Security Hardening Approach:

**SECURITY HARDENING**

1. During the design and build - Kickstart file with rules

2. SELinux

3. STIGs

4. Firewalls

5. Cronjobs

6. Authentication

7. Log Management & Monitoring

8. SSL Certificates

9. Group Policies

10. Patching and upgrades

11. User Management & Permissions

12 Microsoft Defender

13. Safelisting and Blocklisting

14. OpenSCAP

15. Change requests

16. Troubleshooting

17. Audit rules and exceptions

18. Event Handlers like Naggios and alerting

**AUTOMATION**

1. Bash scripting

2. Configuration Management using Ansible

Documentation and remediation of security vulnerability.

Script to automate recurring tasks.

Online resource:

**25 Hardening Security Tips for Linux Servers**

accuses Apple of cracking down on jailbreaking

Everybody says that **Linux** is secure by default and agreed to some extend (It’s debatable topics). However, Linux has in-built security model in place by default. Need to tune it up and customize as per your need which may help to make more secure system. Linux is harder to manage but offers more flexibility and configuration options.



**25 Linux Security and Hardening Tips**

Securing a system in a production from the hands of **hackers** and **crackers** is a challenging task for a **System Administrator**. This is our first article related to “**How to Secure Linux box**” or “**Hardening a Linux Box** “. In this post We’ll explain **25 useful tips & tricks** to secure your Linux system. Hope, below tips & tricks will help you some extend to secure your system.

**1. Physical System Security**

Configure the **BIOS** to disable booting from **CD/DVD**, **External Devices**, **Floppy Drive** in **BIOS**. Next, enable **BIOS** password & also protect **GRUB** with password to restrict physical access to your system.

1. [Set GRUB Password to Protect Linux Servers](https://www.tecmint.com/password-protect-grub-in-linux/)
   * 1. **How to Set GRUB2 Password in RHEL-based Systems**
        1. **GRand Unified Bootloader** (**GRUB**) is a [default bootloader](https://www.tecmint.com/best-linux-boot-loaders/) in all Unix-like operating systems. As promised in our earlier article “[How to reset a forgotten root password](https://www.tecmint.com/reset-forgotten-root-password-in-rhel-centos-and-fedora/)“, here we are going to review how to protect **GRUB** with a password.
        2. As we mentioned before, someone can log in to single-user mode and change system settings, which is a big security risk. To stop unauthorized access, we need to protect Grub with a password in [RHEL-based distributions](https://www.tecmint.com/redhat-based-linux-distributions/) such as Fedora, CentOS Stream, Rocky, and Alma Linux.
        3. Here, we’ll learn how to stop users from getting into single-user mode and changing system settings, especially if they can directly access the system physically.
2. **How to Password Protect GRUB Bootloader in Linux**
3. Create a password for **GRUB**, be a **root** user, and open the command prompt, type the below command.
4. grub2-setpassword
5. When prompted type **grub password** twice and press enter.
6. A computer screen shot of a black screen

   AI-generated content may be incorrect.

**I have my home lab from rockylinux-myfirstdroplet**

A computer screen with text on it

AI-generated content may be incorrect.

A computer screen shot of a black screen

AI-generated content may be incorrect.

**Password is now set on the Grub loader**

**2. Disk Partitions**

It’s important to have different partitions to obtain higher data security in case any disaster happens. By creating different partitions, data can be separated and grouped. When an unexpected accident occurs, only data of that partition will be damaged, while the data on other partitions survived. Make sure you have the following separate partitions and sure that third party applications should be installed on separate file systems under **/opt**.

Use command df -h to view current partitions:

A black background with white lines

AI-generated content may be incorrect.

/

/boot

/usr

/var

/home

/tmp

/opt

**3. Minimize Packages to Minimize Vulnerability**

Do you really want all sorts of services installed? It’s recommended to avoid installing useless packages to avoid vulnerabilities in packages. This may minimize the risk that compromise of one service may lead to compromise of other services. Find and remove or disable unwanted services from the server to minimize vulnerability. Use the ‘**chkconfig**‘command, to find out services which are running on **runlevel 3**.

