Linux Hardening Techniques

By: Parveen Dhankhar

Contents

[1. Disable Remote Root Access 3](#_Toc108378753)

[2. Set a boot password for GRUB 4](#_Toc108378754)

[3. Linux Hard disk Encryption: 6](#_Toc108378755)

[4. Security-Enhanced Linux(SELinux) 7](#_Toc108378756)

[5. Disable USB 8](#_Toc108378757)

[6. Enable Firewall 8](#_Toc108378758)

[7. Reduce attack surface 10](#_Toc108378759)

[8. Split Disk Partitions 10](#_Toc108378760)

[9. Install Logwatch / Logcheck 11](#_Toc108378761)

[10. Install IDS Solutions 11](#_Toc108378762)

[11. Install IPS Solutions 11](#_Toc108378763)

[12. Set Password Expiration Dates 12](#_Toc108378764)

[13. Maintain User Accounts and Password Policy 12](#_Toc108378765)

[14. Lock Login Attempts after Failure 13](#_Toc108378766)

[15. Disable Login as Super User 13](#_Toc108378767)

[16. Verify Listening Ports 13](#_Toc108378768)

[17. Keep boot directory as read only 14](#_Toc108378769)

[18. Disable Ctrl+Alt+Delete 14](#_Toc108378770)

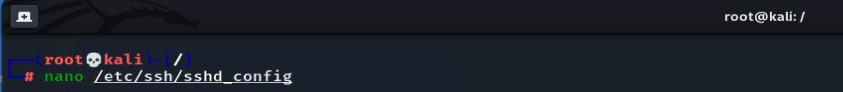
[19. Create Regular Backups 15](#_Toc108378771)

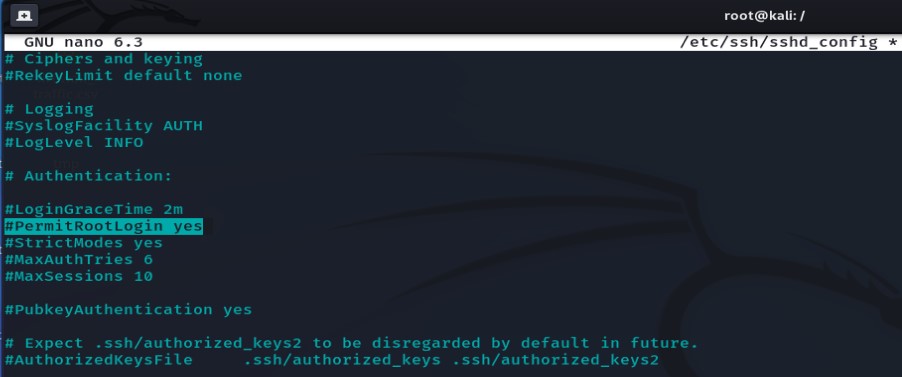
# **Disable Remote Root Access**

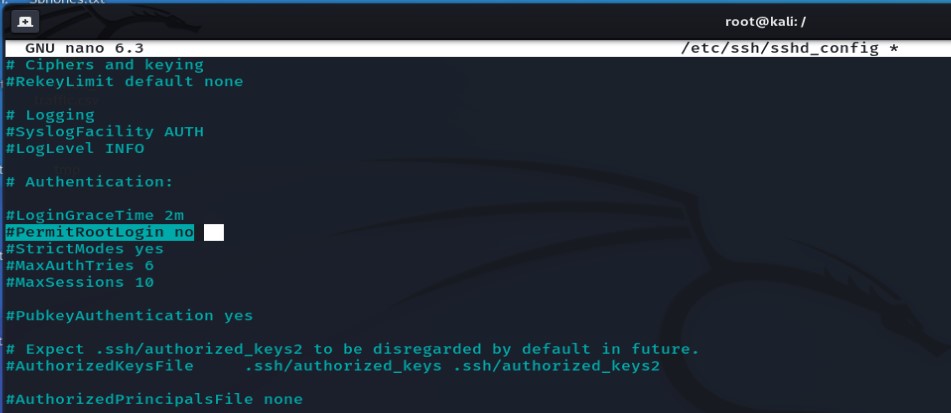
The root user in Linux has full, unlimited access to the system; by preventing root login directly, we can increase security, as attackers often try to hack the root account.

