35 am

gr8 job yaar

mrf February 10, 2010 at 12:24 pm

wow this is heaven for me he3x ☺ thx mr vivek

thyag February 19, 2010 at 7:05 pm

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

cyvan February 26, 2010 at 6:29 am

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

Andre April 25, 2010 at 10:12 pm

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

Pradeep Singh April 30, 2010 at 3:48 am

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

bharath May 25, 2010 at 11:14 pm

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

vanni linux June 9, 2010 at 10:11 am

Thanks for giving this ...
TuxRacer

Abdullah June 16, 2010 at 7:21 am

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 June 17, 2010 at 2:50 pm

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 June 21, 2010 at 3:54 pm

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 June 30, 2010 at 11:05 pm

@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.

In 2002 I had to strengthen the security for an e-commerce company. They kept the clear customer passwords in a database. You wouldn't believe how many email logins and passwords work. BTW: Passwords should be stored as hashes. Sending an email with a link to change the password is not different from a email 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 July 20, 2010 at 10:34 am

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** July 23, 2010 at 2:55 am
>
> 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** January 15, 2016 at 10:48 am
>
> Perhaps you are referring to FTP/S instead? That is not SFTP.

A G33k July 23, 2010 at 2:43 am

ANswer.. Get rid of the end user and hire someone who can remember a password..
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 August 13, 2010 at 11:46 am

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 August 27, 2010 at 9:18 pm

Well written! Wow. Great great great article!

jeffatrackaid September 8, 2010 at 2:26 am

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 policies, such as the Center for Internet Security guidelines.

Liya Comp September 16, 2010 at 3:17 pm

Good article

jef October 14, 2010 at 12:28 am

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

Eric Gillette October 15, 2010 at 8:27 pm

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 November 8, 2010 at 11:29 am

@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 November 24, 2010 at 3:45 pm

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

JR December 26, 2010 at 3:08 pm

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 January 21, 2011 at 2:30 pm

There are several things that should be added:

* For ssh disable password authentication, using public keys (on authorized_keys) is safer.

* Don't disable IPv6, learn about it, use it, promote it.

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

Satish January 27, 2011 at 11:55 am

Great Article very help full for Unix admins..

Hasib February 7, 2011 at 7:54 pm

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

Juan February 11, 2011 at 9:17 pm

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 February 22, 2011 at 8:19 pm

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** January 15, 2016 at 10:46 am
>
> 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 hit but the surface area for abuse is narrowed – and that's a good thing. Yes, set sudoku up – take the hit and then address functionality that is broken and engineer solutions to them from a better/secure starting point (you'll find that most of the things that were broken were badly written or don't really need addressing).

Christopher Quinn March 5, 2011 at 4:47 am

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 March 24, 2011 at 12:53 am

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 April 29, 2011 at 12:39 pm

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 April 30, 2011 at 5:03 am

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 ldap ?
I have so many doubts are there on ldap scenario. And how can join windows client to linux openldap server ? . If joins, how to do that ? .. So, could you explain detailedly...

with best regards..

thanks,

Ramakrishna – krrish

d0rk-E May 28, 2011 at 8:56 pm

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

ckdie92hc8899s9 July 20, 2011 at 7:31 pm

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 July 24, 2011 at 5:57 am

Fantastic Article !! Very useful one.

justme19 August 6, 2011 at 3:51 pm

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

michael anderson August 20, 2011 at 1:53 am

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

thanks,

venkat September 6, 2011 at 12:54 pm

Great work Vivek sir ji...

Venkat

iasava September 12, 2011 at 10:43 am

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

Rajasekhar October 14, 2011 at 4:44 am

ThanQ

renjith October 19, 2011 at 5:54 am

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 November 12, 2011 at 5:32 am

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

layer 3 switch November 19, 2011 at 12:35 pm

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 December 2, 2011 at 9:21 pm

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.

You should try to do these, but they're costly:

#4: Kernel upgrades – This is expensive in time, but worthwhile.
#11: Iptables/TCPwrappers – If #9 is done correctly and you've got a good 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 December 9, 2011 at 7:18 pm

Excellent article!