# /sbin/chkconfig --list |grep '3:on'

The above command did not work for my system. So I am doing a google search for alternative processes.

**The chkconfig command can also be used to activate and deactivate services. The chkconfig --list command displays a list of system services and whether they are started (on) or stopped (off) in runlevels 0-6. At the end of the list is a section for the services managed by xinetd. First I had to install the package:**

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**I have 479 services on my system (WOW!):**

A screenshot of a computer

AI-generated content may be incorrect.

Once you’ve find out any unwanted service is running, disable them using the following command.

# chkconfig serviceName off

**systemctl disable SERVICE\_NAME**

Use the **RPM** package manager such as “**yum**” or “**apt-get**” tools to list all installed packages on a system and remove them using the following command.

# yum -y remove package-name

# sudo apt-get remove package-name

# sudo yum/dnf upgrade

**4. Check Listening Network Ports**

With the help of ‘**netstat**‘ networking, command you can view all open ports and associated programs. As I said above use ‘**chkconfig**‘ command to disable all unwanted network services from the system.

# netstat -tulpn

A screenshot of a computer program

AI-generated content may be incorrect.

**5. Use Secure Shell (SSH)**

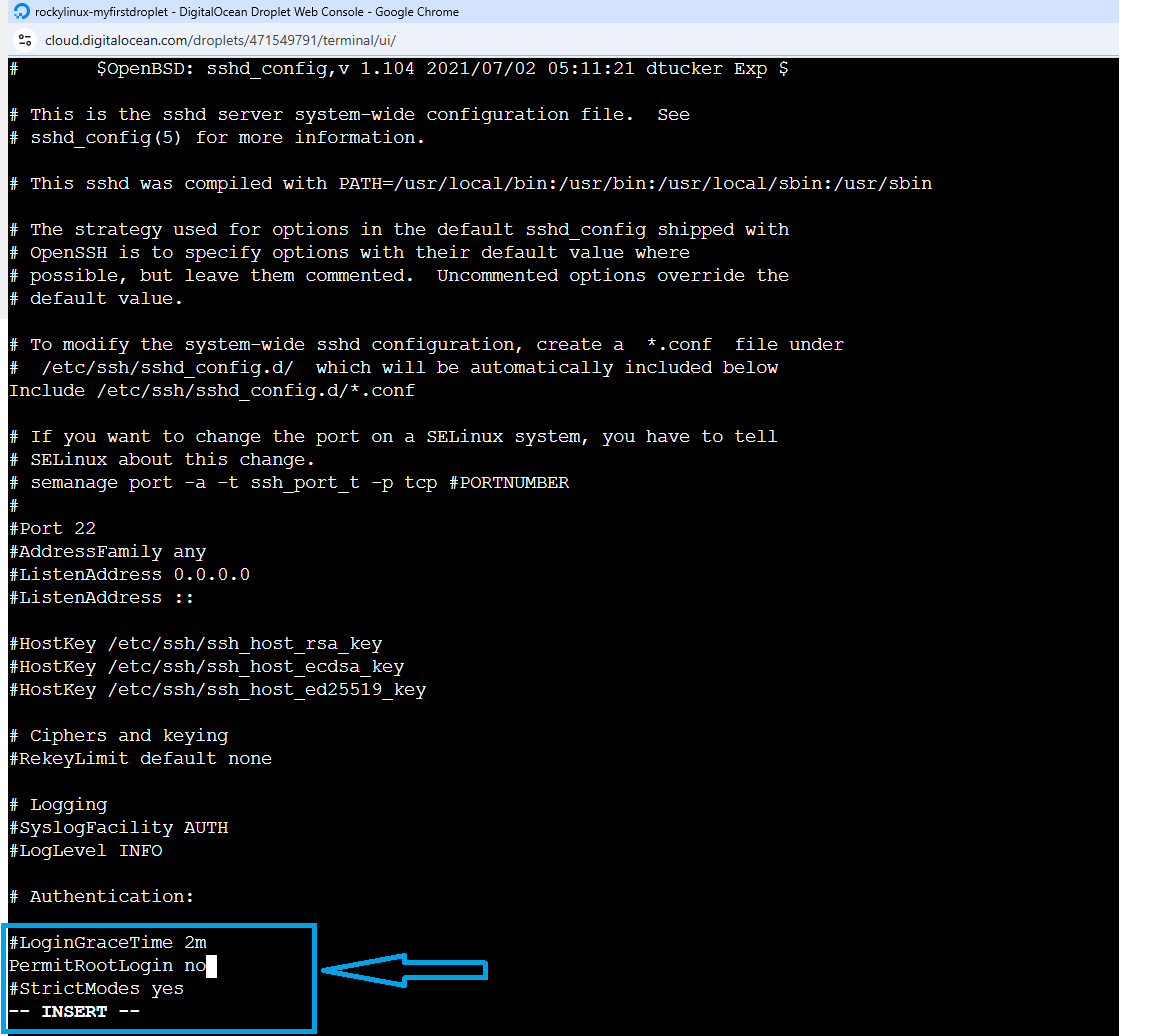
**Telnet** and **rlogin** protocols use plain text, not encrypted format which is the security breaches. **SSH** is a secure protocol that uses encryption technology during communication with server.