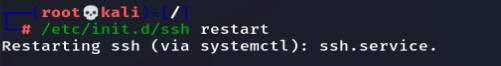
There are two config files ‘ssh\_config’ and ‘sshd\_config’ under /etc/ssh/ directory. ssh\_config file contains configuration pertaining to outbound SSH connections. whereas sshd\_config file contains configuration parameters which controls inbound SSH connections to the server itself.

Open /etc/ssh/sshd\_config file in your favorite text editor and find the line labeled PermitRootLogin.









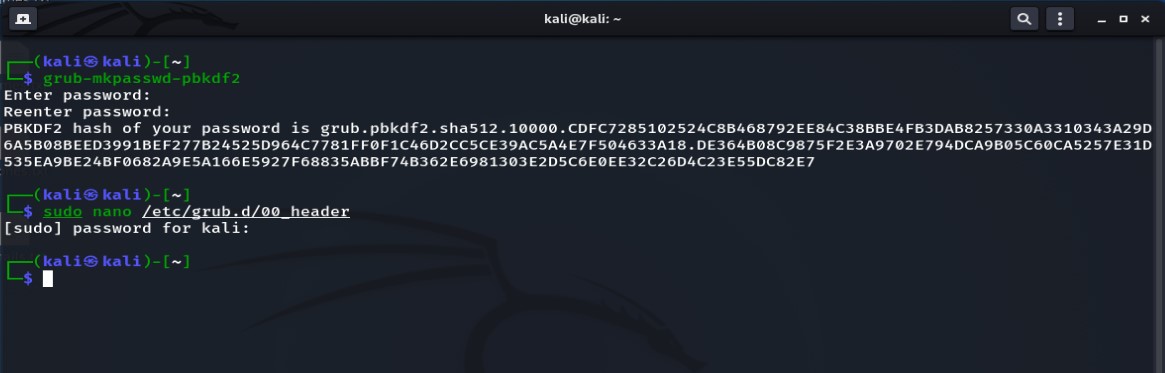
# **Set a boot password for GRUB**

* $ grub-mkpasswd-pbkdf2

This command is to generate a password hash.

* $ sudo nano /etc/grub.d/00\_header

Now we need to do some changes to the GRUB configuration file.



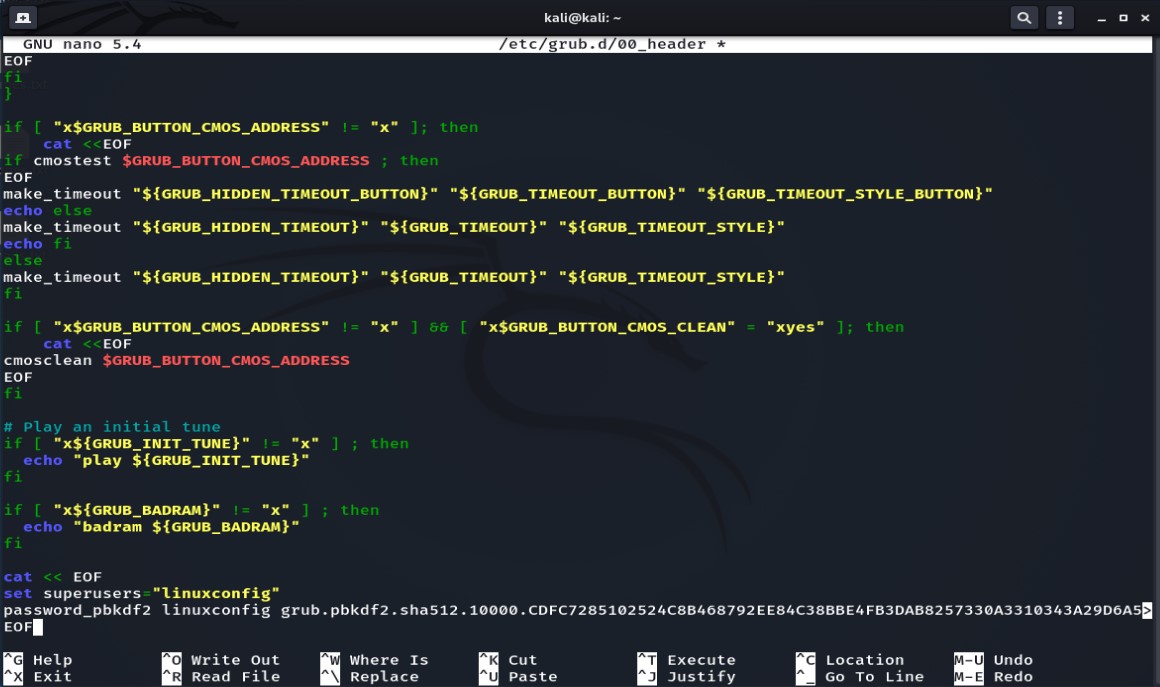
At the bottom of this file, we need to add the following code:

cat << EOF

set superusers="linuxconfig"

password\_pbkdf2 linuxconfig INSERT HASH

EOF

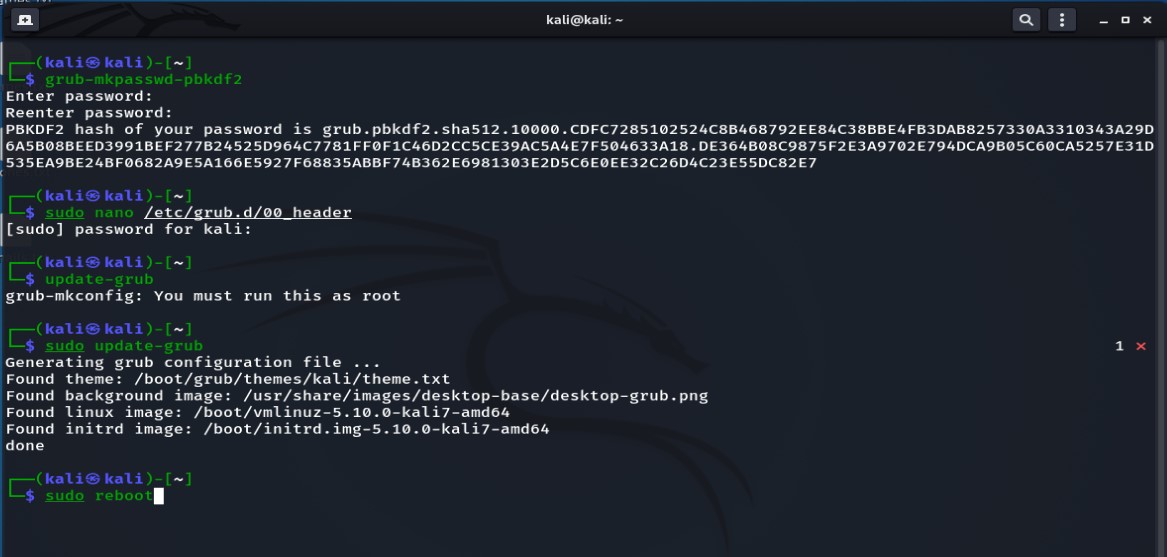


* $ sudo update-grub

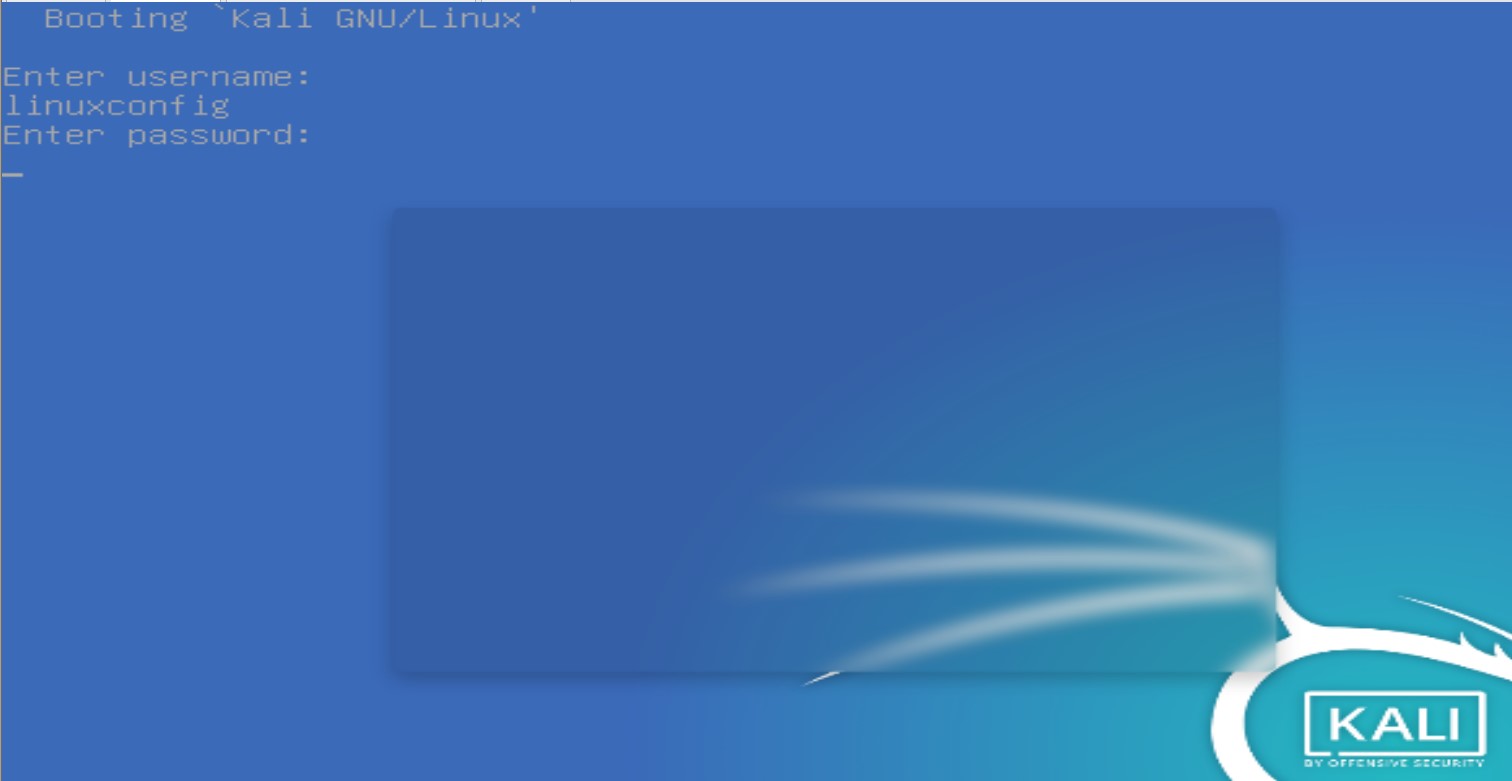
This command will set the GRUB login and password.

* $ sudo reboot

Then reboot the system by above command.



