

Sri Lanka Institute of Information Technology

Specializing in Cyber Security



**IE2012 - Systems and Network Programming
SNP Assignment**

IT23238794 – DISSANAYAKE Y Y

Table of Contents.

1. Basics of Linux Environment.

1. Virtual Machine Setup
2. Command Line Introduction
 1. Basic Navigation Commands
 2. File Manipulation Commands
3. System Information and User Management
 1. System Information Commands
 2. User Management Commands

2. DHCP, DNS and NTP Services.

1. DHCP (Domain Host Configuration Protocol)
2. DNS (Domain Name System)
3. NTP (Network Time Protocol)

3. Shell Scripting and Security.

1. Shell Scripting
2. SSH (Secure Shell)
3. Iptables and ACLs

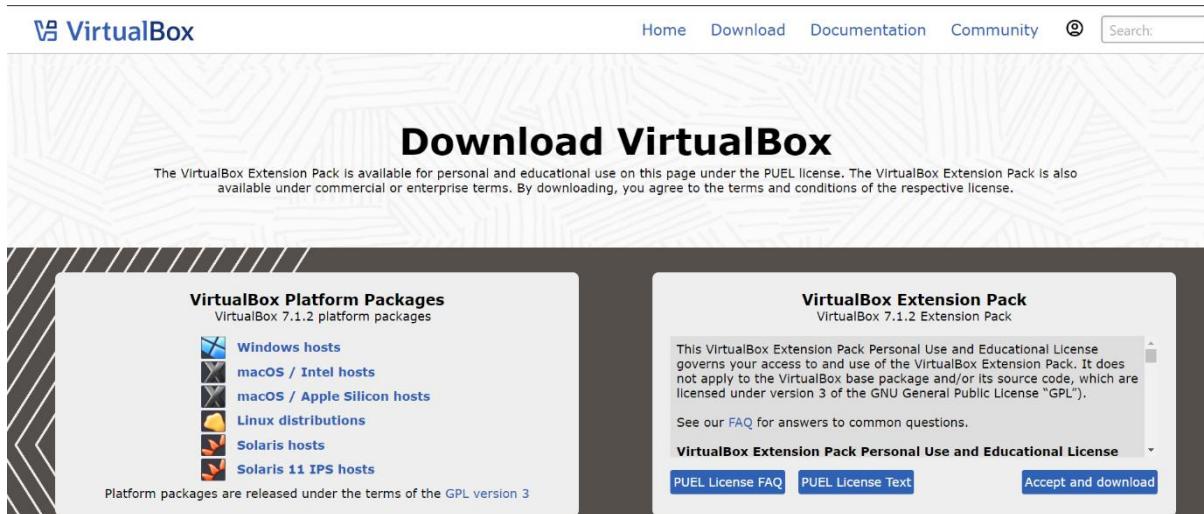
4. Best Practices.

1. Implementing a Firewall in Kali Linux.
2. Minimize Open Ports and Services.
3. Disable Network Interfaces.
4. Modify the Default Password Policy Settings.
5. Change Default SSH Port.

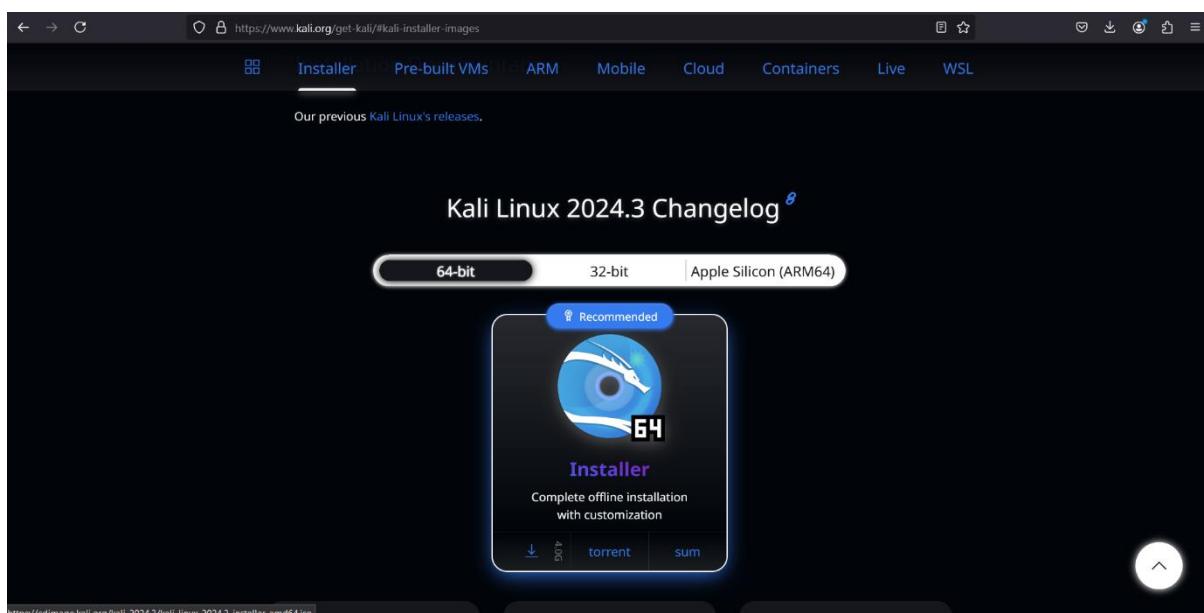
1. Basics of Linux Environment.

1.1 Virtual Machine Setup

Install a virtual emulator on your Mac, Windows, or Linux computer first. I like Oracle VirtualBox. Run the setup file and carry out the Windows system installation after the download is finished.

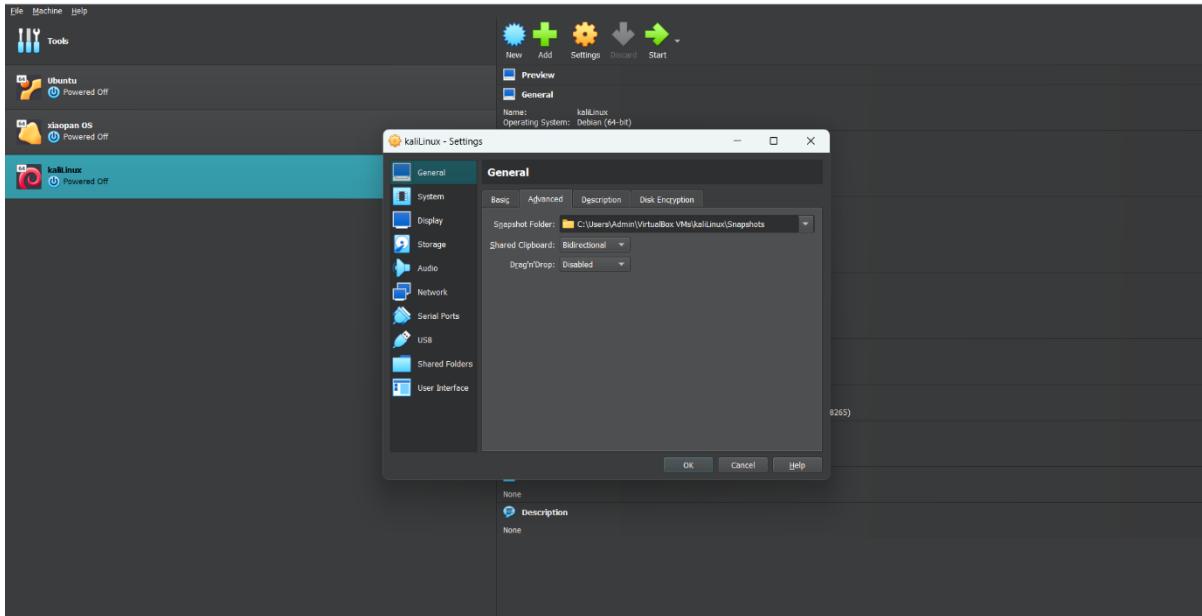


Go to the official website and download the Kali operating system.

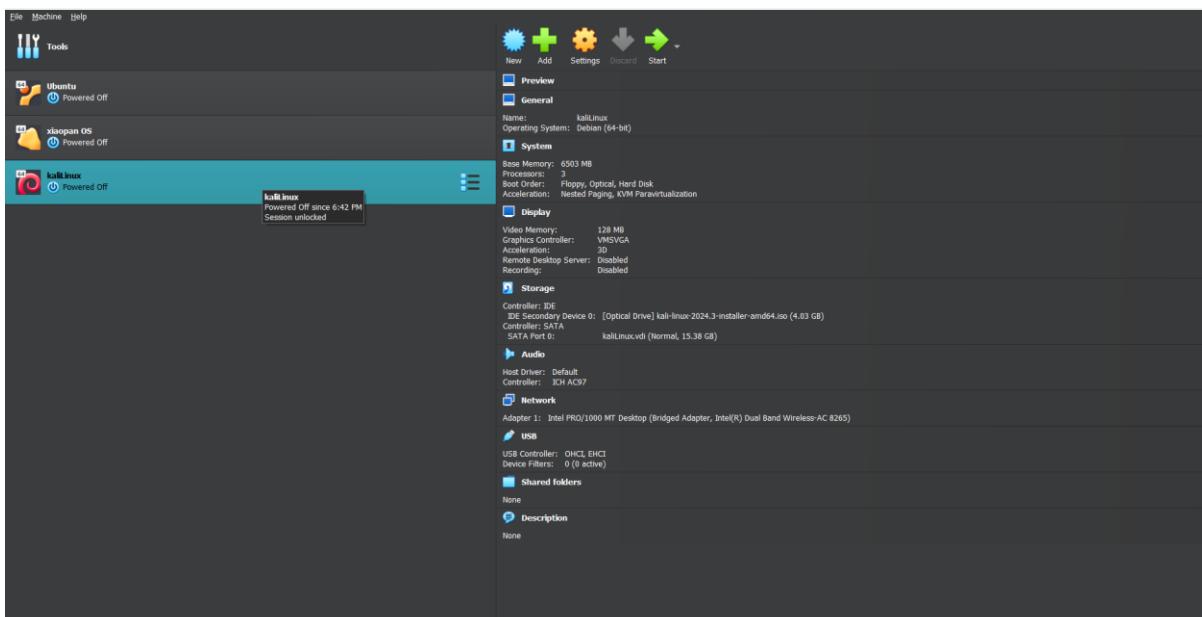


Install and configure VirtualBox. Click the New button and fill in the Create Virtual Machine. Then click Next button and type username and password. Create Virtual Machine.

Then change settings in virtual box.



successfully installed Kali as a virtual machine.



Then change the criteria after power on the kali Linux.

KALI

Select a language

Choose the language to be used for the installation process. The selected language will also be the default language for the installed system.

Language:

Chinese (Traditional)	- 中文(繁體)
Croatian	- Hrvatski
Czech	- Čeština
Danish	- Dansk
Dutch	- Nederlands
Dzongkha	- གྱଣଧ୍ୱନୀ
English	- English
Esperanto	- Esperanto
Estonian	- Eesti

KALI

Select your location

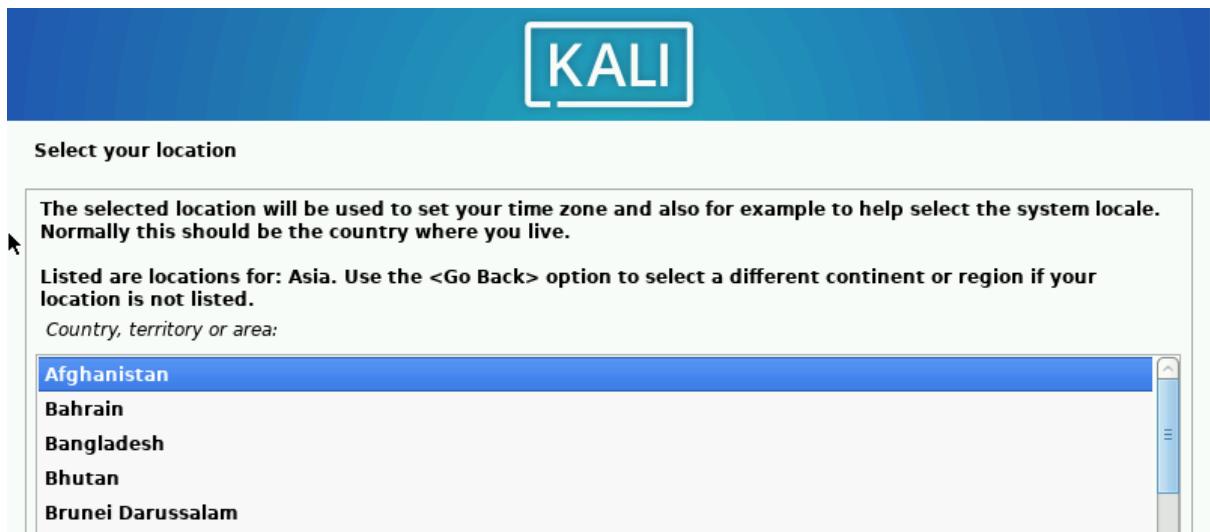
The selected location will be used to set your time zone and also for example to help select the system locale. Normally this should be the country where you live.

Select the continent or region to which your location belongs.

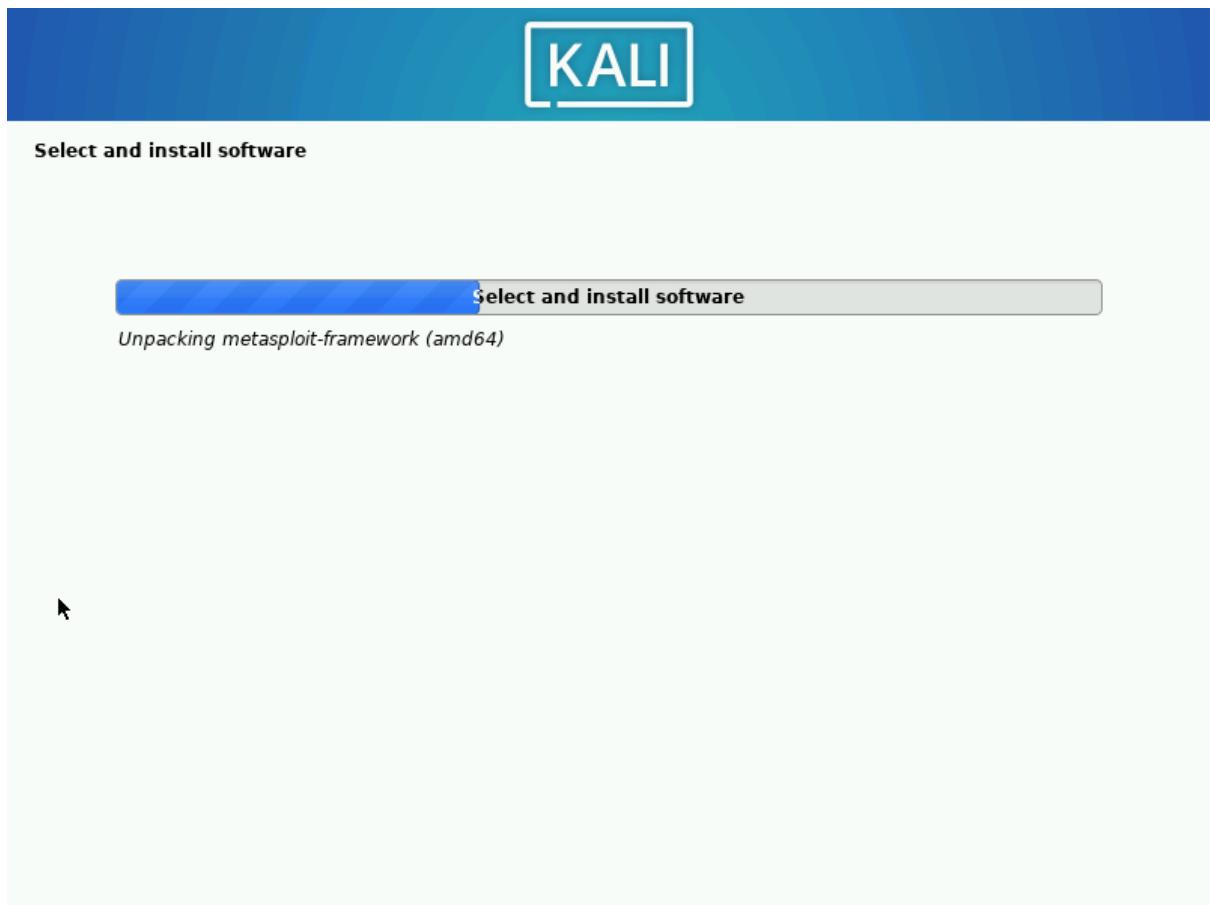
Continent or region:

Africa
Antarctica
Asia
Atlantic Ocean
Caribbean
Central America
Europe
Indian Ocean
North America
Oceania
South America

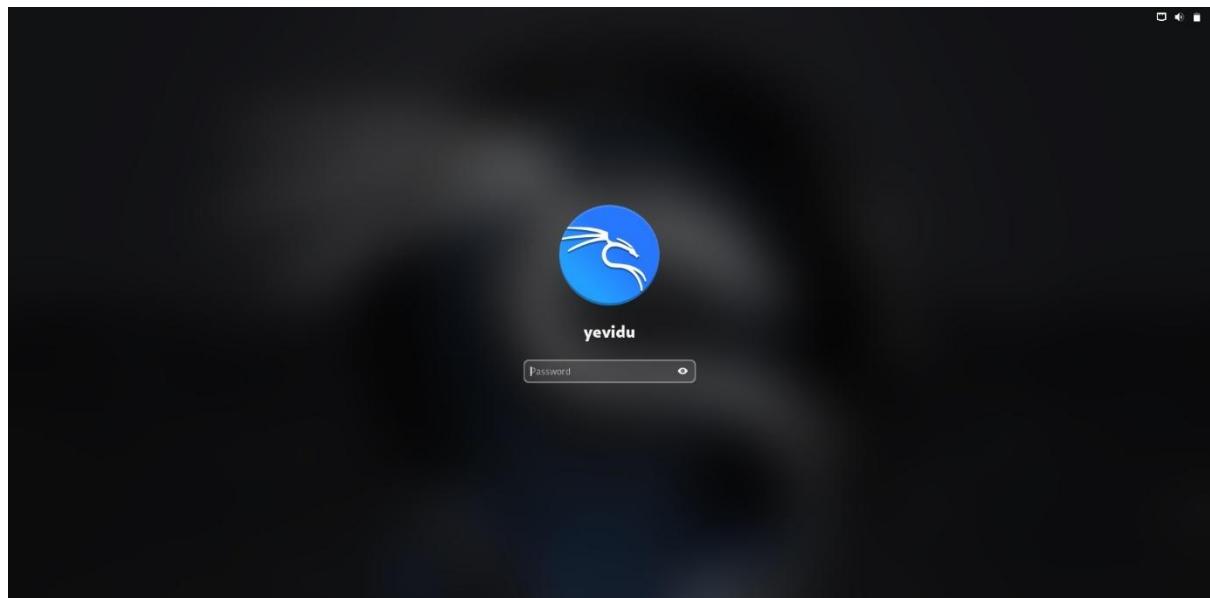
Screenshot **Go Back** **Continue**



Then waiting for installing process.