Never login directly as **root** unless necessary. Use “**sudo**” to execute commands. sudo are specified in **/etc/sudoers** file also can be edited with the “**visudo**” utility which opens in **VI** editor.

It’s also recommended to change default **SSH 22** port number with some other higher level port number. Open the main **SSH** configuration file and make some following parameters to restrict users to access.

# vi /etc/ssh/sshd\_config

**Disable root Login**

****

A computer screen shot of a program

AI-generated content may be incorrect.

PermitRootLogin no

**Only allow Specific Users**

AllowUsers username

**Use SSH Protocol 2 Version**

Protocol 2

**6. Keep System updated**

Always keep system updated with latest releases patches, security fixes and kernel when it’s available.

# yum updates

# yum check-update

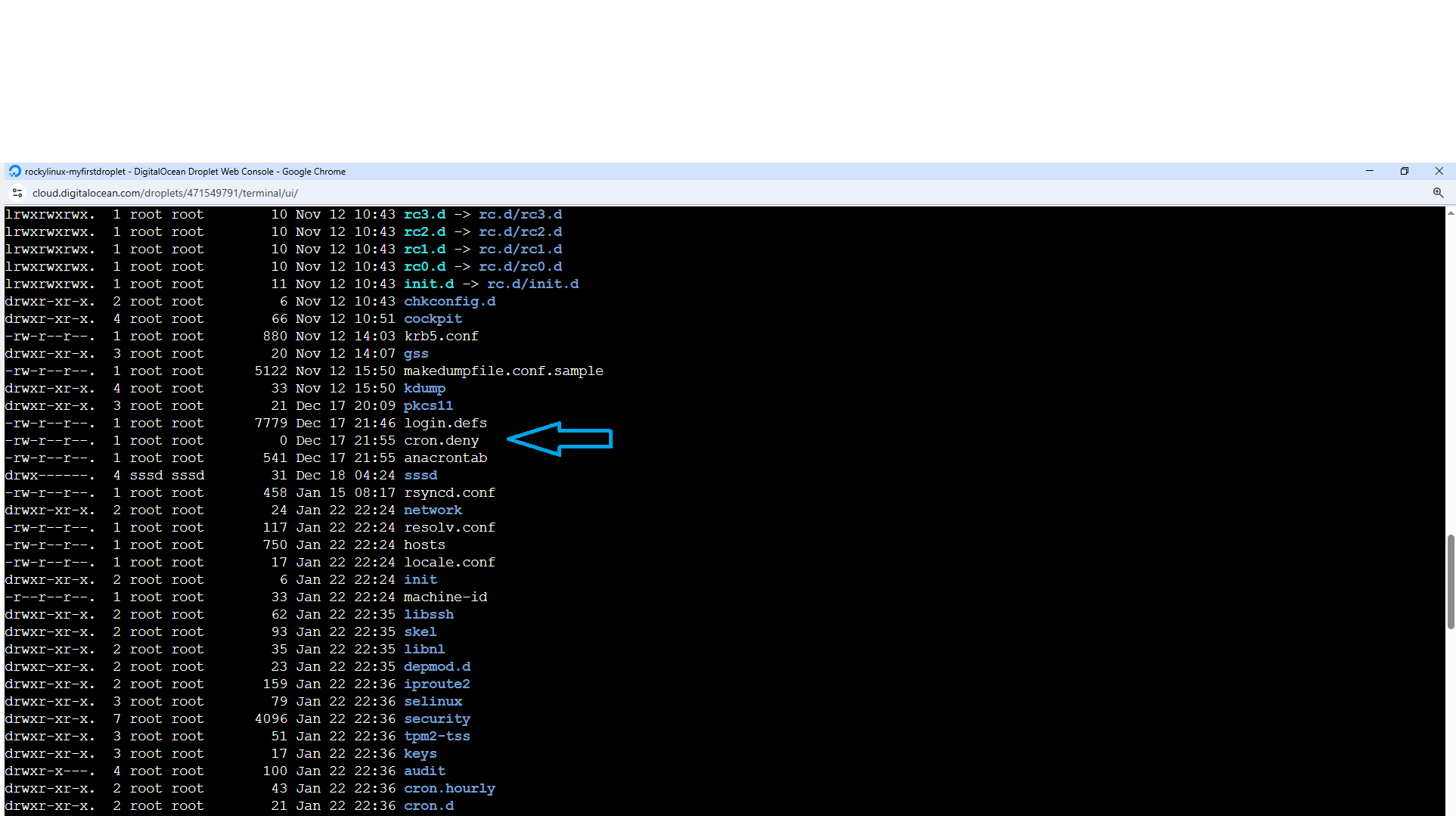
A screenshot of a computer screen

AI-generated content may be incorrect.

**7. Lockdown Cronjobs**

**Cron** has its own built-in feature, where it allows to specify who may, and who may not want to run jobs. This is controlled by the use of files called **/etc/cron.allow** and **/etc/cron.deny**. To lock a user using cron, simply add usernames in **cron.deny** and to allow a user to run cron add in **cron.allow** file. If you would like to disable all users from using cron, add the ‘**ALL**‘ line to **cron.deny** file.