Matteo "roghan" Cappelli December 21, 2011 at 3:10 pm

Very good article!! 😀

nbasileu January 11, 2012 at 1:38 pm

#1
7. Nginx SSL
http://wiki.nginx.org/HttpSslModule

Thx to add this ☺

bash_coder January 22, 2012 at 8:49 pm

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 March 13, 2012 at 2:43 pm

Great thanks a lot...

adhishesh May 17, 2012 at 1:01 pm

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 at 6:16 pm

Hello Dear,

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

kc May 27, 2012 at 12:12 am

Don't forget fail2ban

satya August 2, 2012 at 10:24 am

very useful information, thankful to u for sharing this information

leon August 5, 2012 at 9:25 am

thanks for your valueable comments

hhalat August 10, 2012 at 2:40 am

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

Shyam yeduru August 28, 2012 at 2:51 pm

Really nice glance on linux securities..

Remesh September 5, 2012 at 2:35 pm

Thanks a lot. Its very useful.

John Airey September 21, 2012 at 9:04 am

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 September 21, 2012 at 5:54 pm

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 October 11, 2012 at 12:27 pm

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

Santosh Bhabal January 16, 2013 at 7:24 am

Gr8 post.
Many thanks to uploader.

Rahul krishnan February 18, 2013 at 12:08 pm

Thanks for the mass of information!

Jason March 25, 2013 at 4:36 am

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

Ravi July 18, 2013 at 11:27 am

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

Ozjon July 20, 2013 at 6:45 am

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 August 1, 2013 at 6:19 pm

+1, very handy

Systems Administrator
Disney Interactive

suresh September 16, 2013 at 9:18 am

Hi,

Great Article... ☺

Thanks

James October 16, 2013 at 4:11 pm

Thanks I needed this for a new server project that we have..

Great post!

Sepahrad Salour March 18, 2014 at 5:53 am

Thanks for your great article ☺
I really love your website...

Muhasa Ivans Enock April 28, 2014 at 1:34 pm

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

Cody July 26, 2014 at 1:14 pm

#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 September 19, 2014 at 9:53 pm

#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 identifying the security hole, repairing it, and preventing future intrusions by such means. this helps a security analyst decide whether or not the entire system has been compromised, or just part of it. and this leads me to number three

#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 kernel hardening features. i can guarantee that a large majority of production servers are running software without these features compiled in. settings kernel flags becomes a MOOT POINT if the software it self has not been compiled to USE THEM! this often means compiling and installing software from a more 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 September 19, 2014 at 10:13 pm

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 important as your chroot security. chroot is NOT a replacement for an overall audit.

chroot is still relevent in a wide range of use case scenarios. so do not be afraid 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 September 27, 2014 at 1:12 pm

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 October 9, 2014 at 11:03 am

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

Mohammad Hossein October 9, 2014 at 11:03 am

* Number #3

sudheer April 4, 2015 at 5:26 pm

Thanks a lot for securing my server in simple steps

Pankaj September 5, 2015 at 12:14 pm

Nice information sir

securityinfo September 9, 2015 at 5:25 pm

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

IRE July 9, 2016 at 12:22 pm

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

thanks

Ben Dover September 30, 2016 at 5:36 pm

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

Smin Rana November 8, 2016 at 6:17 am

Great!

Vadhvi March 29, 2017 at 7:17 pm

Thanks great tips for my CentOS 6.8 server.

raju seth March 29, 2017 at 7:47 pm

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

Vadhvi April 18, 2017 at 10:57 pm

I love this awesome tutorial. thanks you!!! Can you update it for CentOS 7? Pretty please!!!

LinuxHostSupport.com May 5, 2017 at 4:23 pm

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 at 6:55 am

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 August 21, 2017 at 3:35 am

Well listed items,
Thanks

Have a question? Post it on our forum!

Tagged as: Hardening centos, Hardening red hat, Linux, Security