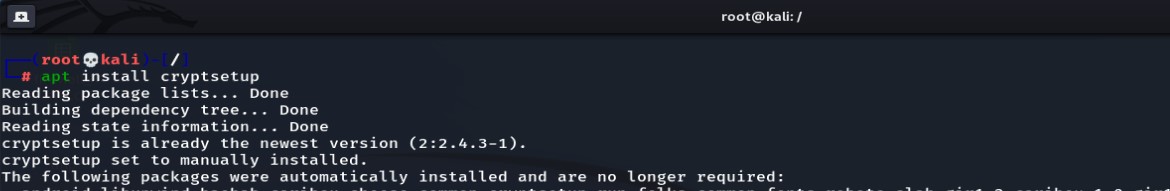
After rebooting system will ask for login and password in booting process.



# **Linux Hard disk Encryption:**

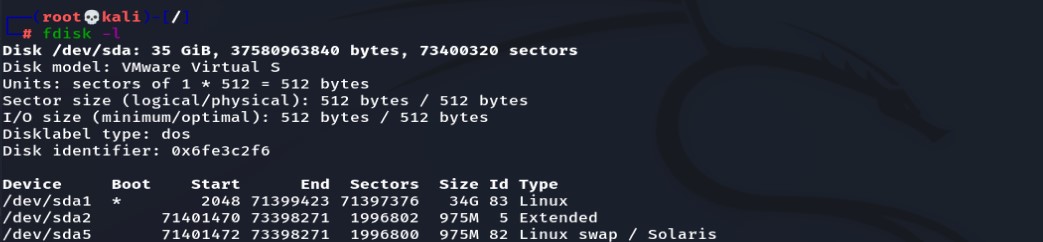
You need to install the following package. It contains cryptsetup, a utility for setting up encrypted filesystems.