# echo ALL >>/etc/cron.deny

****

**8. Disable USB stick to Detect**

Many times, it happens that we want to restrict users from using **USB** stick in systems to protect and secure data from stealing. Create a file ‘**/etc/modprobe.d/no-usb**‘ and adding below line will not detect **USB** storage.

install usb-storage /bin/true

**9. Turn on SELinux**

**Security-Enhanced Linux** (**SELinux**) is a compulsory access control security mechanism provided in the kernel. Disabling **SELinux** means removing security mechanism from the system. Think twice carefully before removing, if your system is attached to internet and accessed by the public, then think some more on it.

**SELinux** provides three basic modes of operation and they are.

1. **Enforcing**: This is default mode which enable and enforce the **SELinux** security policy on the machine.
2. **Permissive**: In this mode, **SELinux** will not enforce the security policy on the system, only warn and log actions. This mode is very useful in term of troubleshooting **SELinux** related issues.
3. **Disabled**: **SELinux** is turned off.

You can view current status of **SELinux** mode from the command line using ‘**system-config-selinux**‘, ‘**getenforce**‘ or ‘**sestatus**‘ commands.

# sestatus

If it is disabled, enable **SELinux** using the following command.

# setenforce enforcing

It also can be managed from ‘**/etc/selinux/config**‘ file, where you can enable or disable it.