Finally installing completed.



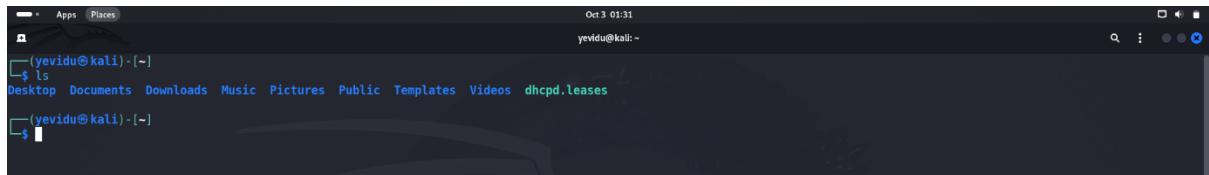
After type password navigate to kali interface.



1.2 Command Line Introduction.

1.2.1 Basic Navigation Commands

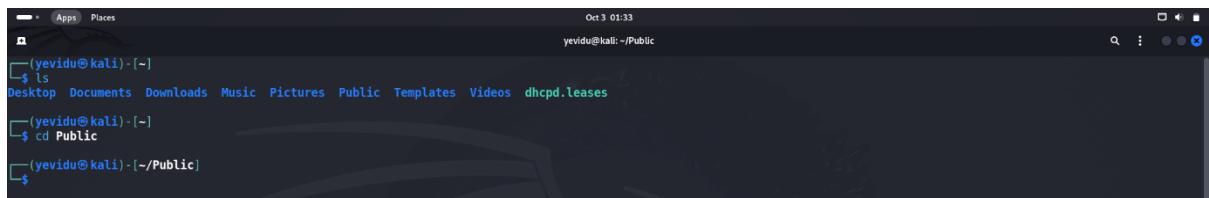
ls : Lists files and directories in the current or specified directory.



```
Oct 3 01:31
yevidu@kali:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos dhcpcd.leases
yevidu@kali:~$
```

A screenshot of a terminal window titled 'Places' at the top. The window shows a dark background with white text. The prompt '(yevidu㉿kali)-[~]' is at the top left. The command 'ls' is entered, followed by a list of files and directories: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos, and dhcpcd.leases. The cursor is at the bottom of the list.

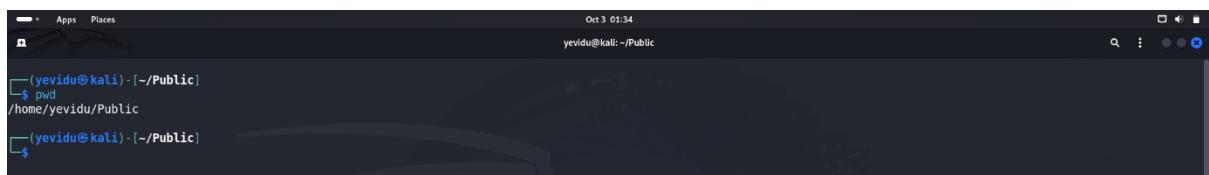
cd : Changes the active working directory.



```
Oct 3 01:33
yevidu@kali:~/Public$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos dhcpcd.leases
yevidu@kali:~/Public$ cd Public
yevidu@kali:~/Public$
```

A screenshot of a terminal window titled 'Places'. The prompt '(yevidu㉿kali)-[~]' is at the top left. The user runs 'ls' to show files in the current directory. Then, they enter 'cd Public', changing the directory to ~/Public. The prompt now shows the new directory path: '~/Public'.

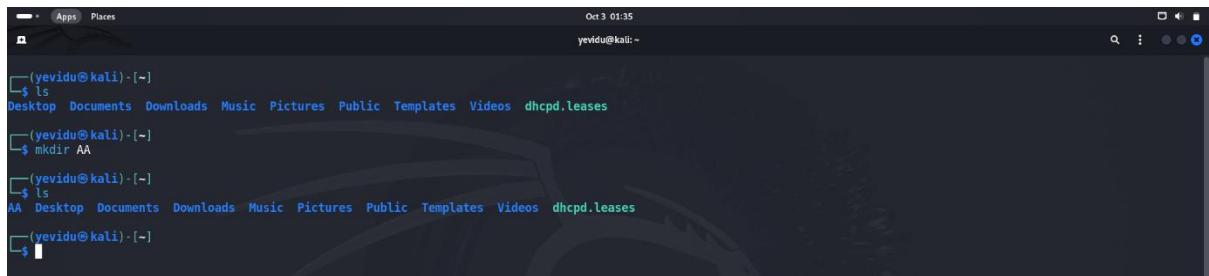
pwd : Displays the full path of the current directory.



```
Oct 3 01:34
yevidu@kali:~/Public$ pwd
/home/yevidu/Public
yevidu@kali:~/Public$
```

A screenshot of a terminal window titled 'Places'. The prompt '(yevidu㉿kali)-[~/Public]' is at the top left. The user runs 'pwd', which outputs the full path '/home/yevidu/Public'. The cursor is at the end of the path.

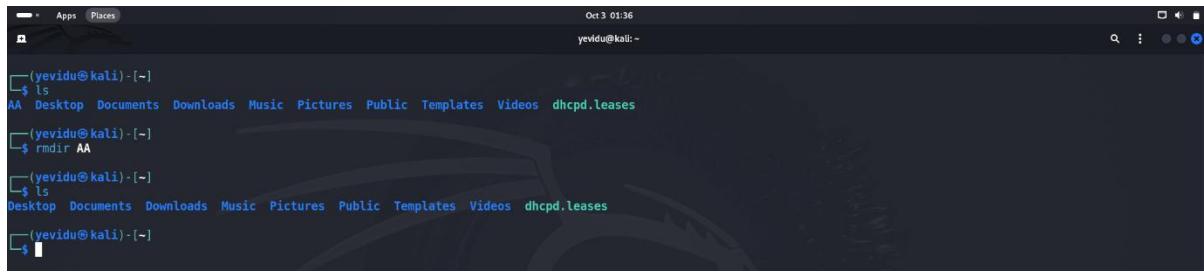
mkdir : Creates a new directory.



```
Oct 3 01:35
yevidu@kali:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos dhcpcd.leases
yevidu@kali:~$ mkdir AA
yevidu@kali:~$ ls
AA Desktop Documents Downloads Music Pictures Public Templates Videos dhcpcd.leases
yevidu@kali:~$
```

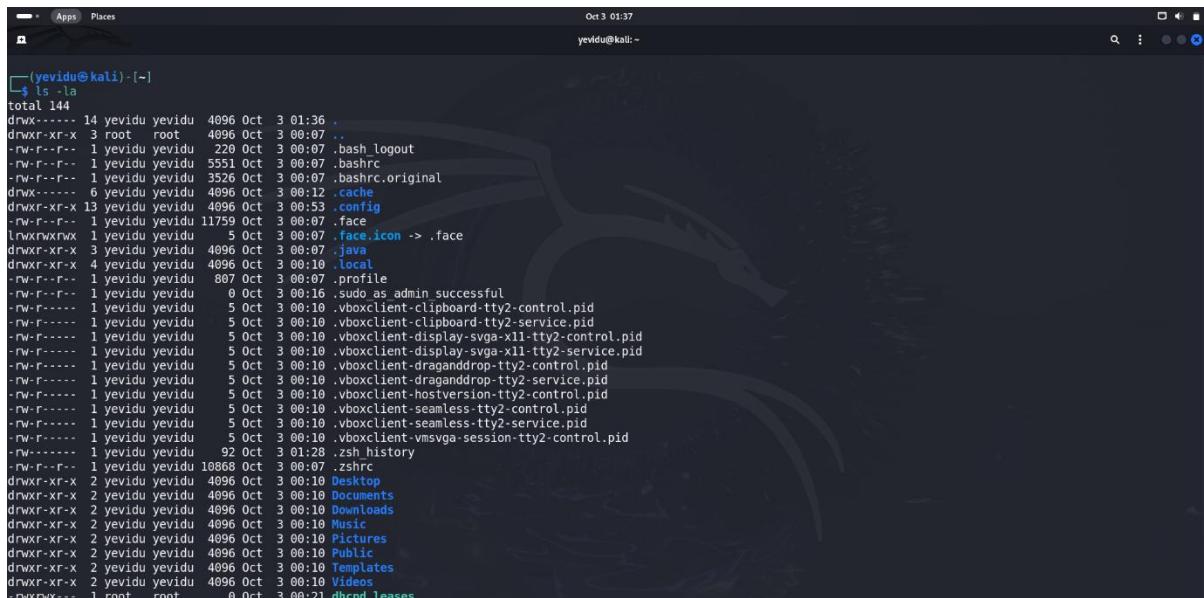
A screenshot of a terminal window titled 'Places'. The prompt '(yevidu㉿kali)-[~]' is at the top left. The user runs 'ls' to see the current directory contents. They then run 'mkdir AA', creating a new directory named 'AA'. Finally, they run 'ls' again to verify that the new directory 'AA' is present.

rmkdir : Removes an empty directory.



```
(yevidu㉿kali)-[~]
$ ls
AA Desktop Documents Downloads Music Pictures Public Templates Videos dhcpcd.leases
(yevidu㉿kali)-[~]
$ rmmdir AA
(yevidu㉿kali)-[~]
$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos dhcpcd.leases
(yevidu㉿kali)-[~]
```

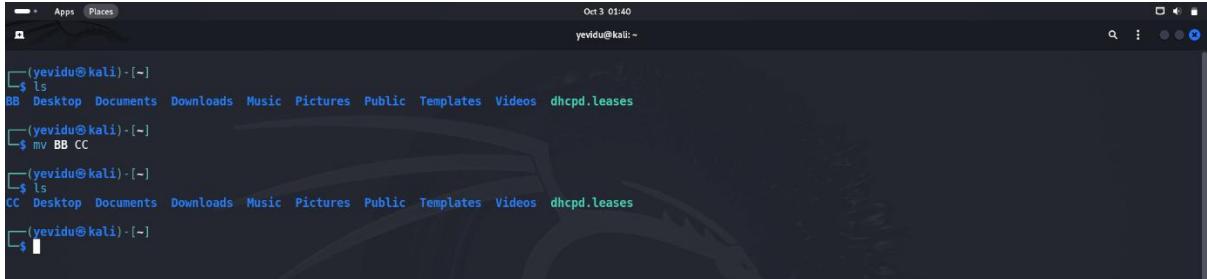
ls -la : provides detailed information about each file or directory.



```
(yevidu㉿kali)-[~]
$ ls -la
total 144
drwx----- 14 yevidu yevidu 4096 Oct  3 01:36 .
drwxr-xr-x  3 root  root  4096 Oct  3 00:07 ..
-rw-r--r--  1 yevidu yevidu   220 Oct  3 00:07 .bash_logout
-rw-r--r--  1 yevidu yevidu  5551 Oct  3 00:07 .bashrc
-rw-r--r--  1 yevidu yevidu 3526 Oct  3 00:07 .bashrc.original
drwx-----  6 yevidu yevidu 4096 Oct  3 00:12 .cache
drwxr-xr-x 13 yevidu yevidu 4096 Oct  3 00:53 .config
-rw-r--r--  1 yevidu yevidu 11759 Oct  3 00:07 .face
l-rwxrwxrwx  1 yevidu yevidu     5 Oct  3 00:07 .face.icon -> .face
drwxr-xr-x  3 yevidu yevidu 4096 Oct  3 00:07 .java
drwxr-xr-x  4 yevidu yevidu 4096 Oct  3 00:10 .local
-rw-r--r--  1 yevidu yevidu   807 Oct  3 00:07 .profile
-rw-r--r--  1 yevidu yevidu     0 Oct  3 00:16 .suds_as_admin_successful
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-clipboard-tty2-control.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-clipboard-tty2-service.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-display-svga-x11-tty2-control.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-display-svga-x11-tty2-service.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-draganddrop-tty2-control.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-draganddrop-tty2-service.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-hostversion-tty2-control.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-seamless-tty2-control.pid
-rw-r-----  1 yevidu yevidu    5 Oct  3 00:10 .vboxclient-seamless-tty2-service.pid
-rw-r-----  1 yevidu yevidu   92 Oct  3 01:28 .zsh_history
-rw-r--r--  1 yevidu yevidu 10868 Oct  3 00:07 .zshrc
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Desktop
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Documents
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Downloads
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Music
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Pictures
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Public
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Templates
drwxr-xr-X  2 yevidu yevidu 4096 Oct  3 00:10 Videos
drwxrwx---  1 root  root    0 Oct  3 00:21 dhcpcd.leases
```

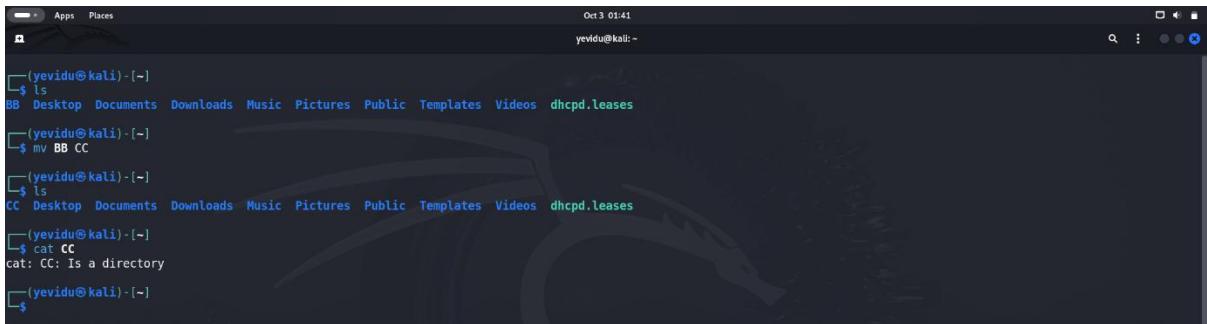
1.2.2 File Manipulation Commands

mv : Moves or renames files or directories.



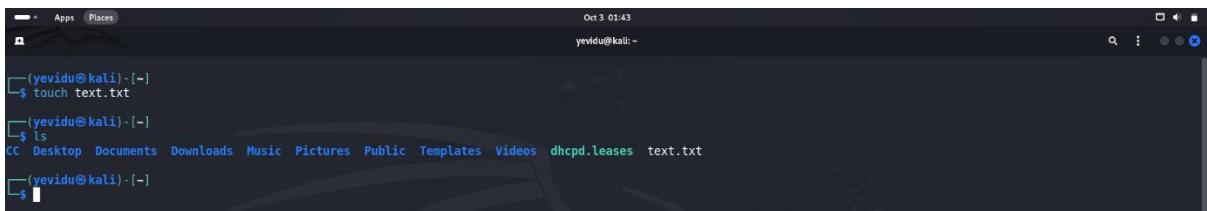
```
(yevidu㉿kali)-[~]
$ ls
BB Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases
(yevidu㉿kali)-[~]
$ mv BB CC
(yevidu㉿kali)-[~]
$ ls
CC Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases
(yevidu㉿kali)-[~]
$
```

cat : Displays the content of a file.



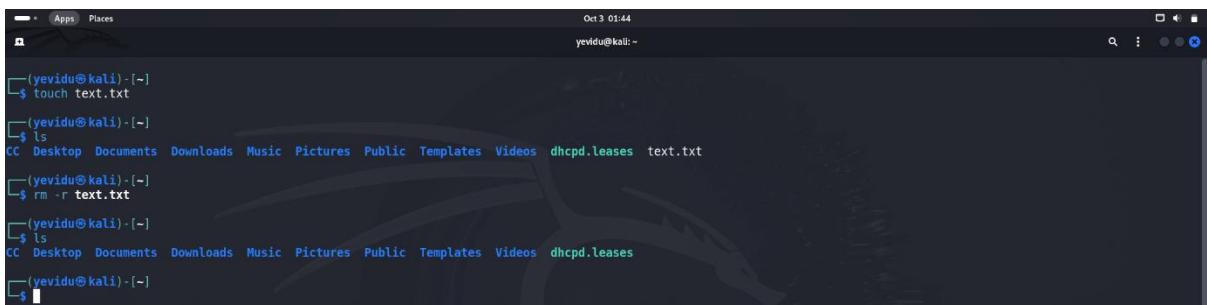
```
(yevidu㉿kali)-[~]
$ ls
BB Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases
(yevidu㉿kali)-[~]
$ mv BB CC
(yevidu㉿kali)-[~]
$ ls
CC Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases
(yevidu㉿kali)-[~]
$ cat CC
cat: CC: Is a directory
(yevidu㉿kali)-[~]
$
```

touch : Creates a new, empty file.



```
(yevidu㉿kali)-[~]
$ touch text.txt
(yevidu㉿kali)-[~]
$ ls
CC Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases text.txt
(yevidu㉿kali)-[~]
$
```

rm -r : Removes files or directories.