* apt install cryptsetup



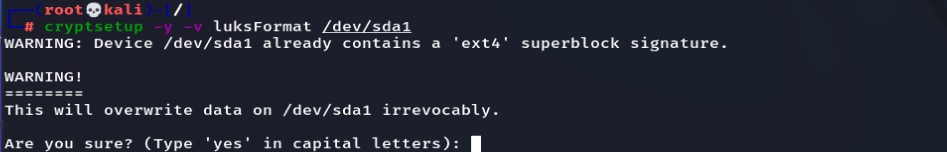
Now check the partitions by following command:

* fdisk -l



Select which partition you want to encrypt. Here we select the first partition, then run the following command:

* cryptsetup -y -v luksFormat /dev/sda1



Type YES to proceed then it will ask for passphrase.



# **Security-Enhanced Linux(SELinux)**

SELinux reduces vulnerability to privilege escalation, provides fine grain access control and separates processes from each other. Processes run in their own domain which prevents them from accessing files used by other processes.

SELinux has 3 modes.

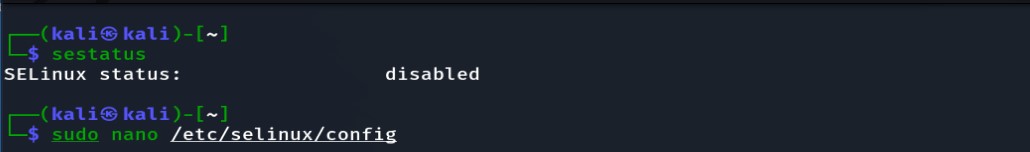
Enforcing mode: This is the default mode. It blocks and logs actions that are against defined policy.

Permissive mode: Allows actions to take place and logs the events in detail. This mode is useful when testing SELinux features. Changing modes between enforcing and permissive does not require a system reboot.

Disabled mode: Allows for all actions and does not log any activity. Changing to this mode requires a system reboot for it to apply.

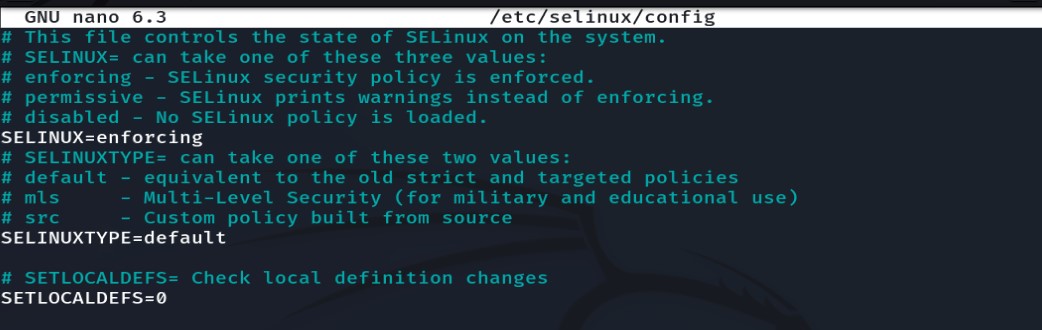
To check the current settings, type the following command in your terminal:

* sestatus



By default, status is disabled , we have to enable it by editing **/etc/selinux/config** file by following command:

* sudo nano /etc/selinux/config



Now Reboot the system by:

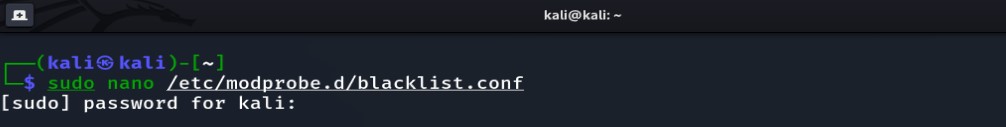
* sudo reboot

# **Disable USB**

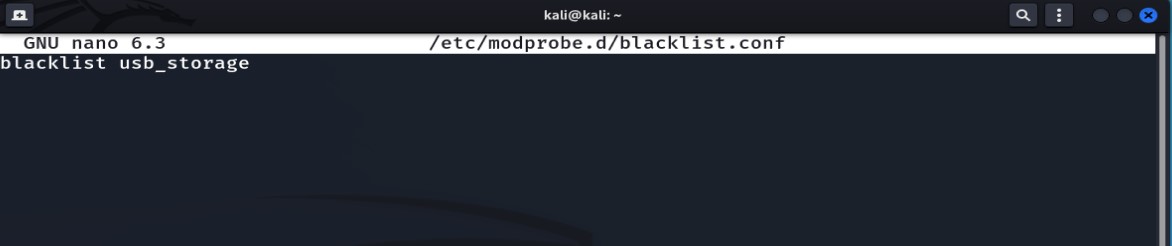
USB sticks contain harmful malware, which may be used to steal your personal data. So, it is a good idea to disable USB.

Open **/etc/modeprobe.d/blacklist.conf** file by following command:

* sudo nano /etc/modeprobe.d/blacklist/conf



Then add **blacklist usb-storage** in the file.



Save and close the file.

# **Enable Firewall**

Firewall decides fate of packets incoming and outgoing in system. IPTables is a rule-based firewall, and it is pre-installed on most of Linux operating system. By default, it runs without any rules.

We can use some other firewalls like “ufw” or “gufw”. Ufw (Uncomplicated Firewall) is more effective than iptables because it works at network layer where iptables works at transport layer.

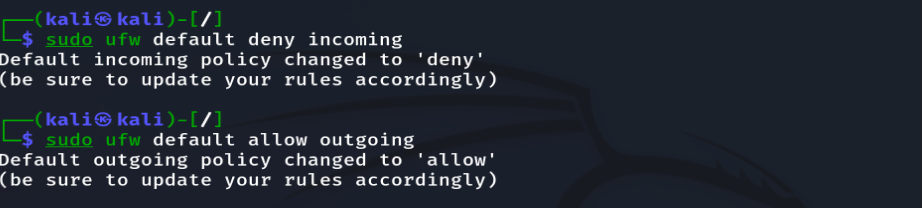
Use below command to install ufw :

* apt-get install ufw



Basic configuration of ufw firewall:

* sudo ufw default deny incoming - to block any traffic towards the server
* sudo ufw default allow outgoing - to allow any outgoing traffic from the server



Allow or deny from specific ip address

* sudo ufw allow from ip\_address
* sudo ufw deny from ip\_address

Allow or deny any port number

* sudo ufw allow port\_number
* sudo ufw deny port\_number

Viewing and deleting existing rules

* sudo ufw status numbered

Enabling, disabling, or resetting UFW:

* sudo ufw enable

To remove it from the automatic start, just use the disable command :

* sudo ufw disable

If you need to reset the UFW settings and then remove any connection acceptance / denial rules and policies, use the reset command

* sudo ufw reset

UFW Firewall log

To enable data logging, the command to be performed is:

* sudo ufw logging on

On the contrary, to disable logging type:

* sudo ufw logging off

The UFW logs can be consulted from the file "/var/log/ufw.log"

* nano /var/log/ufw.log

# **Reduce attack surface**

Your attack surface can be reduced by disabling services that are not needed and by uninstalling or removing packages and software that are not required. The following commands can be used to view the status of installed services, they will list services that are configured to start up on boot.

To check which services are running on the system:

* service --status-all

You can start and stop a service by:

* service service-name start
* service service-name stop

# **Split Disk Partitions**

The Linux filesystem divides everything into several parts based on their use case. You can separate the critical portions of the filesystem into different partitions of your disk storage. For example, the following filesystems should be split into different partitions.

* /usr
* /home
* /var & /var/tmp
* /tmp

You should also create separate partitions for different services like for Apache and FTP server roots. This helps to isolate the sensitive portions of your system. Thus, even if a malicious user gains access to some part of the system, he/she cannot roam freely through the entire system.