**10. Remove KDE/GNOME Desktops**

There is no need to run **X Window** desktops like **KDE** or **GNOME** on your dedicated [LAMP](https://www.tecmint.com/install-apache-mysql-php-on-redhat-centos-fedora/) server. You can remove or disable them to increase security of server and performance. To disable simple open the file ‘**/etc/inittab**‘ and set run level to **3**. If you wish to remove it completely from the system, use the below command.

# yum groupremove "X Window System"

**11. Turn Off IPv6**

If you’re not using a **IPv6** protocol, then you should disable it because most of the applications or policies not required **IPv6** protocol and currently it doesn’t required on the server. Go to network configuration file and add followings lines to disable it.

# vi /etc/sysconfig/network

NETWORKING\_IPV6=no

IPV6INIT=no

**12. Restrict Users to Use Old Passwords**

This is very useful if you want to disallow users to use same old passwords. The old password file is located at **/etc/security/opasswd**. This can be achieved by using **PAM** module.

Open ‘**/etc/pam.d/system-auth**‘ file under **RHEL / CentOS / Fedora**.

# vi /etc/pam.d/system-auth

Open **‘/etc/pam.d/common-password**‘ file under **Ubuntu/Debian/Linux Mint**.

# vi /etc/pam.d/common-password

Add the following line to ‘**auth**‘ section.

auth sufficient pam\_unix.so likeauth nullok

Add the following line to ‘**password**‘ section to disallow a user from re-using last **5** password of his or her.

password sufficient pam\_unix.so nullok use\_authtok md5 shadow remember=5

Only last **5** passwords are remember by server. If you tried to use any of last **5** old passwords, you will get an error like.

Password has been already used. Choose another.

**13. How to Check Password Expiration of User**

In Linux, user’s passwords are stored in ‘**/etc/shadow**‘ file, in encrypted format. To check password expiration of user’s, you need to use ‘**chage**‘ command. It displays information of password expiration details along with last password change date. These details are used by system to decide when a user must change his/her password.

To view any existing user’s aging information such as **expiry date** and **time**, use the following command.

#chage -l username

To change password aging of any user, use the following command.

#chage -M 60 username

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

**Parameters**

1. **-M** Set maximum number of days
2. **-m** Set minimum number of days
3. **-W** Set the number of days of warning

**14. Lock and Unlock Account Manually**

The lock and unlock features are very useful, instead of removing an account from the system, you can lock it for an week or a month. To lock a specific user, you can use the follow command.

# passwd -l accountName

**Note**: The locked user is still available for **root** user only. The locking is performed by replacing encrypted password with an (**!**) string. If someone trying to access the system using this account, he will get an error similar to below.

# su - accountName

This account is currently not available.

To unlock or enable access to an locked account, use the command as. This will remove (**!**) string with encrypted password.

# passwd -u accountName

**15. Enforcing Stronger Passwords**

A number of users use soft or weak passwords and their password might be hacked with a **dictionary based** or [brute-force](https://www.tecmint.com/block-ssh-server-attacks-brute-force-attacks-using-denyhosts/) attacks. The ‘**pam\_cracklib**‘ module is available in **PAM** (**Pluggable Authentication Modules**) module stack which will force user to set strong passwords. Open the following file with an editor.

Read Also:

# vi /etc/pam.d/system-auth

And add line using credit parameters as (**lcredit**, **ucredit**, **dcredi**t and/or **ocredit** respectively lower-case, upper-case, digit and other)

/lib/security/$ISA/pam\_cracklib.so retry=3 minlen=8 lcredit=-1 ucredit=-2 dcredit=-2 ocredit=-1

**16. Enable Iptables (Firewall)**

It’s highly recommended to enable **Linux firewall** to secure unauthorized access of your servers. Apply rules in **iptables** to filters **incoming**, **outgoing** and **forwarding** packets. We can specify the source and destination address to allow and deny in specific **udp/tcp** port number.

**17. Disable Ctrl+Alt+Delete in Inittab**

In most Linux distributions, pressing ‘**CTRL-ALT-DELETE’** will takes your system to reboot process. So, it’s not a good idea to have this option enabled at least on production servers, if someone by mistakenly does this.