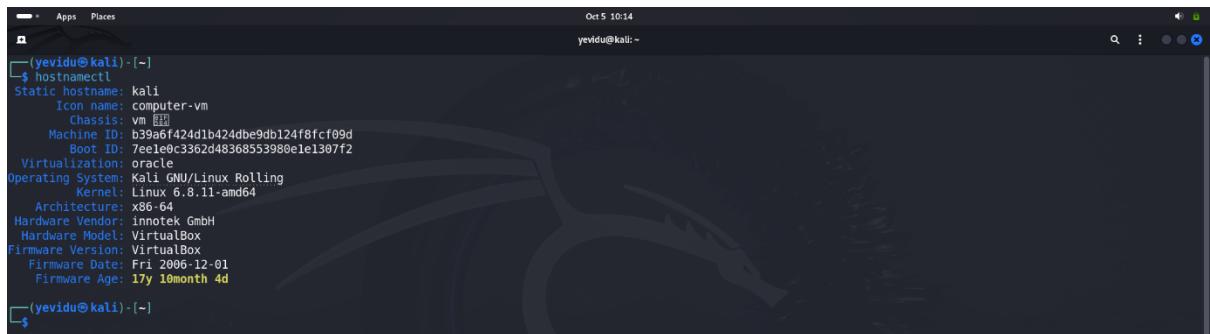


```
(yevidu㉿kali)-[~]
$ touch text.txt
(yevidu㉿kali)-[~]
$ ls
CC Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases text.txt
(yevidu㉿kali)-[~]
$ rm -r text.txt
(yevidu㉿kali)-[~]
$ ls
CC Desktop Documents Downloads Music Pictures Public Templates Videos dhcpd.leases
(yevidu㉿kali)-[~]
$
```

1.3 System Information and User Management.

1.3.1 System Information Commands.

hostnamectl : Display system hostname and OS details with kernel version and architecture.



```
(yevidu㉿kali)-[~]
$ hostnamectl
Static hostname: kali
  Icon name: computer-vm
    Chassis: vm 
  Machine ID: b39af424d1b424dbe9db124f8fcf09d
    Boot ID: 7ec1e0c3362d48368553980e1e1307f2
  Virtualization: oracle
Operating system: Kali GNU/Linux Rolling
  Kernel: Linux 6.8.11-amd64
  Architecture: x86_64
Hardware Vendor: innotek GmbH
Hardware Model: VirtualBox
Firmware Version: VirtualBox
  Firmware Date: Fri, 2006-12-01
  Firmware Age: 17y 10month 4d
(yevidu㉿kali)-[~]
```

lscpu : Provide details about CPU architecture.



```
(yevidu㉿kali)-[~]
$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         39 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                3
On-line CPU(s) list:  0-2
Vendor ID:             GenuineIntel
Model name:            Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz
CPU family:            6
Model:                 142
Thread(s) per core:   1
Core(s) per socket:   3
Socket(s):             1
Stepping:              10
BogoMIPS:              3600.00
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 cx16 pcid sse4_1 sse4_2 movbe popcnt aes rdrand hypervisor lahf_lm abm 3dnowprefetch
Virtualization features:
  Hypervisor vendor:   KVM
  Virtualization type: full
Caches (sum of all):
  L1d:                 96 KiB (3 instances)
  L1i:                 96 KiB (3 instances)
  L2:                  768 KiB (3 instances)
  L3:                  18 MiB (3 instances)
NUMA:
  NUMA node(s):        1
  NUMA node0 CPU(s):   0-2
Vulnerabilities:
  Gather data sampling: Not affected
  Itlb multihit:       Not affected
  L1tf:                Mitigation; PTE Inversion
  Mds:                 Mitigation; Clear CPU buffers; SMT Host state unknown
  Meltdown:            Mitigation; PTT
```

df -h : Show disk usage.



```
(yevidu㉿kali)-[~]
$ df -h
Filesystem      Size   Used  Avail Use% Mounted on
udev            3.2G     0  3.2G  0% /dev
tmpfs           644M  1.2M 643M  1% /run
/dev/sdal        19G  14G  4.0G  78% /
tmpfs           3.2G     0  3.2G  0% /dev/shm
tmpfs           5.0M     0  5.0M  0% /run/lock
tmpfs           1.0M     0  1.0M  0% /run/credentials/systemd-journal.service
tmpfs           1.0M     0  1.0M  0% /run/credentials/systemd-udev-load-credentials.service
tmpfs           1.0M     0  1.0M  0% /run/credentials/systemd-tmpfiles-setup-dev-early.service
tmpfs           1.0M     0  1.0M  0% /run/credentials/systemd-sysctl.service
tmpfs           1.0M     0  1.0M  0% /run/credentials/systemd-sysusers.service
tmpfs           3.2G     0  3.2G  0% /tmp
tmpfs           1.0M     0  1.0M  0% /run/credentials/systemd-tmpfiles-setup-dev.service
tmpfs           644M 144K 644M  1% /run/user/1000

```

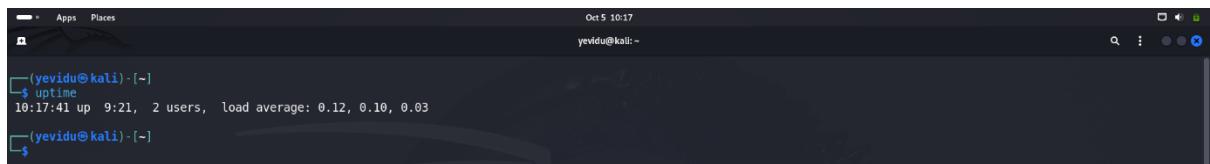
uname -a : Display kernel version and other information.



```
(yevidu㉿kali)-[~]
$ uname -a
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kaliz (2024-05-30) x86_64 GNU/Linux

```

uptime : Show how long the system has been running with load average.



```
(yevidu㉿kali)-[~]
$ uptime
10:17:41 up  9:21,  2 users,  load average: 0.12, 0.10, 0.03

```

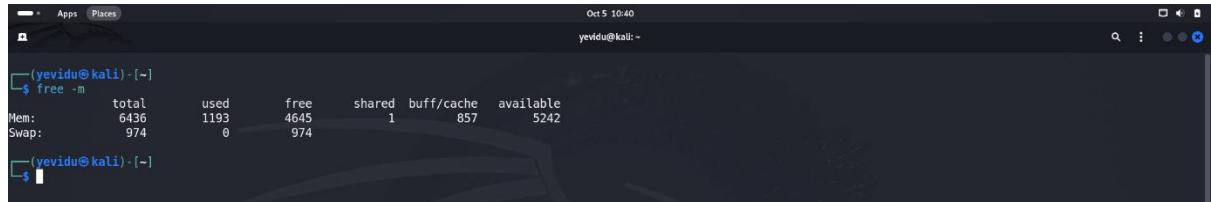
ip -a : Show detailed network interface information including IP addresses.



```
(yevidu㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 0.0.0.0 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 brd 0.0.0.0 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b5:39:da brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.216/24 brd 192.168.42.255 scope global dynamic noprefixroute eth0
        valid_lft 3990sec preferred_lft 3590sec
    inet6 fe80::a00:27ff:feb5:39da/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

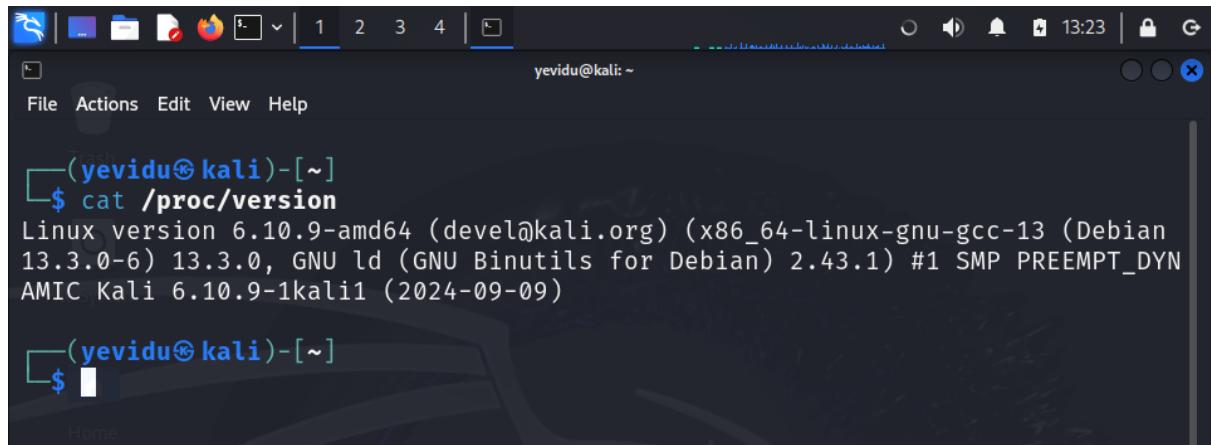
free -m : Display total memory, used memory, free memory, shared memory, buffers, and cache.



```
(yevidu㉿kali)-[~]
$ free -m
total        used        free      shared  buff/cache   available
Mem:       6436        1193       4645           1        857       5242
Swap:      974          0         974

(yevidu㉿kali)-[~]
```

cat /proc/version : outputs detailed information about the Linux kernel version and system build.



```
(yevidu㉿kali)-[~]
$ cat /proc/version
Linux version 6.10.9-amd64 (devel@kali.org) (x86_64-linux-gnu-gcc-13 (Debian 13.3.0-6) 13.3.0, GNU ld (GNU Binutils for Debian) 2.43.1) #1 SMP PREEMPT_DYNAMIC Kali 6.10.9-1kali1 (2024-09-09)

(yevidu㉿kali)-[~]
```

1.3.2 User Management Commands.

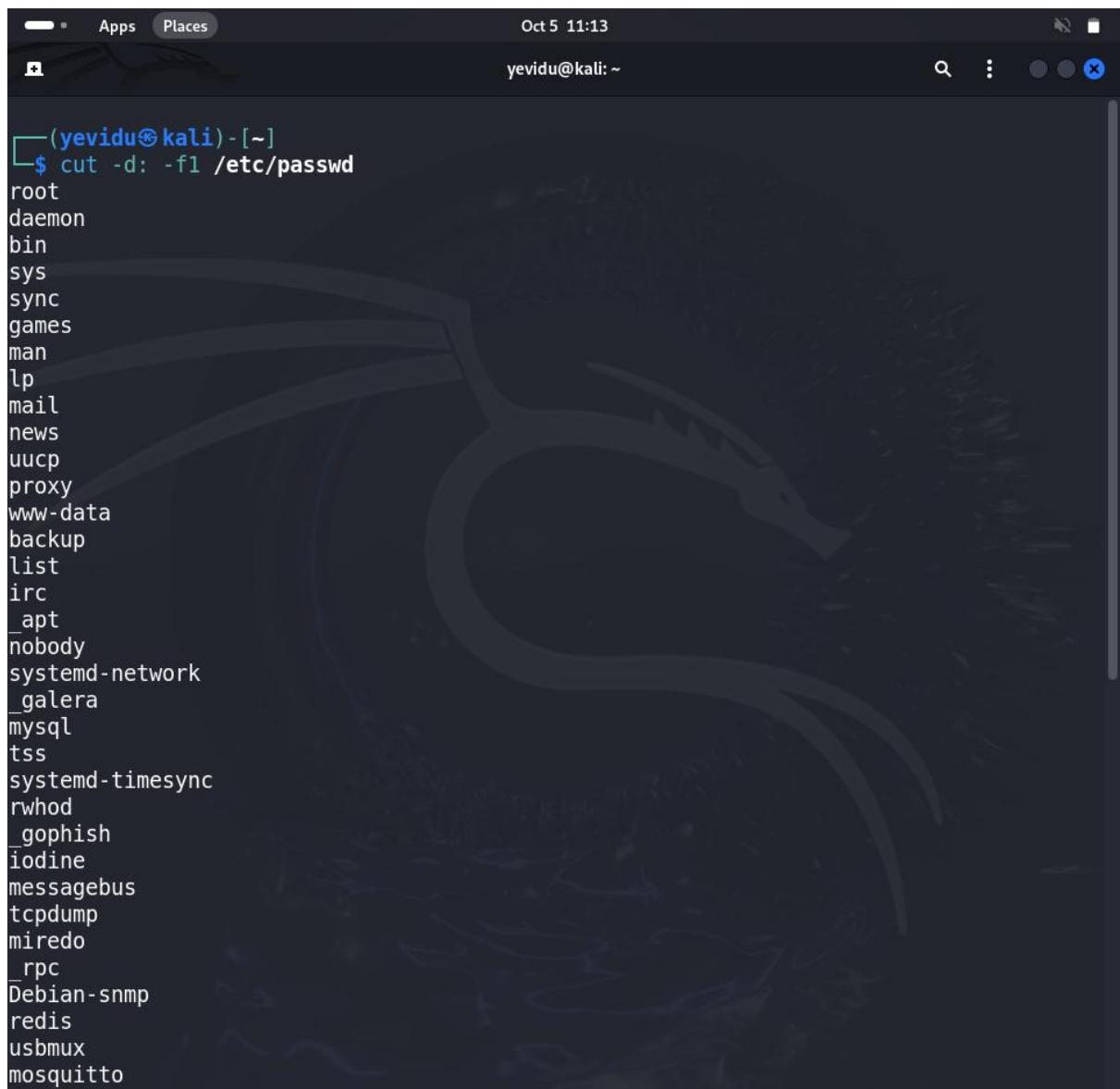
sudo adduser username : Add new user.

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title bar says '(yevidu㉿kali)-[~]'. The window title is 'Terminal'. The status bar at the top right shows the date and time as 'Oct 5 11:04' and the user as 'yevidu@kali: ~'. The terminal content is as follows:

```
(yevidu㉿kali)-[~]
$ sudo adduser yashmika
info: Adding user `yashmika' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `yashmika' (1002) ...
info: Adding new user `yashmika' (1002) with group `yashmika (1002)' ...
info: Creating home directory `/home/yashmika' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for yashmika
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []
Is the information correct? [Y/n] y
info: Adding new user `yashmika' to supplemental / extra groups `users' ...
info: Adding user `yashmika' to group `users' ...

(yevidu㉿kali)-[~]
```

cut -d: -f1 /etc/passwd : View list of all users.



```
(yevidu㉿kali)-[~]
$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
_apt
nobody
systemd-network
_galera
mysql
tss
systemd-timesync
rwhod
_gophish
iodine
messagebus
tcpdump
miredo
_rpc
Debian-snmp
redis
usbmux
mosquitto
```

```
Oct 5 11:13  
yevidu@kali: ~  
  
rwhod  
_gophish  
iodine  
messagebus  
tcpdump  
miredo  
_rpc  
Debian-snmp  
redis  
usbmux  
mosquitto  
redsocks  
stunnel4  
sshd  
dnsmasq  
sslh  
postgres  
avahi  
_gvm  
speech-dispatcher  
fwupd-refresh  
inetsim  
geoclue  
gnome-remote-desktop  
statd  
saned  
polkitd  
rtkit  
colord  
Debian-gdm  
yevidu  
newuser  
bind  
yashmika  
  
└─(yevidu㉿kali)-[~]  
└─$
```

sudo deluser username : Delete a user.

```
Oct 5 11:14  
yevidu@kali: ~  
  
└─(yevidu㉿kali)-[~]  
└─$ sudo deluser yashmika  
info: Removing crontab ...  
info: Removing user `yashmika' ...  
  
└─(yevidu㉿kali)-[~]  
└─$
```

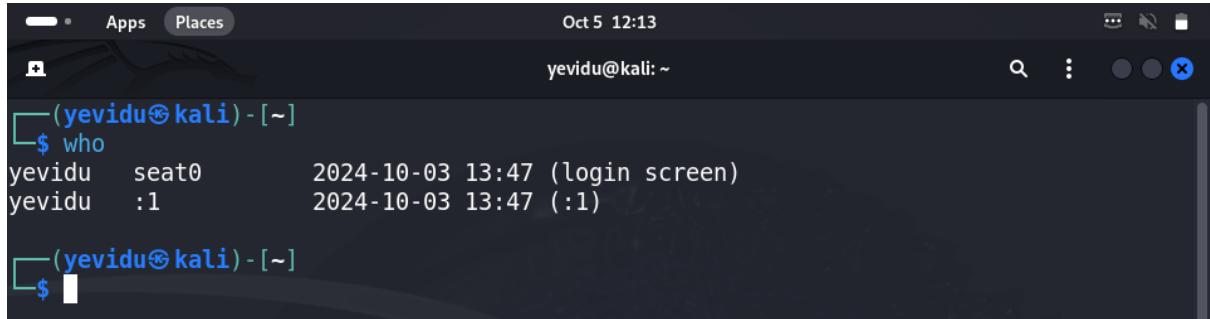
id : view a user's details.



```
(yevidu㉿kali)-[~]
$ id yevidu
uid=1000(yevidu) gid=1000(yevidu) groups=1000(yevidu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),117(bluetooth),121(wireshark),126(scanner),132(vboxsf),133(kaboxer)

(yevidu㉿kali)-[~]
$
```

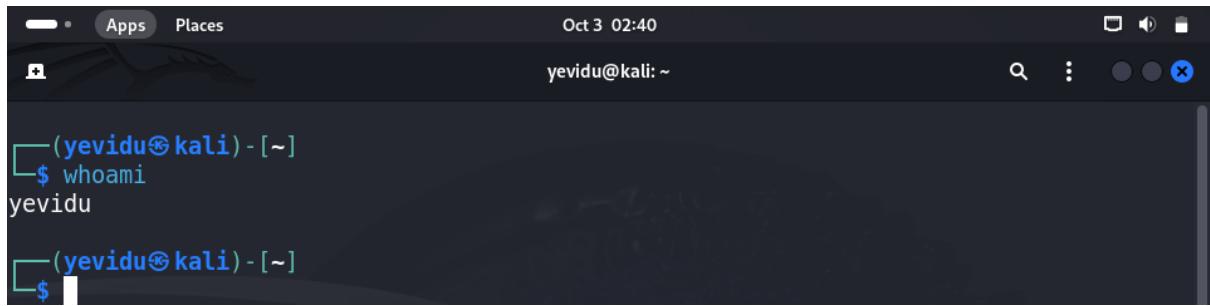
who : Display users logged in to the system.



```
(yevidu㉿kali)-[~]
$ who
yevidu  seat0      2024-10-03 13:47 (login screen)
yevidu  :1        2024-10-03 13:47 (:1)

(yevidu㉿kali)-[~]
$
```

whoami : Display current user log in to the system.