# **Install Logwatch / Logcheck**

Log files typically contain a wealth of data, most of which is unimportant for hardening Linux. Administrators can monitor suspicious logs using tools like Logwatch and Logcheck. They filter out common entries that are expected in your logs and only draw your attention to unusual entries.

**Logwatch** is an extremely powerful log analyser that can make log management much easier.

We can install it by using following command:

* sudo apt-get install logwatch

**Logcheck** is considerably simpler compared to logwatch. It mails admin as soon as there occur any suspicious logs.

* sudo apt-get install logcheck

# **Install IDS Solutions**

Using an IDS is one of the greatest ways to harden Linux on servers (Intrusion Detection Software). The Advanced Intrusion Detection Environment (AIDE) comes highly recommended for this job. It is a host-based IDS with a wide range of powerful capabilities, such as support for compression, several message digest algorithms, file attributes, and regex.

* sudo apt-get aide

# **Install IPS Solutions**

Network servers are shielded from brute force attacks by IPS, or intrusion prevention software. Setting up a suitable IPS will benefit you in the long term because a considerable number of malicious people and bots are attempting to access your remote server.

**Fail2Ban** is one of the most popular IPS solutions for Unix-like systems. It is written using Python and is available on all POSIX-compliant platforms. It will look for obtrusive network requests all the time and block them as soon as possible. Install Fail2Ban using the below command.

* sudo apt-get install -y fail2ban

**DenyHosts** is another popular IPS solution for Linux hardening. It will protect your ssh servers from intrusive brute force attempts. Use the following commands to install:

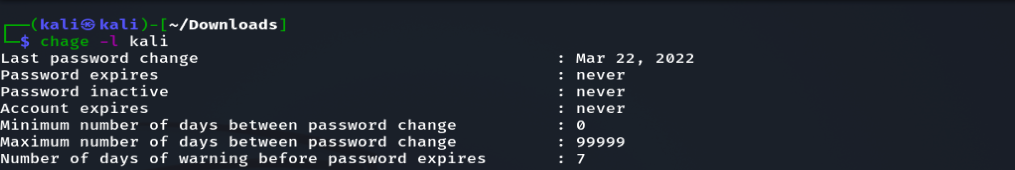
* sudo apt-get install -y denyhosts

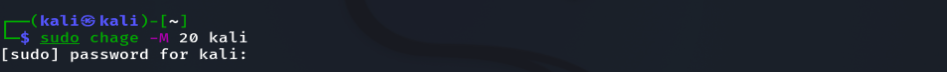
# **Set Password Expiration Dates**

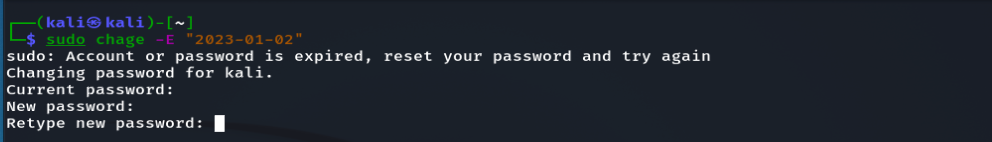
Another Linux hardening method is to enable password expiration for all user accounts. You can easily set expiration dates for user passwords by utilizing the **chage** command in Linux. Your system will ask users to set a new password once their existing once expire.

* chage -l kali
* chage -M 20 kali
* chage -E "2023-01-02"

The first command lists the current password expiration date for the user kali. The second command sets the expiration date after 30 days. You can also set this date using a YYYY-MM-DD format by using the third command.







# **Maintain User Accounts and Password Policy**

There are typically many user accounts on Unix computers. The security of your system depends on the users that utilise it. Therefore, ensure that a system can only be operated by authorised users. To create and manage new user accounts on your workstation, use the useradd/usermod programmes.

Always impose strict password policies. A strong password should have at least a mix of letters, numbers, and special characters, and be longer than eight characters.

# **Lock Login Attempts after Failure**

Administrators should ensure that after a given number of unsuccessful attempts, users are unable to get into their server. By reducing password threats, this raises the system's overall security. The failed login attempts can be seen using the Linux **faillog** command.