This is defined in ‘**/etc/inittab**‘ file, if you look closely in that file you will see a line similar to below. By default line is not commented out. We have to comment it out. This particular key sequence signalling will shut-down a system.

# Trap CTRL-ALT-DELETE

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

**18. Checking Accounts for Empty Passwords**

Any account having an empty password means it’s opened for unauthorized access to anyone on the web and it’s a part of security within a Linux server. So, you must make sure all accounts have strong passwords and no one has any authorized access. Empty password accounts are security risks and that can be easily hackable. To check if there were any accounts with empty password, use the following command.

# cat /etc/shadow | awk -F: '($2==""){print $1}'

**19. Display SSH Banner Before Login**

It’s always a better idea to have a legal banner or security banners with some security warnings before SSH authentication. To set such banners read the following article.

**20. Monitor User Activities**

If you are dealing with lots of users, then it’s important to collect the information of each user’s activities and processes consumed by them and analyze them at a later time or in case if any kind of performance, security issues. But how we can monitor and collect user activities information.

There are two useful tools called ‘**psacct**‘ and ‘**acct**‘ are used for monitoring user activities and processes on a system. These tools runs in a system background and continuously tracks each user activity on a system and resources consumed by services such as **Apache**, **MySQL**, **SSH**, **FTP**, etc. For more information about installation, configuration and usage, visit the below url.

1. [Monitor User Activity with psacct or acct Commands](https://www.tecmint.com/how-to-monitor-user-activity-with-psacct-or-acct-tools)

**21. Review Logs Regularly**

Move logs in dedicated log server, this may prevent intruders to easily modify local logs. Below are the Common Linux default log files name and their usage:

1. **/var/log/message** – Where whole system logs or current activity logs are available.
2. **/var/log/auth.log** – Authentication logs.
3. **/var/log/kern.log** – Kernel logs.
4. **/var/log/cron.log** – Crond logs (cron job).
5. **/var/log/maillog** – Mail server logs.
6. **/var/log/boot.log** – System boot log.
7. **/var/log/mysqld.log** – MySQL database server log file.
8. **/var/log/secure** – Authentication log.
9. **/var/log/utmp** or **/var/log/wtmp** : Login records file.
10. **/var/log/yum.lo**g: Yum log files.

**22. Important file Backup**

In a production system, it is necessary to take important files backup and keep them in safety vault, remote site or offsite for Disasters recovery.

**23. NIC Bonding**

There are two types of mode in **NIC** bonding, need to mention in bonding interface.

1. **mode=0** – Round Robin
2. **mode=1** – Active and Backup

**NIC Bonding** helps us to avoid a single point of failure. In **NIC** bonding, we bond two or more **Network Ethernet Cards** together and make one single virtual Interface where we can assign **IP** addresses to talk with other servers. Our network will be available in case one **NIC Card** is down or unavailable due to any reason.

**24. Keep /boot as read-only**

Linux kernel and its related files are in **/boot** directory which is by default as **read-write**. Changing it to **read-only** reduces the risk of unauthorized modification of critical boot files. To do this, open “**/etc/fstab**” file.

# vi /etc/fstab

Add the following line at the bottom, save and close it.

LABEL=/boot /boot ext2 defaults,ro 1 2

Please note that you need to reset the change to read-write if you need to upgrade the kernel in future.

**25. Ignore ICMP or Broadcast Request**

Add following line in “**/etc/sysctl.conf**” file to ignore **ping** or **broadcast** request.

Ignore ICMP request:

net.ipv4.icmp\_echo\_ignore\_all = 1

Ignore Broadcast request:

net.ipv4.icmp\_echo\_ignore\_broadcasts = 1

Load new settings or changes, by running following command

#sysctl -p

If you’ve missed any important security or hardening tip in the above list, or you’ve any other tip that needs to be included in the list. Please drop your comments in our comment box. **TecMint** is always interested in receiving comments, suggestions as well as discussion for improvement.