```
(yevidu㉿kali)-[~]
$ whoami
yevidu

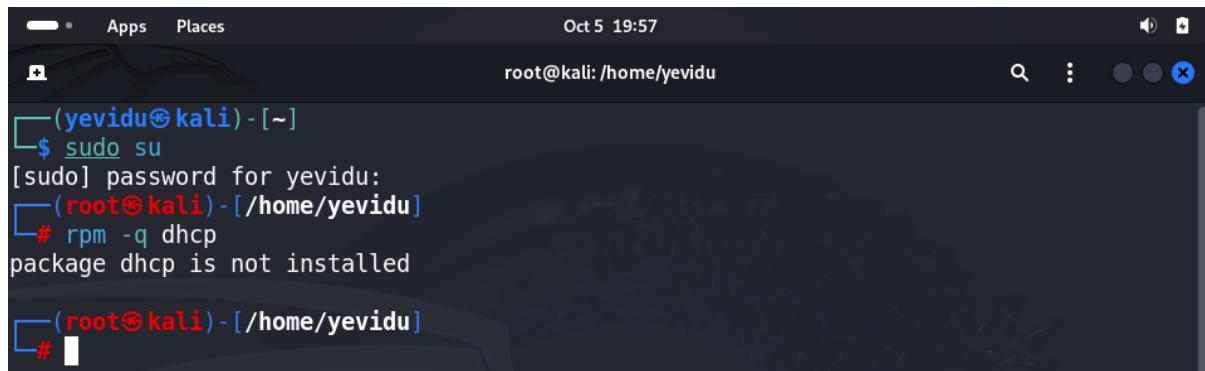
(yevidu㉿kali)-[~]
$
```

2. DHCP, DNS and NTP Services.

2.1 DHCP (Domain Host Configuration Management).

First check DHCP is already installed in the Linux.

Command **rpm -q dhcp**.



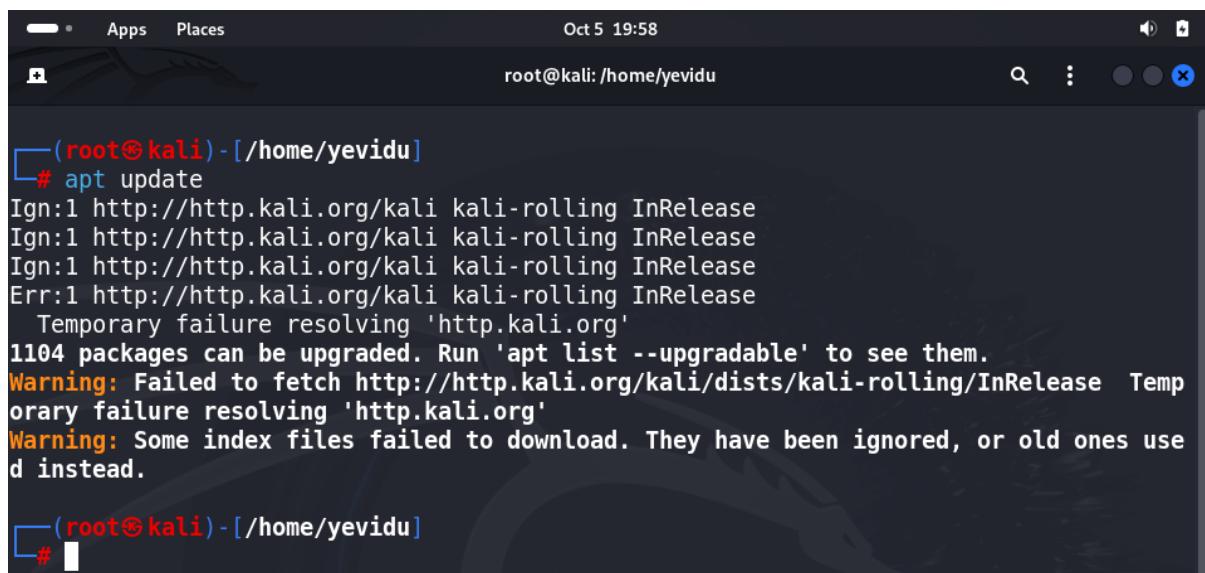
```
(yevidu㉿kali)-[~]
$ sudo su
[sudo] password for yevidu:
(root㉿kali)-[/home/yevidu]
# rpm -q dhcp
package dhcp is not installed

#
```

A screenshot of a terminal window on a Kali Linux desktop. The title bar says "root@kali:/home/yevidu". The terminal shows the user switching to root using "sudo su", then running the command "rpm -q dhcp". The output indicates that the package "dhcp" is not installed. The prompt then changes back to the regular user level with "#".

It does not install. So, we update the package.

Command **apt update**.



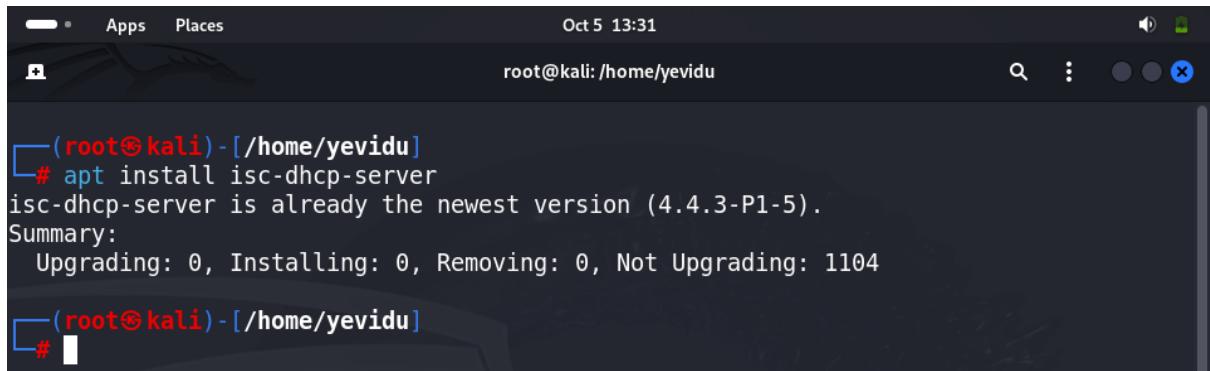
```
(root㉿kali)-[/home/yevidu]
# apt update
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Err:1 http://http.kali.org/kali kali-rolling InRelease
  Temporary failure resolving 'http.kali.org'
1104 packages can be upgraded. Run 'apt list --upgradable' to see them.
Warning: Failed to fetch http://http.kali.org/kali/dists/kali-rolling/InRelease  Temp
orary failure resolving 'http.kali.org'
Warning: Some index files failed to download. They have been ignored, or old ones use
d instead.

#
```

A screenshot of a terminal window on a Kali Linux desktop. The title bar says "root@kali:/home/yevidu". The terminal shows the user running "apt update". It lists several Ign:1 entries for the "kali-rolling" repository at "http://http.kali.org/kali" and one Err:1 entry for the same repository, both indicating a "Temporary failure resolving 'http.kali.org'". It also displays a warning about failed index file downloads. The prompt then changes back to the regular user level with "#".

Then we install the DHCP server.

Command **apt install isc-dhcp-server**.



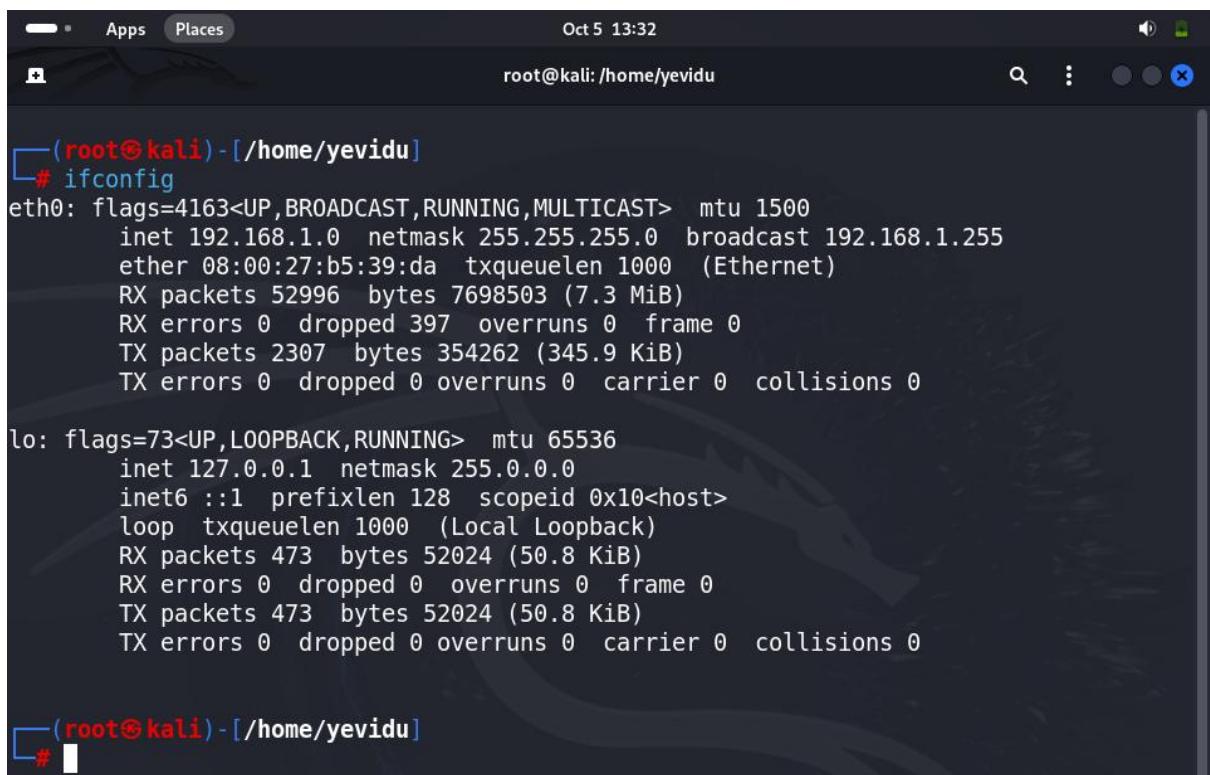
A screenshot of a terminal window on a Kali Linux system. The title bar shows 'Oct 5 13:31' and 'root@kali: /home/yevidu'. The terminal prompt is '(root㉿kali)-[~/home/yevidu]'. The user runs the command '# apt install isc-dhcp-server'. The output indicates that 'isc-dhcp-server' is already the newest version (4.4.3-P1-5). A summary table shows 0 upgrades, 0 installs, 0 removes, and 1104 not upgrading. The terminal ends with a blank line and a cursor.

```
(root㉿kali)-[~/home/yevidu]
# apt install isc-dhcp-server
isc-dhcp-server is already the newest version (4.4.3-P1-5).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1104

(root㉿kali)-[~/home/yevidu]
# 
```

Check IP address.

Command **ifconfig**.



A screenshot of a terminal window on a Kali Linux system. The title bar shows 'Oct 5 13:32' and 'root@kali: /home/yevidu'. The terminal prompt is '(root㉿kali)-[~/home/yevidu]'. The user runs the command '# ifconfig'. The output shows details for the 'eth0' and 'lo' interfaces. For 'eth0', flags are UP, BROADCAST, RUNNING, MULTICAST, MTU is 1500, and it has an IP of 192.168.1.0. For 'lo', flags are LOOPBACK, RUNNING, and it has an IP of 127.0.0.1. The terminal ends with a blank line and a cursor.

```
(root㉿kali)-[~/home/yevidu]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.0 netmask 255.255.255.0 broadcast 192.168.1.255
        ether 08:00:27:b5:39:da txqueuelen 1000 (Ethernet)
          RX packets 52996 bytes 7698503 (7.3 MiB)
          RX errors 0 dropped 397 overruns 0 frame 0
          TX packets 2307 bytes 354262 (345.9 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 473 bytes 52024 (50.8 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 473 bytes 52024 (50.8 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(root㉿kali)-[~/home/yevidu]
# 
```

We need to edit interfaces.txt file.

```
(root㉿kali)-[~/home/yevidu]
└─# cd /etc

└─(root㉿kali)-[/etc]
└─# cd network

└─(root㉿kali)-[/etc/network]
└─# cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

└─(root㉿kali)-[/etc/network]
└─#
```

Then open interfaces.txt

Command **nano interfaces**.

```
(root㉿kali)-[/etc/network]
└─# nano interfaces
```

Then edit interfaces file.

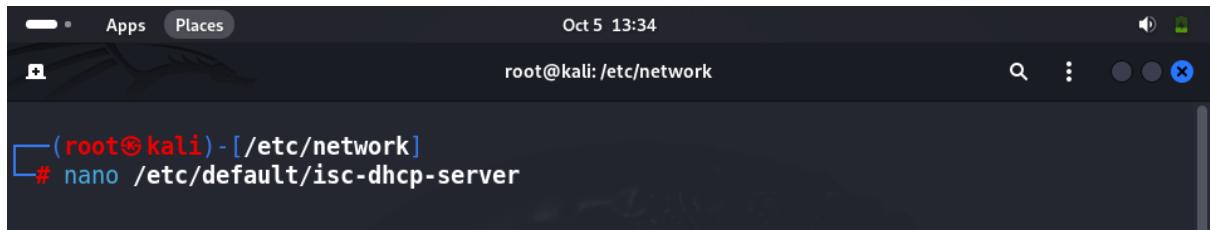
```
GNU nano 8.1           interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
```

Open isc-dhcp-server file.

Command **nano /etc/default/isc-dhcp-server**.



```
Oct 5 13:34
root@kali:/etc/network
(boot㉿kali)-[~/etc/network]
# nano /etc/default/isc-dhcp-server
```

Edit the INTERFACESv4 = " eth0".



```
Oct 5 13:35
root@kali:/etc/network
GNU nano 8.1          /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

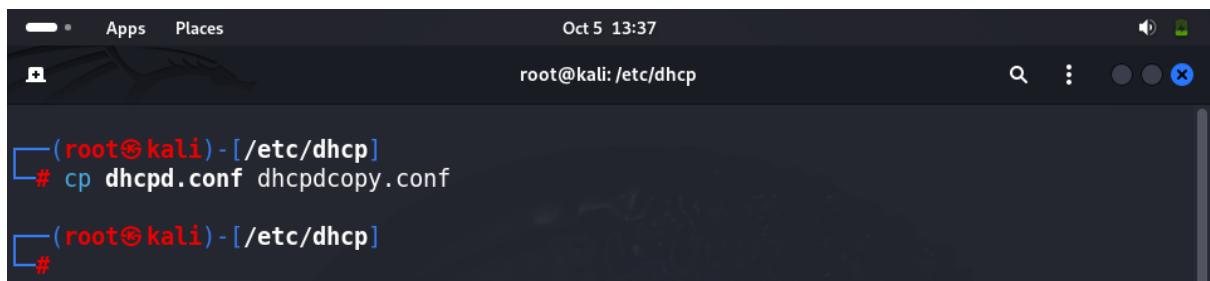
# Path to dhcpcd's config file (default: /etc/dhcp/dhcpcd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpcd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpcd6.conf

# Path to dhcpcd's PID file (default: /var/run/dhcpcd.pid).
#DHCPDv4_PID=/var/run/dhcpcd.pid
#DHCPDv6_PID=/var/run/dhcpcd6.pid

# Additional options to start dhcpcd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpcd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="eth0"
INTERFACESv6=""
```

Make backup of dhcp.conf file.

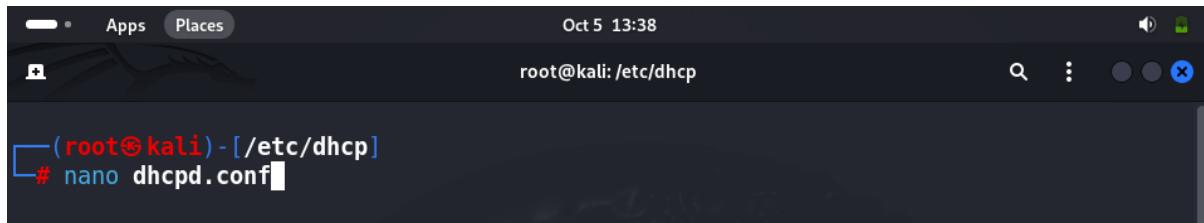


```
Oct 5 13:37
root@kali:/etc/dhcp
(boot㉿kali)-[~/etc/dhcp]
# cp dhcpcd.conf dhcpcdcopy.conf

(boot㉿kali)-[~/etc/dhcp]
#
```

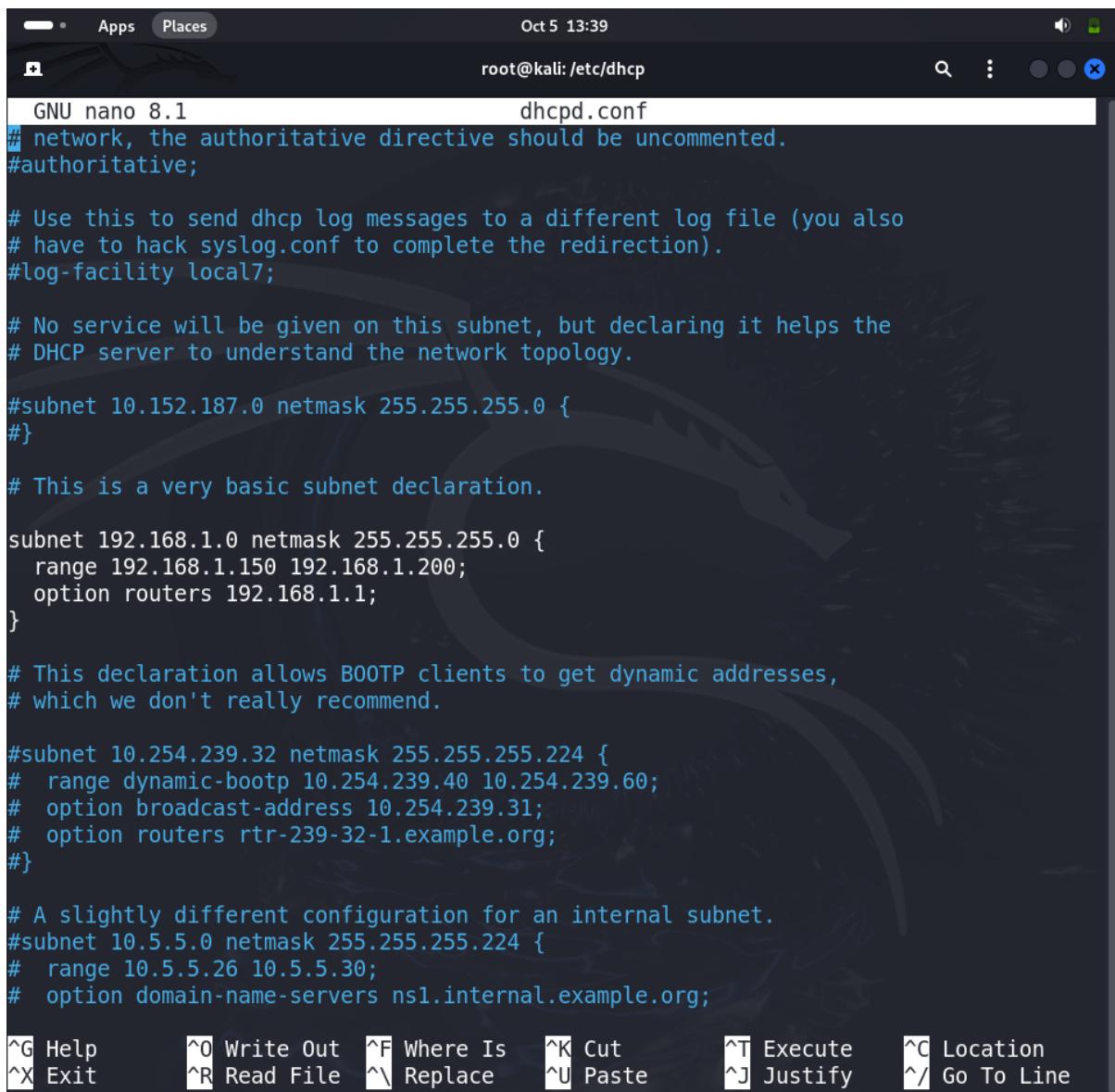
Open dhcp.conf file.

Command **nano dhcpd.conf**.



```
Oct 5 13:38
root@kali: /etc/dhcp
# nano dhcpd.conf
```

Edit dhcp.conf file.



```
GNU nano 8.1          dhcpd.conf
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {
#}

# This is a very basic subnet declaration.

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.150 192.168.1.200;
    option routers 192.168.1.1;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#    range dynamic-bootp 10.254.239.40 10.254.239.60;
#    option broadcast-address 10.254.239.31;
#    option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
#subnet 10.5.5.0 netmask 255.255.255.224 {
#    range 10.5.5.26 10.5.5.30;
#    option domain-name-servers ns1.internal.example.org;
```

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^M Replace ^U Paste ^J Justify ^/ Go To Line

Start the dhcp server.