* **sudo faillog** - Display the failed login attempts for users from the /var/log/faillog database
* **sudo faillog -m 3** - Sets the maximum number of allowed failed login attempts to 3.
* **sudo faillog -l 1800 -** Sets a lock of 1800 seconds or 30 minutes
* **sudo faillog -r -u <username>** To unlock a user once they’re prohibited from login.

# **Disable Login as Super User**

Admins should not frequently log in as root to maintain server security. Instead, you can use sudo execute Linux terminal commands that require low-level privileges. The below command shows how to create a new user with sudo privileges.

* adduser <username> sudo

You can also grant sudo privileges to existing users by using the below command.

* usermod -a -G sudo <username>

# **Verify Listening Ports**

Network attacks are extremely common on servers. If you want to maintain a secure server, you should validate the listening network ports occasionally. This will provide you essential information about your network.

* netstat -tulpn
* ss -tulpn
* nmap -sT -O localhost
* nmap -sT -O server.example.com

You can use any of the above commands to see which ports are listening for incoming requests.

# **Keep boot directory as read only**

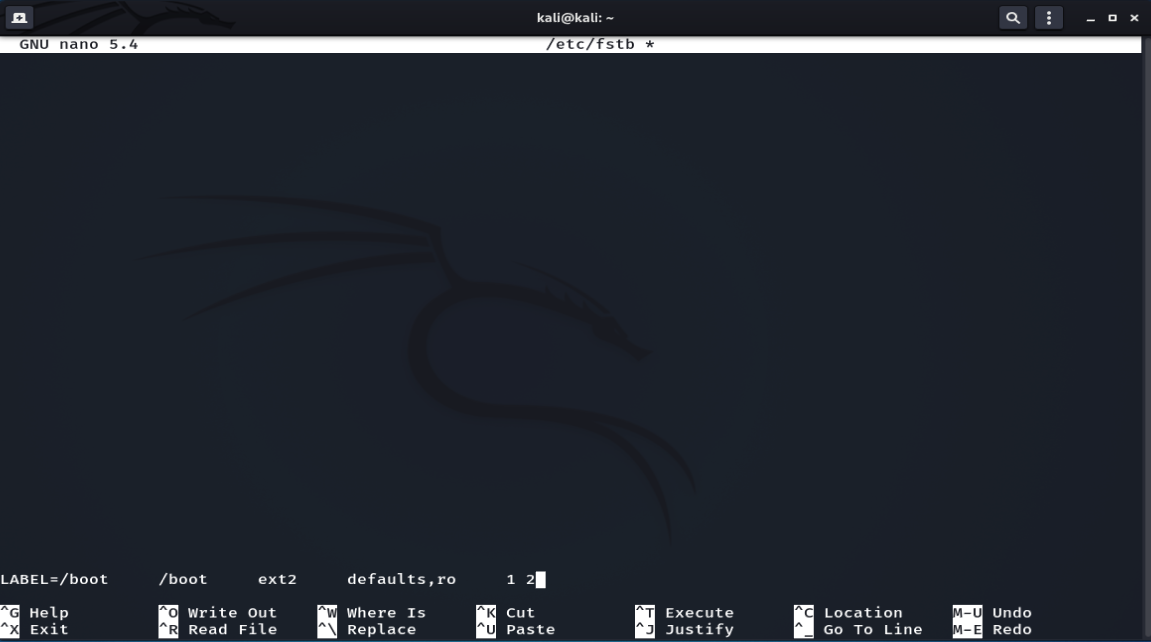
The boot directory contains important files related to the Linux kernel, so you need to make sure that this directory is locked down to read-only permissions by following the next simple steps. Open “/etc/fstab” file.

* sudo nano /etc/fstb



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

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

****

# **Disable Ctrl+Alt+Delete**

The **Ctrl+Alt+Delete** key combinations allow users to force reboot many Linux distributions. This can be particularly problematic if you’re managing a secure server. Admins should disable this hotkey to maintain proper Linux hardening. You can run the following command to disable this in systemd-based systems.

* systemctl mask ctrl-alt-del.target



# **Create Regular Backups**

No matter how many methods of hardening Linux you employ, you must always be ready for unforeseen issues. In the long term, backing up your desktop or server can be quite advantageous. To make system backups simpler, a variety of backup utilities for Linux are thankfully available. Additionally, you need to secure your system data storage.