```
Oct 5 13:40 root@kali:/ [root@kali]# nano /etc/dhcp/dhcpd.conf
[...]
Oct 5 13:40 root@kali:/ [root@kali]# cd ..
[...]
Oct 5 13:40 root@kali:/ [root@kali]# /etc/init.d/isc-dhcp-server start
Starting isc-dhcp-server (via systemctl): isc-dhcp-server.service.
[...]
Oct 5 13:40 root@kali:/ [root@kali]#
```

Now let's check the status to confirm whether DHCP is working properly.

Command **systemctl status isc-dhcp-server.service**.

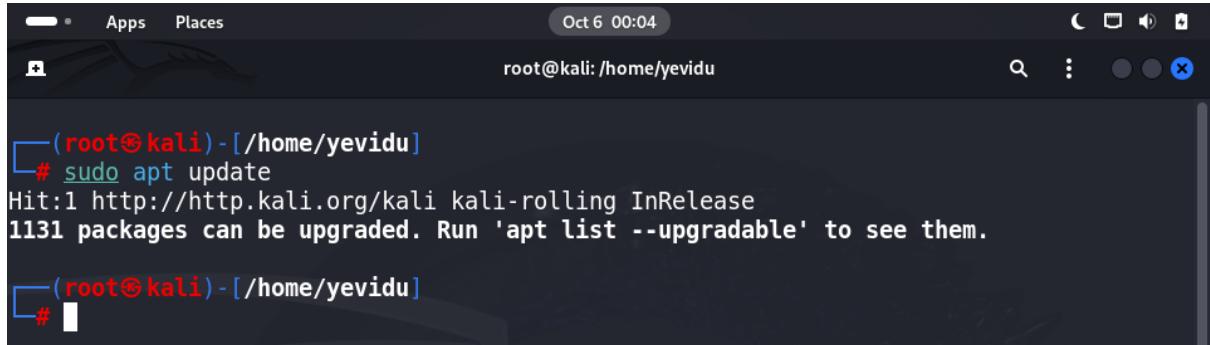
```
Oct 5 13:40 root@kali:/ [root@kali]# systemctl status isc-dhcp-server.service
● isc-dhcp-server.service - LSB: DHCP server
  Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
  Active: active (running) since Sat 2024-10-05 13:21:50 +0530; 18min ago
    Invocation: f62401d67cccd4556a908d4436bff5968
      Docs: man:systemd-sysv-generator(8)
   Process: 8594 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0>
     Tasks: 1 (limit: 7623)
    Memory: 4.2M (peak: 6.3M)
      CPU: 323ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─8607 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf eth0

Oct 05 13:33:22 kali dhcpcd[8607]: DHCPOffer on 192.168.1.150 to 08:00:27:b5:39:da (k>
Oct 05 13:33:22 kali dhcpcd[8607]: reuse lease: lease age 71 (secs) under 25% thresho>
Oct 05 13:33:22 kali dhcpcd[8607]: DHCPREQUEST for 192.168.1.150 (192.168.1.0) from 0>
Oct 05 13:33:22 kali dhcpcd[8607]: DHCPACK on 192.168.1.150 to 08:00:27:b5:39:da (kal>
Oct 05 13:33:22 kali dhcpcd[8607]: reuse lease: lease age 71 (secs) under 25% thresho>
Oct 05 13:33:22 kali dhcpcd[8607]: DHCPDISCOVER from 08:00:27:b5:39:da (kali) via eth0
Oct 05 13:33:22 kali dhcpcd[8607]: DHCPOffer on 192.168.1.150 to 08:00:27:b5:39:da (k>
Oct 05 13:33:22 kali dhcpcd[8607]: reuse lease: lease age 71 (secs) under 25% thresho>
Oct 05 13:33:22 kali dhcpcd[8607]: DHCPREQUEST for 192.168.1.150 (192.168.1.0) from 0>
Oct 05 13:33:22 kali dhcpcd[8607]: DHCPACK on 192.168.1.150 to 08:00:27:b5:39:da (kal>
lines 1-22/22 (END)
```

2.2 DNS (Domain Name System).

Before installing BIND9, ensure your package list is up to date. So 1st update package.

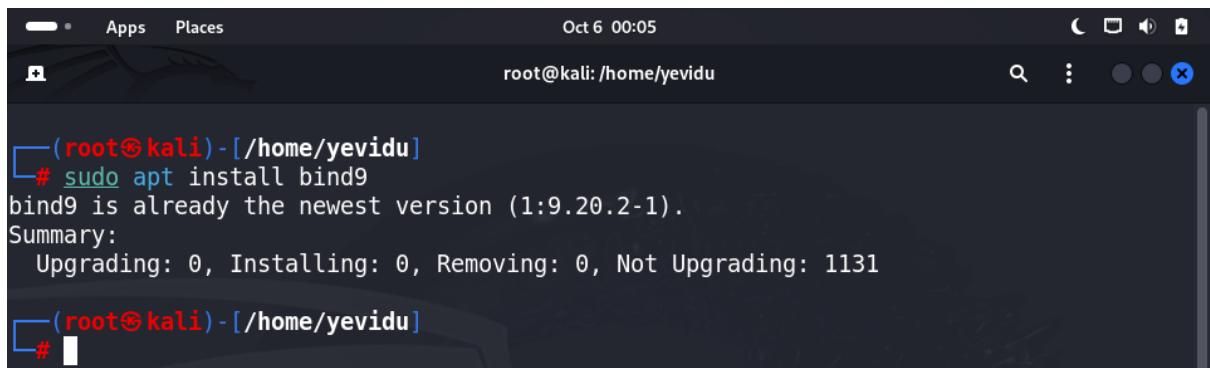
Command **sudo apt update**.



```
Oct 6 00:04
root@kali:/home/yevidu
( root@kali ) - [ /home/yevidu ]
# sudo apt update
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1131 packages can be upgraded. Run 'apt list --upgradable' to see them.
( root@kali ) - [ /home/yevidu ]
#
```

Install BIND9 package.

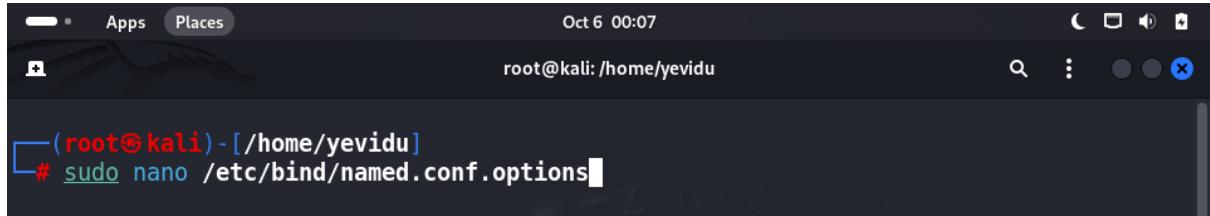
Command **sudo apt install bind9**.



```
Oct 6 00:05
root@kali:/home/yevidu
( root@kali ) - [ /home/yevidu ]
# sudo apt install bind9
bind9 is already the newest version (1:9.20.2-1).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1131
( root@kali ) - [ /home/yevidu ]
#
```

After installation, you need to configure BIND9. So open main configuration file.

Command **sudo nano /etc/bind/named.conf.options**

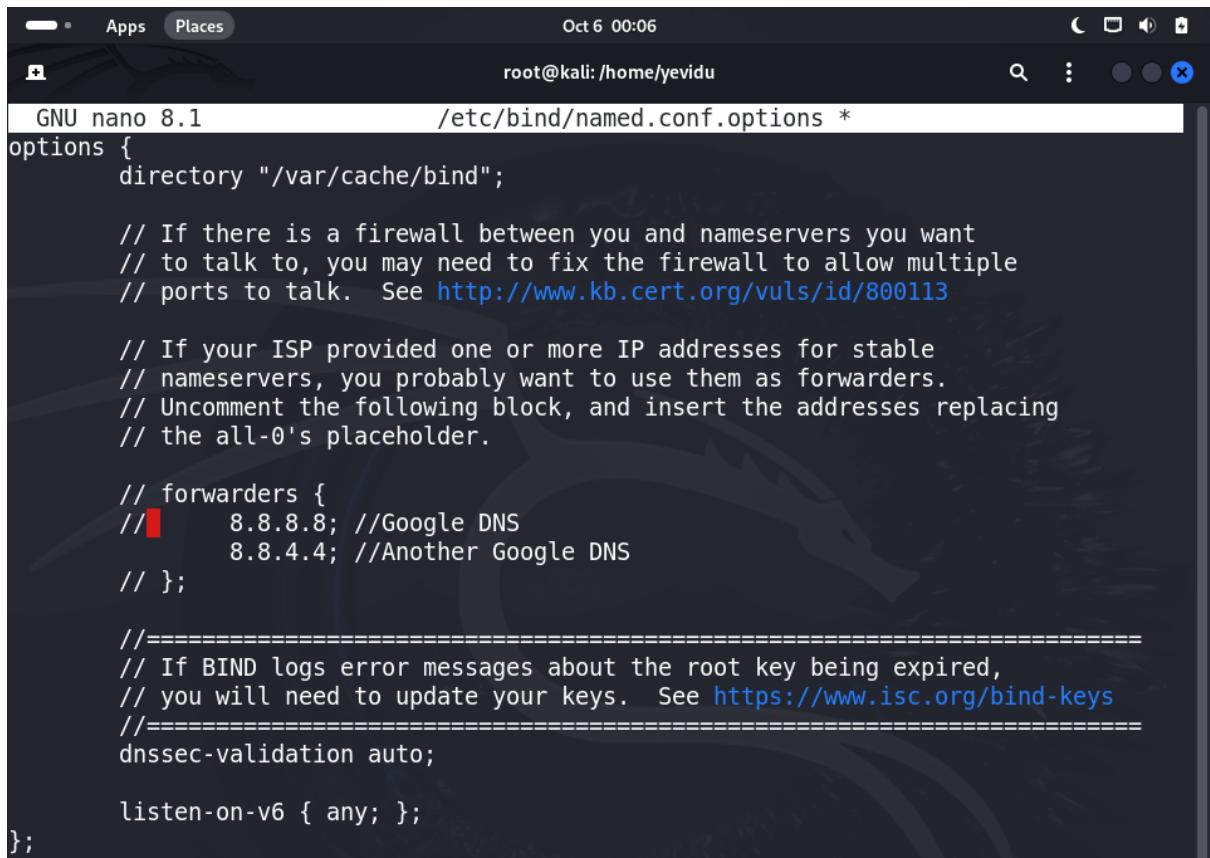


A terminal window titled 'root@kali: /home/yevidu'. The status bar shows 'Oct 6 00:07'. The command '# sudo nano /etc/bind/named.conf.options' is entered in the terminal.

Then edit main configuration file.

// Forwarders

```
forwarders {  
    8.8.8; // Google DNS  
    8.8.4.4; // Google DNS  
};
```

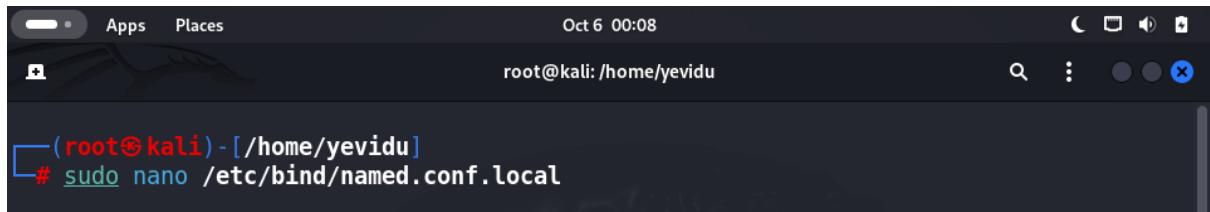


A terminal window titled 'root@kali: /home/yevidu'. The status bar shows 'Oct 6 00:06'. The file '/etc/bind/named.conf.options' is being edited with GNU nano 8.1. The code shown includes the 'forwarders' section with Google DNS addresses, and other BIND9 configuration options like 'dnssec-validation auto' and 'listen-on-v6 { any; }'.

```
GNU nano 8.1          /etc/bind/named.conf.options *  
options {  
    directory "/var/cache/bind";  
  
    // If there is a firewall between you and nameservers you want  
    // to talk to, you may need to fix the firewall to allow multiple  
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113  
  
    // If your ISP provided one or more IP addresses for stable  
    // nameservers, you probably want to use them as forwarders.  
    // Uncomment the following block, and insert the addresses replacing  
    // the all-0's placeholder.  
  
    // forwarders {  
    //     8.8.8; //Google DNS  
    //     8.8.4.4; //Another Google DNS  
    // };  
  
    //=====  
    // If BIND logs error messages about the root key being expired,  
    // you will need to update your keys. See https://www.isc.org/bind-keys  
    //=====  
    dnssec-validation auto;  
  
    listen-on-v6 { any; };  
};
```

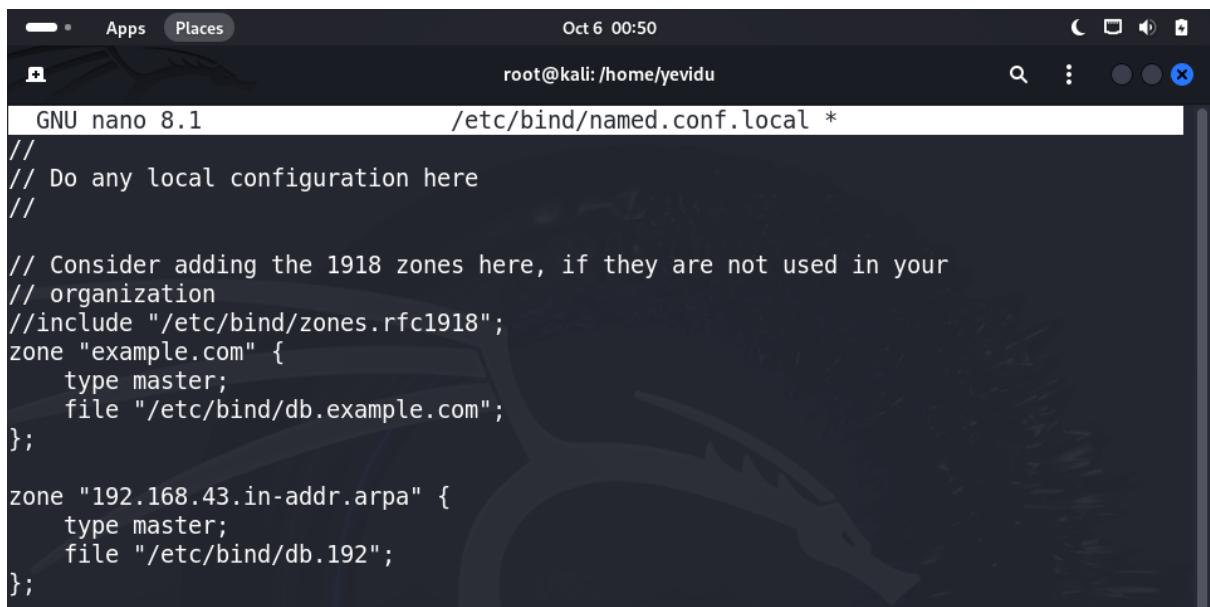
Next, define your zones by editing the named.conf.local file.

Command **sudo nano /etc/bind/named.conf.local**



```
(root㉿kali)-[~/home/yevidu]
# sudo nano /etc/bind/named.conf.local
```

Edit local file.

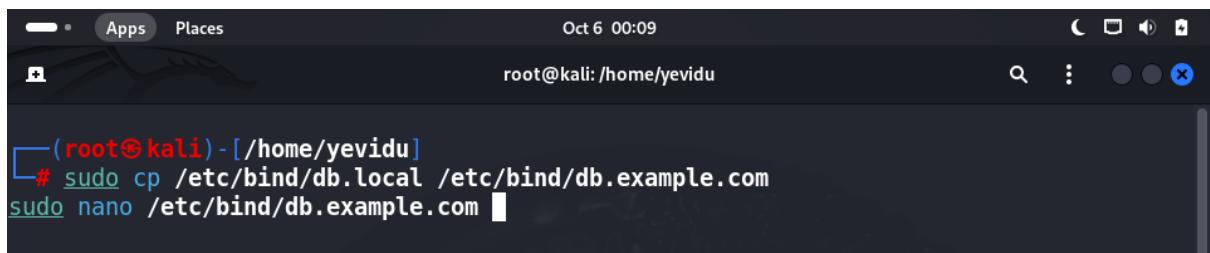


```
GNU nano 8.1          /etc/bind/named.conf.local *
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "example.com" {
    type master;
    file "/etc/bind/db.example.com";
};

zone "192.168.43.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192";
};
```

Create zone files for our domain.

Command **sudo cp /etc/bind/db.empty /etc/bind/db.example.com**



```
(root㉿kali)-[~/home/yevidu]
# sudo cp /etc/bind/db.local /etc/bind/db.example.com
sudo nano /etc/bind/db.example.com
```

Edit new zone file.

```
GNU nano 8.1          /etc/bind/db.example.com *

;
; BIND data file for local loopback interface
;
$TTL    604800
@      IN      SOA    localhost. root.localhost. (
                      2           ; Serial
                      604800       ; Refresh
                      86400        ; Retry
                     2419200      ; Expire
                     604800 )     ; Negative Cache TTL
;
; Name servers
@      IN      NS     ns.example.com.

; A records for hosts
ns    IN      A      192.168.1.10   ; Your DNS server's IP address
www   IN      A      192.168.1.20   ; Your web server's IP address
```

Check whether the BIND9 server running correctly.

Command **sudo systemctl status bind9**

```
Oct 6 00:03
root@kali:/home/yevidu
( root@kali ) - [ /home/yevidu ]
# sudo systemctl status named
● named.service - BIND Domain Name Server
  Loaded: loaded (/usr/lib/systemd/system/named.service; disabled; preset: disabled)
  Active: active (running) since Sun 2024-10-06 00:02:05 +0530; 1min 9s ago
    Invocation: 7ea33bbc498e45a9a5de1200732616e1
      Docs: man:named(8)
    Main PID: 5336 (named)
      Status: "running"
        Tasks: 14 (limit: 7623)
       Memory: 36M (peak: 38.1M)
         CPU: 86ms
      CGroup: /system.slice/named.service
              └─5336 /usr/sbin/named -f -u bind

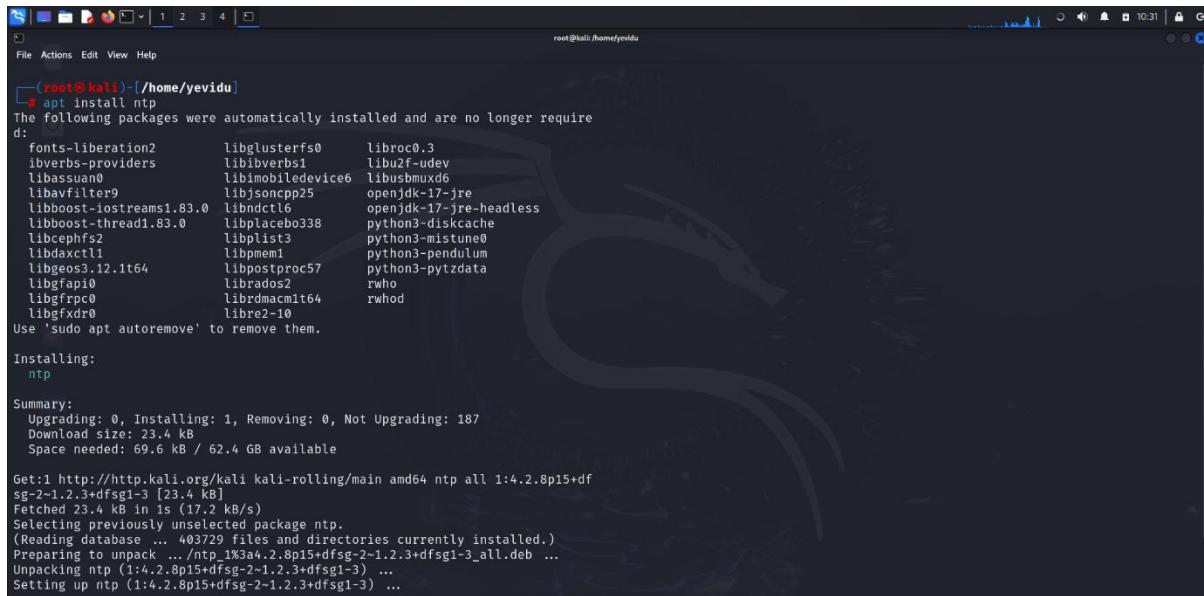
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './NS/IN': 2801:1b8:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './DNSKEY/IN': 2001:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './NS/IN': 2001:500:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './DNSKEY/IN': 2001:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './NS/IN': 2001:500:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './DNSKEY/IN': 2001:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './NS/IN': 2001:7fe:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './DNSKEY/IN': 2001:>
Oct 06 00:02:05 kali named[5336]: network unreachable resolving './NS/IN': 2001:dc3:>
Oct 06 00:02:05 kali named[5336]: managed-keys-zone: Initializing automatic trust an>

( root@kali ) - [ /home/yevidu ]
#
```

2.3 NTP (Network Time Protocol).

First install the NTP package.

Command **apt install ntp**



```
root@kali:[/home/yevidu]
[~]# apt install ntp
The following packages were automatically installed and are no longer required:
d:
 fonts-liberation2      libglusterfs0      libroco.3
 libverbs-providers     libibusverbs1     libub2f-udev
 libassuan0             libimobiledevice6 libusbmuxd6
 libavfilter9            libjsoncpp25      openjdk-17-jre
 libboost-iostreams1.83.0 libndctl6        openjdk-17-jre-headless
 libboost-thread1.83.0   libplacebo338    python3-diskcache
 libcephfs2              libplist3         python3-mistune0
 libdaxctl1              libpmem1          python3-pendulum
 libeos3.12.1t64         libpostproc57   python3-ptzdata
 libgfapi0               librados2        rwho
 libgRPC0                librdmacm1t64   rwhod
 libgfXdr0               libred2-10      Use 'sudo apt autoremove' to remove them.

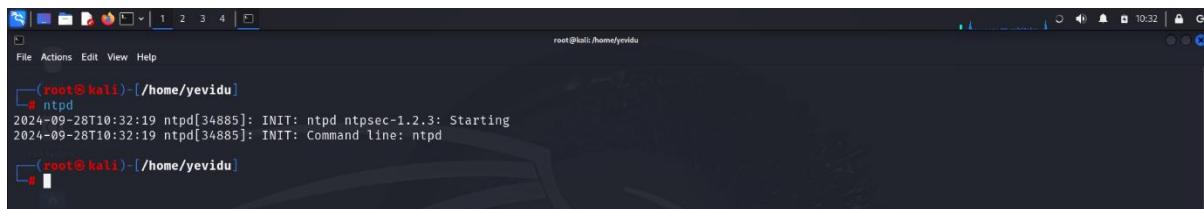
Installing:
 ntp

Summary:
 Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 187
 Download size: 23.4 kB
 Space needed: 69.6 kB / 62.4 GB available

Get:1 http://http.kali.org/kali kali-rolling/main amd64 ntp all 1:4.2.8p15+dfsg-2-1.2.3+dfsg1-3 [23.4 kB]
Fetched 23.4 kB in 1s (17.2 kB/s)
Selecting previously unselected package ntp.
(Reading database ... 403729 files and directories currently installed.)
Preparing to unpack .../ntp_1%3ak4.2.8p15+dfsg-2-1.2.3+dfsg1-3_all.deb ...
Unpacking ntp (1:4.2.8p15+dfsg-2-1.2.3+dfsg1-3) ...
Setting up ntp (1:4.2.8p15+dfsg-2-1.2.3+dfsg1-3) ...
```

Once the installation is complete, the NTP service should start automatically. If not, you can start it manually.

Command **ntpd**

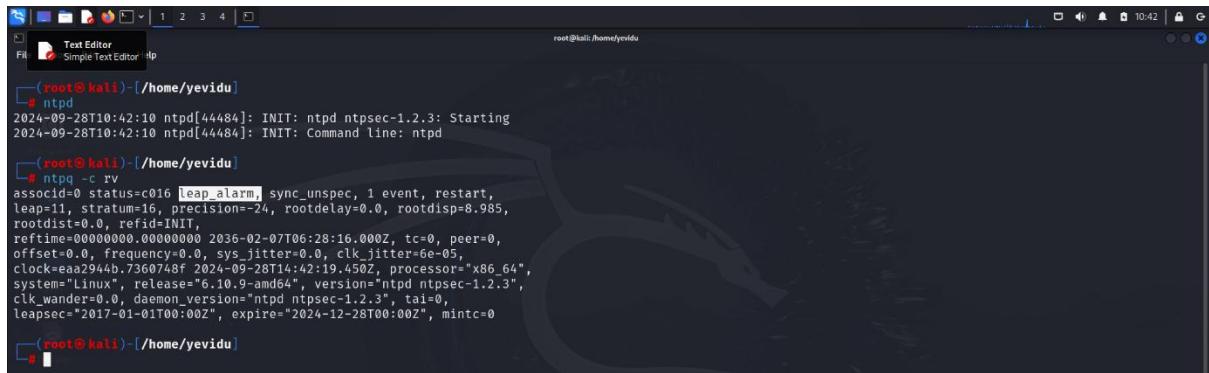


```
root@kali:[/home/yevidu]
[~]# ntpd
2024-09-28T10:32:19 ntpd[34885]: INIT: ntpd ntpsec-1.2.3: Starting
2024-09-28T10:32:19 ntpd[34885]: INIT: Command line: ntpd

[~]#
```

Now you can test our server.

Command **ntpq -c rv**



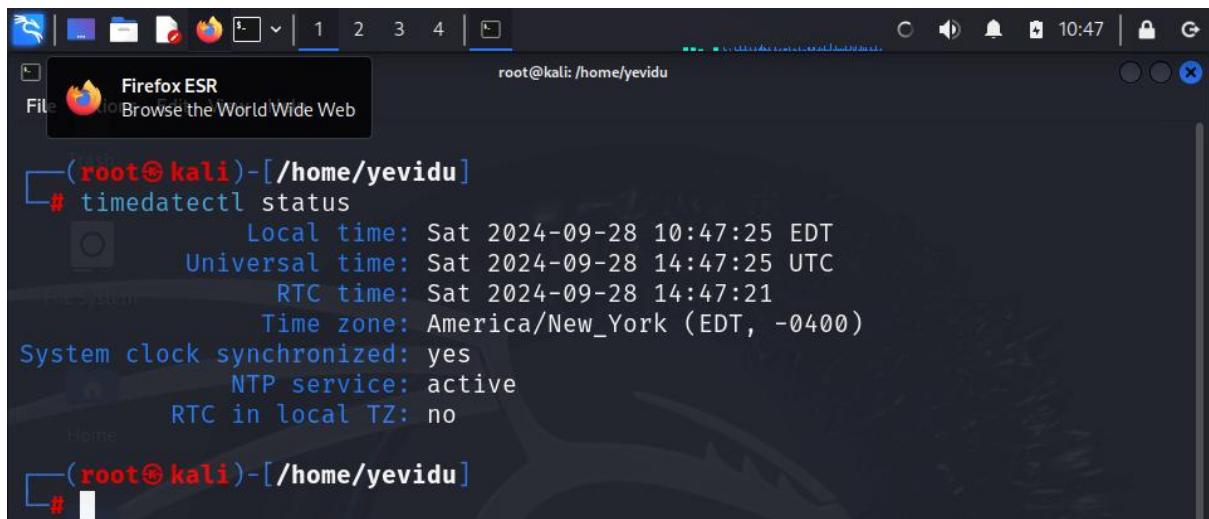
```
(root@kali)-[~/home/yevidu]
# ntpd
2024-09-28T10:42:10 ntpd[44484]: INIT: ntpd ntpsec-1.2.3: Starting
2024-09-28T10:42:10 ntpd[44484]: INIT: Command line: ntpd

[root@kali)-[~/home/yevidu]
# ntpq -c rv
associd=0 status=c016 leap_alarm, sync_unspec, 1 event, restart,
leap=11, stratum=16, precision=-24, rootdelay=0.0, rootdisp=8.985,
rootdist=0.0, refid=INIT,
reftime=00000000.0000000 2036-02-07T06:28:16.000Z, tc=0, peer=0,
offset=0.0, frequency=0.0, sys_jitter=0.0, clk_jitter=6e-05,
clock=eaa2944b.7360748f 2024-09-28T14:42:19.450Z, processor="x86_64",
system="Linux", release="6.10.9-amd64", version="ntpd ntpsec-1.2.3",
clk_wander=0.0, daemon_version="ntpd ntpsec-1.2.3", tai=0,
leapssec="2017-01-01T00:00Z", expire="2024-12-28T00:00Z", mintc=0

[root@kali)-[~/home/yevidu]
#
```

let's check if it is enabled on our server.

Command **timedatectl status**.



```
(root@kali)-[~/home/yevidu]
# timedatectl status
        Local time: Sat 2024-09-28 10:47:25 EDT
        Universal time: Sat 2024-09-28 14:47:25 UTC
            RTC time: Sat 2024-09-28 14:47:21
            Time zone: America/New_York (EDT, -0400)
System clock synchronized: yes
          NTP service: active
    RTC in local TZ: no

[root@kali)-[~/home/yevidu]
#
```

3. Shell Scripting and Security.

3.1 Shell Scripting.

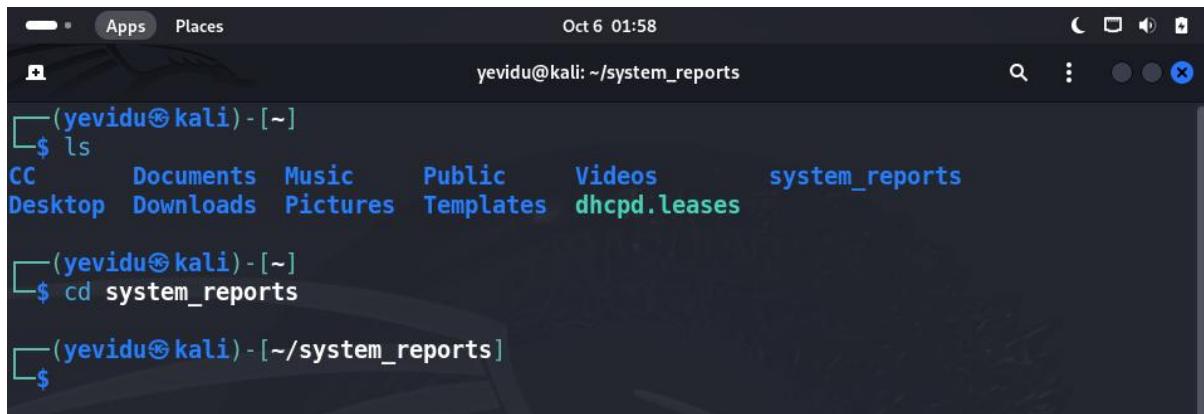
- a). Write a script to automate a report that captures key system details every day.

This script can be scheduled to run using cron jobs. Get System Information such as *Date, Uptime, Free memory and Disk Usage*. Create a report file at the location with the file name as mentioned below.

Destination directory: /home/user/system_reports

Open your terminal and create a new script file named system_report.sh.

Then view created file and navigate to that file.



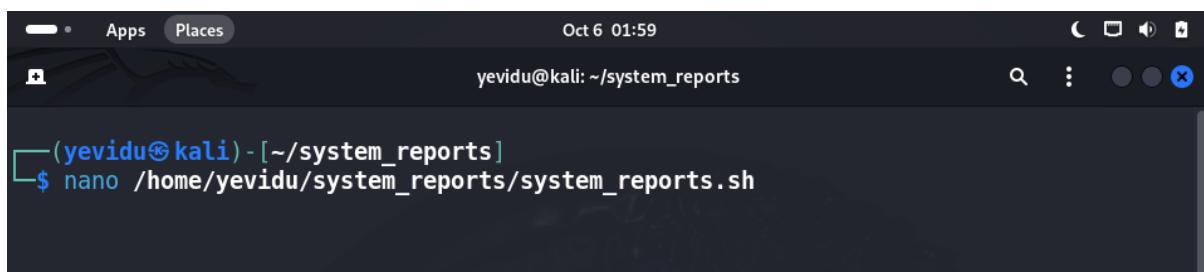
```
Oct 6 01:58
yevidu@kali: ~/system_reports
(yevidu㉿kali)-[~]
$ ls
CC      Documents  Music      Public      Videos      system_reports
Desktop  Downloads  Pictures   Templates  dhcpcd.leases

(yevidu㉿kali)-[~]
$ cd system_reports

(yevidu㉿kali)-[~/system_reports]
$
```

A screenshot of a terminal window titled "yevidu@kali: ~/system_reports". The window shows a command-line interface with the user "yevidu" on a Kali Linux system. The user has navigated to the directory "/home/yevidu/system_reports". The terminal displays a list of files and folders including "CC", "Documents", "Music", "Public", "Videos", "system_reports", "Desktop", "Downloads", "Pictures", "Templates", and "dhcpcd.leases". The prompt is "\$" and the time is "Oct 6 01:58".

Go to new script file for editing.



```
Oct 6 01:59
yevidu@kali: ~/system_reports
(yevidu㉿kali)-[~/system_reports]
$ nano /home/yevidu/system_reports/system_reports.sh
```

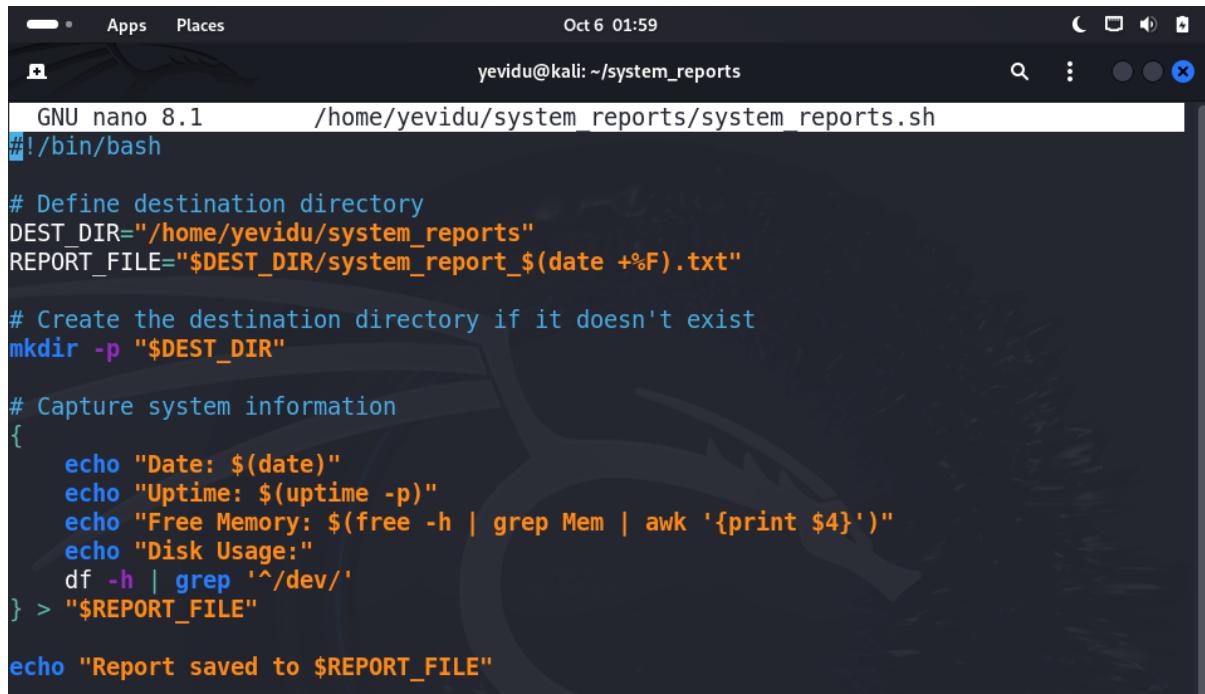
A screenshot of a terminal window titled "yevidu@kali: ~/system_reports". The user has opened a new file named "system_reports.sh" using the "nano" text editor. The prompt is "\$" and the time is "Oct 6 01:59".

Add the following lines to system_report.sh. This script gathers system information and writes it to a report file.

```
#!/bin/bash

DEST_DIR="/home/user/system_reports"
mkdir -p "$DEST_DIR"
DATE=$(date +'%Y-%m-%d')
REPORT_FILE="$DEST_DIR/system_report_$DATE.txt"
{
    echo "System Report for: $DATE"
    echo "=====
    echo "Uptime: $(uptime -p)"
    echo "Free Memory: $(free -h | grep 'Mem' | awk '{print $4}')"
    echo "Disk Usage:"
    df -h | grep '^/dev/'
} > "$REPORT_FILE"

echo "Report generated at: $REPORT_FILE"
```



```
GNU nano 8.1      /home/yevidu/system_reports/system_reports.sh
#!/bin/bash

# Define destination directory
DEST_DIR="/home/yevidu/system_reports"
REPORT_FILE="$DEST_DIR/system_report_$(date +%F).txt"

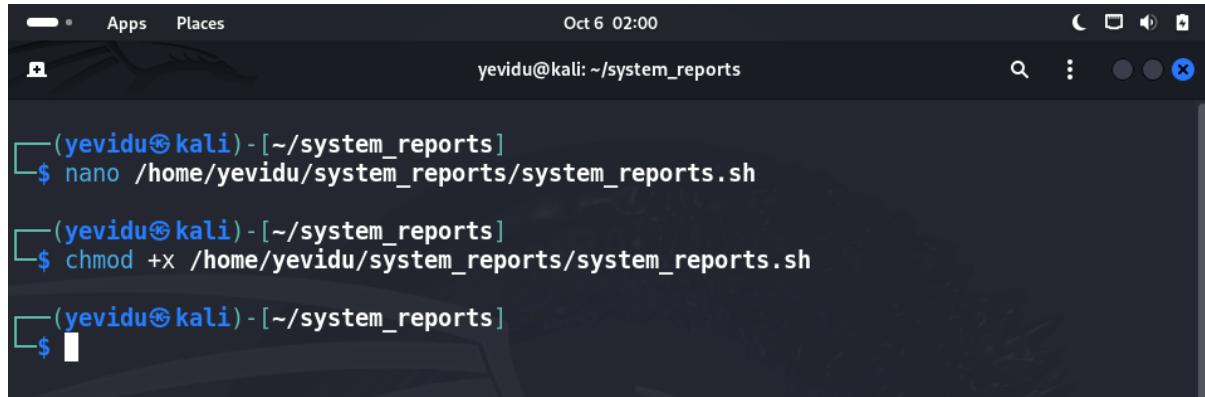
# Create the destination directory if it doesn't exist
mkdir -p "$DEST_DIR"

# Capture system information
{
    echo "Date: $(date)"
    echo "Uptime: $(uptime -p)"
    echo "Free Memory: $(free -h | grep Mem | awk '{print $4}')"
    echo "Disk Usage:"
    df -h | grep '^/dev/'
} > "$REPORT_FILE"

echo "Report saved to $REPORT_FILE"
```

After saving the file.Then make it executable.

Command **chmod +x ~system_report.sh**



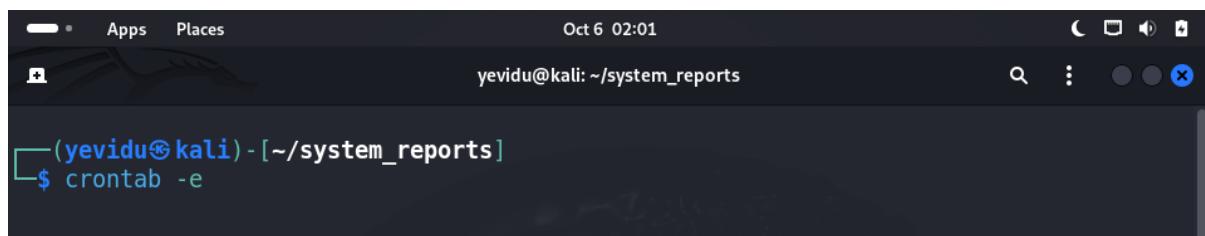
```
(yevidu㉿kali)-[~/system_reports]
$ nano /home/yevidu/system_reports/system_reports.sh

(yevidu㉿kali)-[~/system_reports]
$ chmod +x /home/yevidu/system_reports/system_reports.sh

(yevidu㉿kali)-[~/system_reports]
$
```

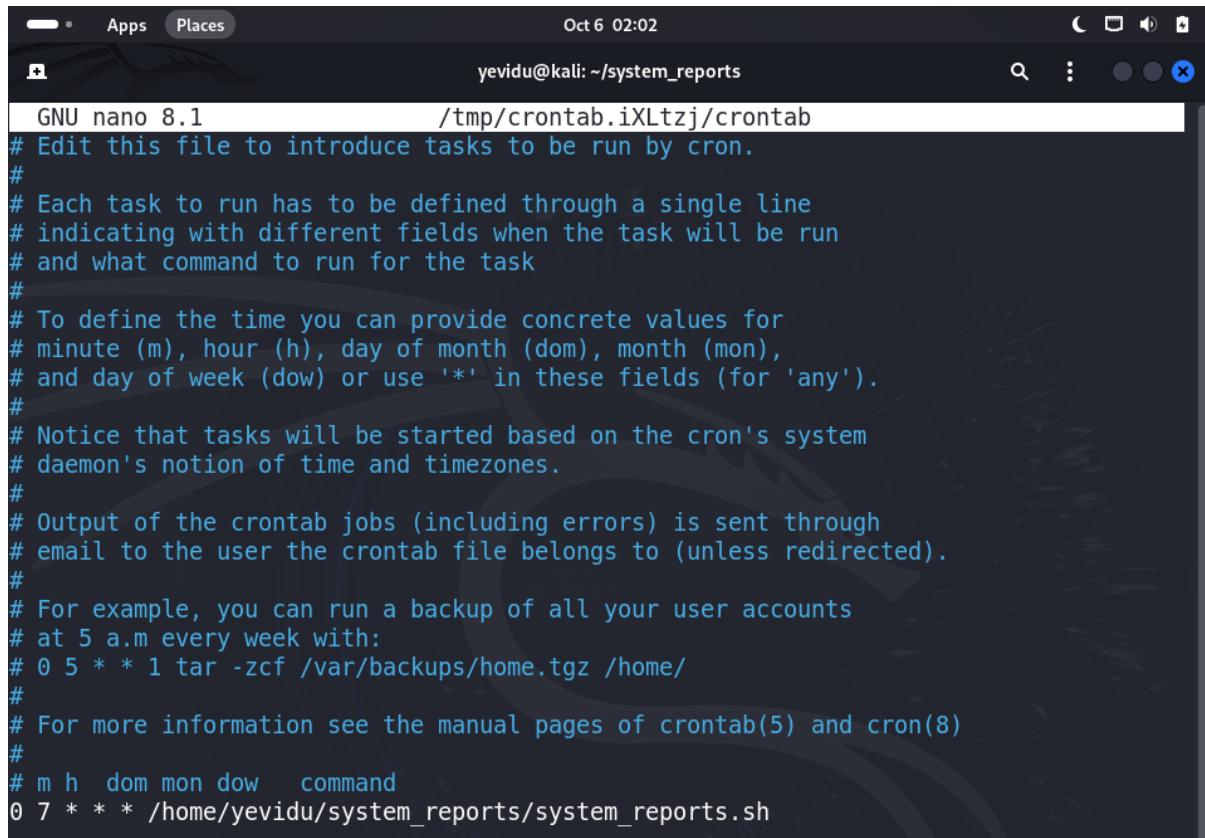
To Set Up a Cron Job lets Open the cron jobs configuration file.

Command **crontab -e**



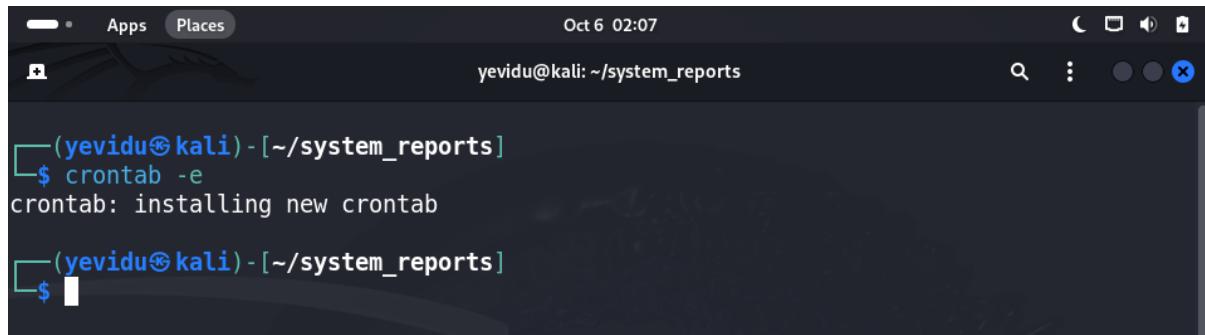
```
(yevidu㉿kali)-[~/system_reports]
$ crontab -e
```

Now let's add a new line to schedule the script to run daily. to run the script every day at 7:00 PM.



The screenshot shows a terminal window titled "yevidu@kali: ~/system_reports". The title bar also displays "Oct 6 02:02". The terminal content is the crontab file, which includes comments about cron scheduling and a single command entry:

```
GNU nano 8.1          /tmp/crontab.iXLtzj/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 7 * * * /home/yevidu/system_reports/system_reports.sh
```



The screenshot shows a terminal window titled "yevidu@kali: ~/system_reports". The title bar also displays "Oct 6 02:07". The terminal shows the user running the "crontab -e" command to edit the crontab file, and then the message "crontab: installing new crontab" indicating the changes have been saved:

```
[yevidu@kali] - [~/system_reports]
$ crontab -e
crontab: installing new crontab

[yevidu@kali] - [~/system_reports]
$
```

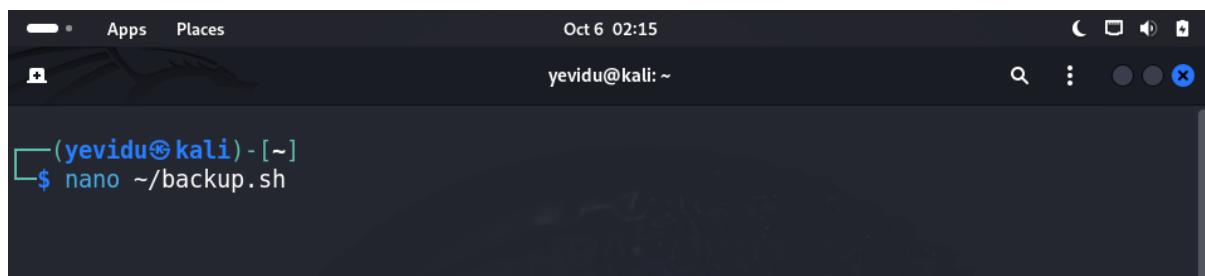
b). Write a script to automate the backup of a critical directory (/home/user/documents) containing important files. This script can be scheduled to run periodically. Make sure to name the backup file with the date.

Source directory : /home/user/documents

Destination directory : /home/user/backup/documents

Open your terminal and create a new file named backup.sh in preferred directory.

Command **nano ~/backup.sh**



Add the following lines to backup.sh.

```
#!/bin/bash
SOURCE_DIR="/home/user/documents"
DEST_DIR="/home/user/backup/documents"
DATE=$(date +'%Y-%m-%d')
BACKUP_FILE="$DEST_DIR/backup_$DATE.tar.gz"
tar -czf "$BACKUP_FILE" -C "$SOURCE_DIR" .
```

```
echo "Backup of $SOURCE_DIR completed successfully at
$BACKUP_FILE"
```

A screenshot of a terminal window titled "GNU nano 8.1". The window shows a shell script named "backup.sh". The script defines source and destination directories, gets today's date in YYYY-MM-DD format, creates a backup file name, creates the backup using tar, and echoes a success message.

```
GNU nano 8.1          /home/yevidu/backup.sh *
#!/bin/bash

# Define source and destination directories
SOURCE_DIR="/home/user/documents"
DEST_DIR="/home/user/backup/documents"

# Get today's date in YYYY-MM-DD format
DATE=$(date +'%Y-%m-%d')

# Create backup file name
BACKUP_FILE="$DEST_DIR/backup_$DATE.tar.gz"

# Create the backup using tar
tar -czf "$BACKUP_FILE" -C "$SOURCE_DIR" .

echo "Backup of $SOURCE_DIR completed successfully at $BACKUP_FILE"
```

After saving the file.Then make it executable.

Command **chmod +x ~system_report.sh**

A screenshot of a terminal window showing the user running the command "chmod +x ~/backup.sh". The terminal shows the user's path as "(yevidu㉿kali)-[~]" and the command being typed as "\$ chmod +x ~/backup.sh".

```
(yevidu㉿kali)-[~]
$ nano ~/backup.sh
(yevidu㉿kali)-[~]
$ chmod +x ~/backup.sh
(yevidu㉿kali)-[~]
$
```

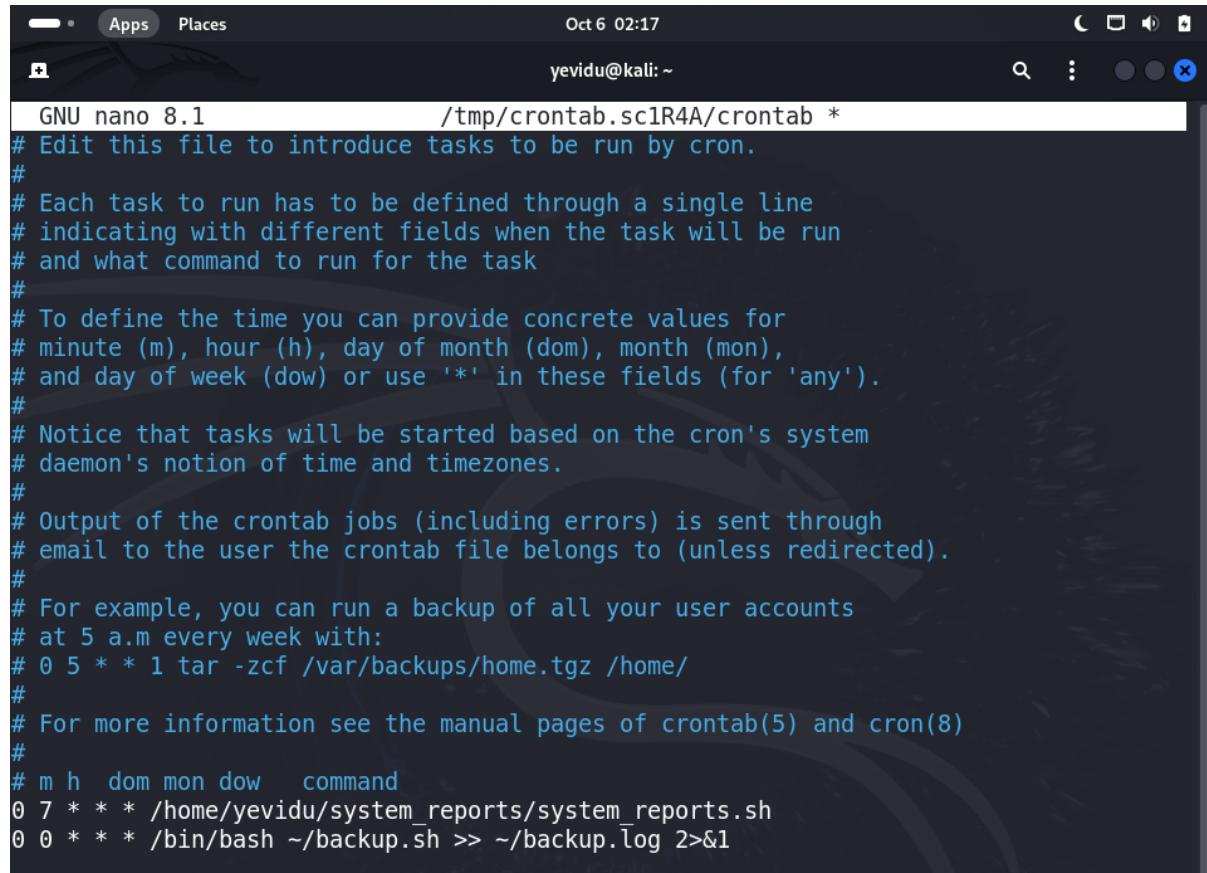
Open crontab.

Command **crontab -e**

A screenshot of a terminal window showing the user running the command "crontab -e". The terminal shows the user's path as "(yevidu㉿kali)-[~]" and the command being typed as "\$ crontab -e".

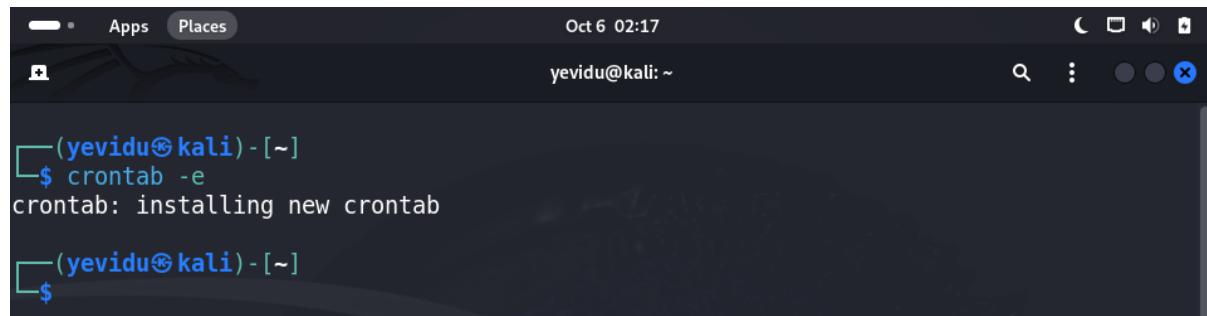
```
(yevidu㉿kali)-[~]
$ crontab -e
```

To schedule the backup to run daily at midnight, added the following line to your crontab:



The screenshot shows a terminal window titled "GNU nano 8.1" with the command "/tmp/crontab.sc1R4A/crontab *". The window displays the contents of a crontab file. The file starts with a comment "# Edit this file to introduce tasks to be run by cron." followed by several explanatory comments about cron syntax and daemon timing. It then shows two specific cron entries: one for running a script at 07:00 every day, and another for running a backup script at 00:00 every day. The terminal window has a dark background with light-colored text.

```
GNU nano 8.1 /tmp/crontab.sc1R4A/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 7 * * * /home/yevidu/system_reports/system_reports.sh
0 0 * * * /bin/bash ~/backup.sh >> ~/backup.log 2>&1
```



The screenshot shows a terminal window with the command "crontab -e" being run. The output shows the message "crontab: installing new crontab", indicating that the changes made in the previous window have been saved. The terminal window has a dark background with light-colored text.

```
(yevidu㉿kali)-[~]
$ crontab -e
crontab: installing new crontab

(yevidu㉿kali)-[~]
$
```

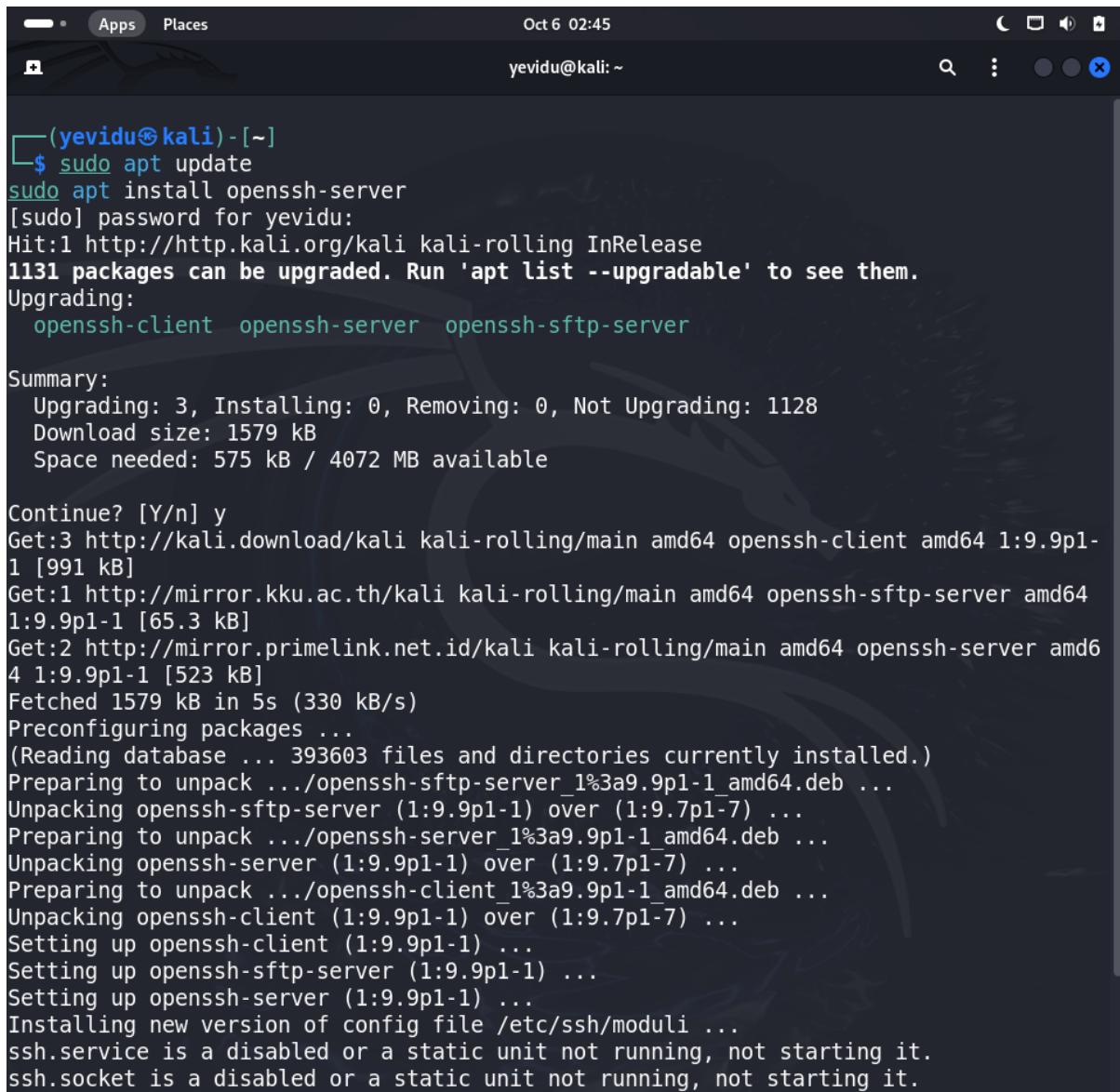
3.2 SSH (Secure Shell).

First install OpenSSH into Linux.

Command

sudo apt update

sudo apt install openssh-server



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title is '(yevidu㉿kali)-[~]'. The command history shows the user running 'sudo apt update' followed by 'sudo apt install openssh-server'. The password prompt '[sudo] password for yevidu:' is visible. The output indicates that 1131 packages can be upgraded. The upgrade summary shows upgrading 3 packages, installing 0, removing 0, and not upgrading 1128 packages. The download size is 1579 kB and space needed is 575 kB / 4072 MB available. The user then types 'y' to continue. The process involves unpacking and preparing multiple packages from various mirrors (kali.download, mirror.kku.ac.th, mirror.primelink.net.id). It also involves setting up the ssh-client and ssh-server packages. A note at the end states that ssh.service and ssh.socket are disabled or static units, so they are not starting.

```
(yevidu㉿kali)-[~]
$ sudo apt update
[sudo] password for yevidu:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1131 packages can be upgraded. Run 'apt list --upgradable' to see them.
Upgrading:
  openssh-client openssh-server openssh-sftp-server

Summary:
  Upgrading: 3, Installing: 0, Removing: 0, Not Upgrading: 1128
  Download size: 1579 kB
  Space needed: 575 kB / 4072 MB available

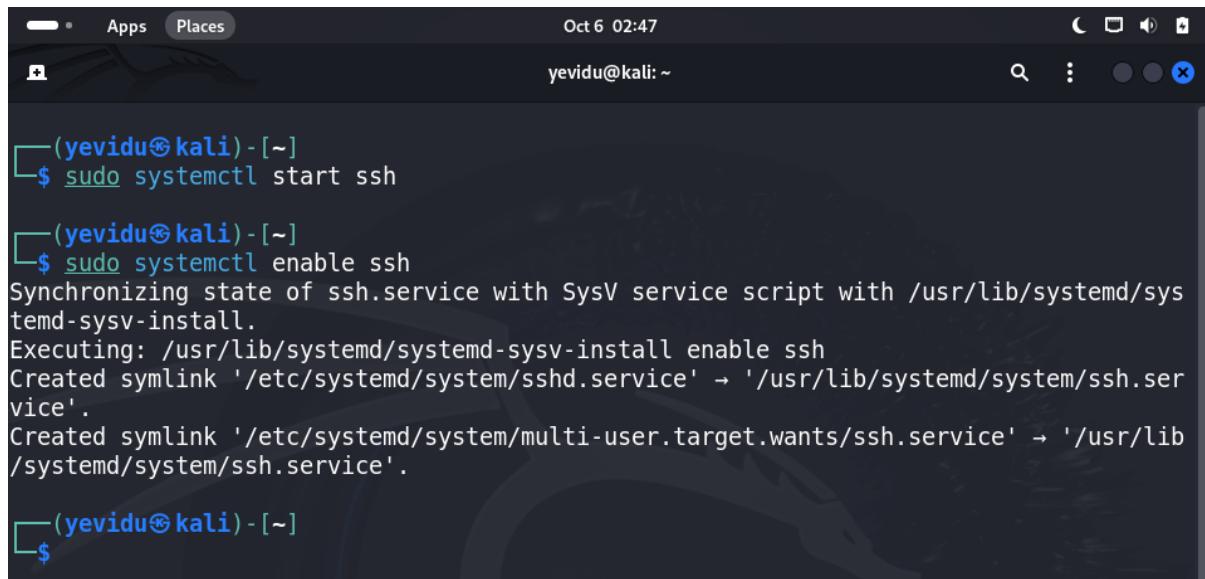
Continue? [Y/n] y
Get:3 http://kali.download/kali kali-rolling/main amd64 openssh-client amd64 1:9.9p1-1 [991 kB]
Get:1 http://mirror.kku.ac.th/kali kali-rolling/main amd64 openssh-sftp-server amd64 1:9.9p1-1 [65.3 kB]
Get:2 http://mirror.primelink.net.id/kali kali-rolling/main amd64 openssh-server amd64 1:9.9p1-1 [523 kB]
Fetched 1579 kB in 5s (330 kB/s)
Preconfiguring packages ...
(Reading database ... 393603 files and directories currently installed.)
Preparing to unpack .../openssh-sftp-server_1%3a9.9p1-1_amd64.deb ...
Unpacking openssh-sftp-server (1:9.9p1-1) over (1:9.7p1-7) ...
Preparing to unpack .../openssh-server_1%3a9.9p1-1_amd64.deb ...
Unpacking openssh-server (1:9.9p1-1) over (1:9.7p1-7) ...
Preparing to unpack .../openssh-client_1%3a9.9p1-1_amd64.deb ...
Unpacking openssh-client (1:9.9p1-1) over (1:9.7p1-7) ...
Setting up openssh-client (1:9.9p1-1) ...
Setting up openssh-sftp-server (1:9.9p1-1) ...
Setting up openssh-server (1:9.9p1-1) ...
Installing new version of config file /etc/ssh/moduli ...
ssh.service is a disabled or a static unit not running, not starting it.
ssh.socket is a disabled or a static unit not running, not starting it.
```

After installation, start the SSH service and enable it.

Command

```
sudo systemctl start sshd
```

```
sudo systemctl enable sshd
```

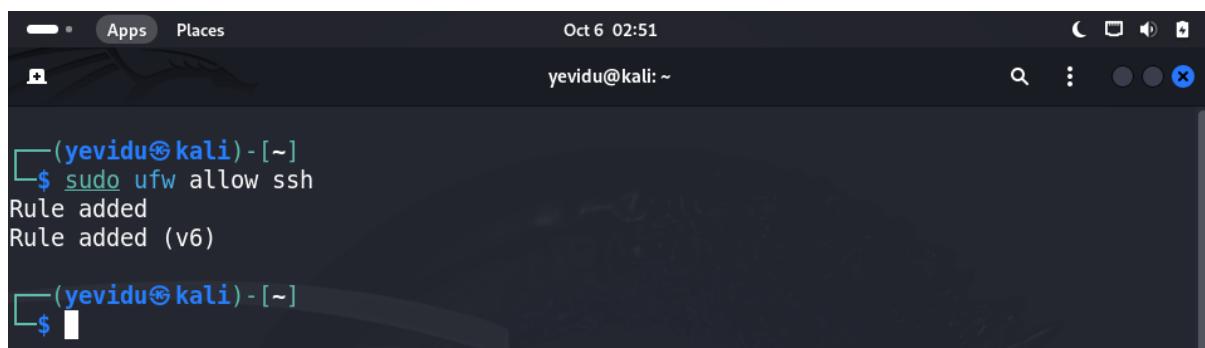


The screenshot shows a terminal window on a Kali Linux desktop. The title bar says "Oct 6 02:47". The user is logged in as "yevidu@kali: ~". The terminal history shows:

```
(yevidu㉿kali)-[~]
$ sudo systemctl start ssh
(yevidu㉿kali)-[~]
$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink '/etc/systemd/system/sshd.service' → '/usr/lib/systemd/system/sshd.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/ssh.service' → '/usr/lib/systemd/system/ssh.service'.
(yevidu㉿kali)-[~]
$
```

Ensure that your firewall allows SSH connections on port 22.

Command **sudo ufw allow ssh**



The screenshot shows a terminal window on a Kali Linux desktop. The title bar says "Oct 6 02:51". The user is logged in as "yevidu@kali: ~". The terminal history shows:

```
(yevidu㉿kali)-[~]
$ sudo ufw allow ssh
Rule added
Rule added (v6)
(yevidu㉿kali)-[~]
$
```

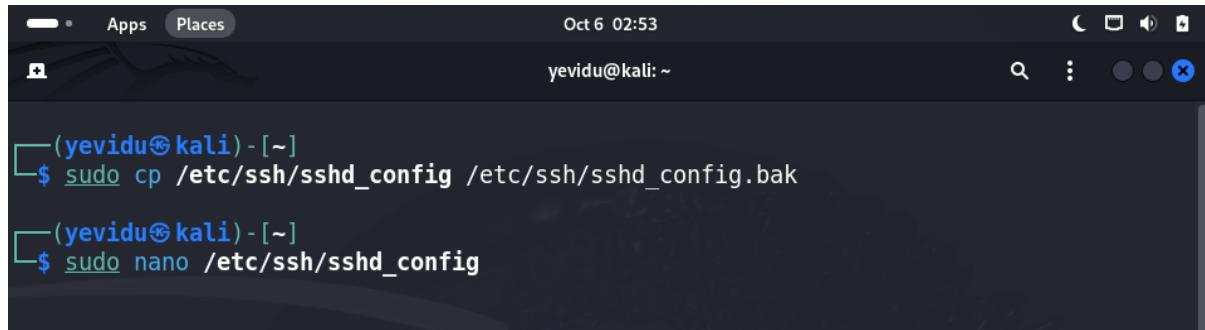
The main configuration file for SSH is located at /etc/ssh/sshd_config. Before making changes, back up the original file.

Then go to configuration file.

Commands

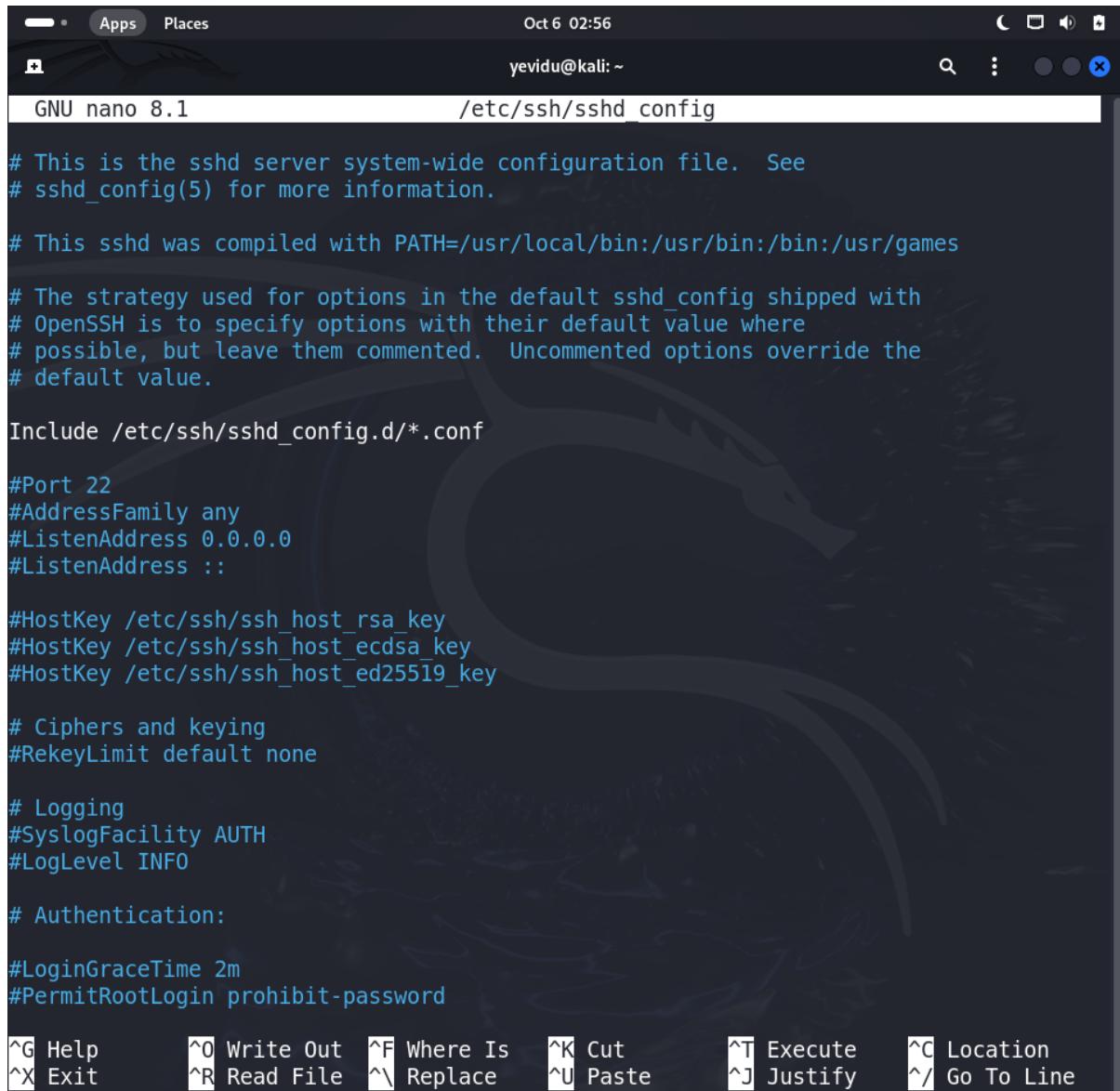
```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

```
sudo nano /etc/ssh/sshd_config
```



```
(yevidu㉿kali)-[~]
$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
(yevidu㉿kali)-[~]
$ sudo nano /etc/ssh/sshd_config
```

Edit the configuration file.



The screenshot shows a terminal window titled "GNU nano 8.1" with the file "/etc/ssh/sshd_config" open. The terminal interface includes a header bar with "Apps", "Places", the date "Oct 6 02:56", and a user "yevidu@kali: ~". Below the header is a toolbar with search, settings, and close buttons. The main area displays the SSHD configuration file with various commented-out and uncommented lines. At the bottom, there is a menu of keyboard shortcuts for nano editor commands.

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

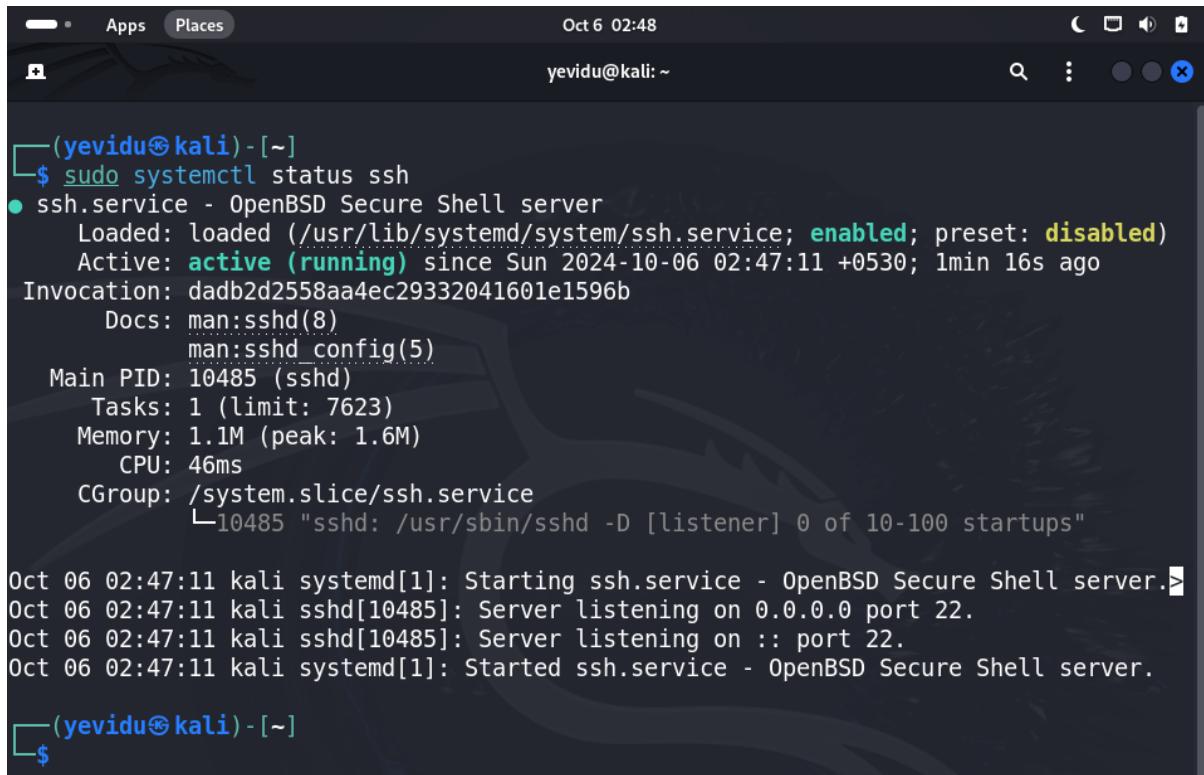
# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password

^G Help      ^O Write Out   ^F Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File   ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Check whether the SSH working properly.

Command **sudo systemctl status ssh**



```
(yevidu㉿kali)-[~]
└$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: disabled)
  Active: active (running) since Sun 2024-10-06 02:47:11 +0530; 1min 16s ago
    Invocation: dadb2d2558aa4ec29332041601e1596b
      Docs: man:sshd(8)
             man:sshd_config(5)
    Main PID: 10485 (sshd)
      Tasks: 1 (limit: 7623)
     Memory: 1.1M (peak: 1.6M)
        CPU: 46ms
      CGroup: /system.slice/ssh.service
              └─10485 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

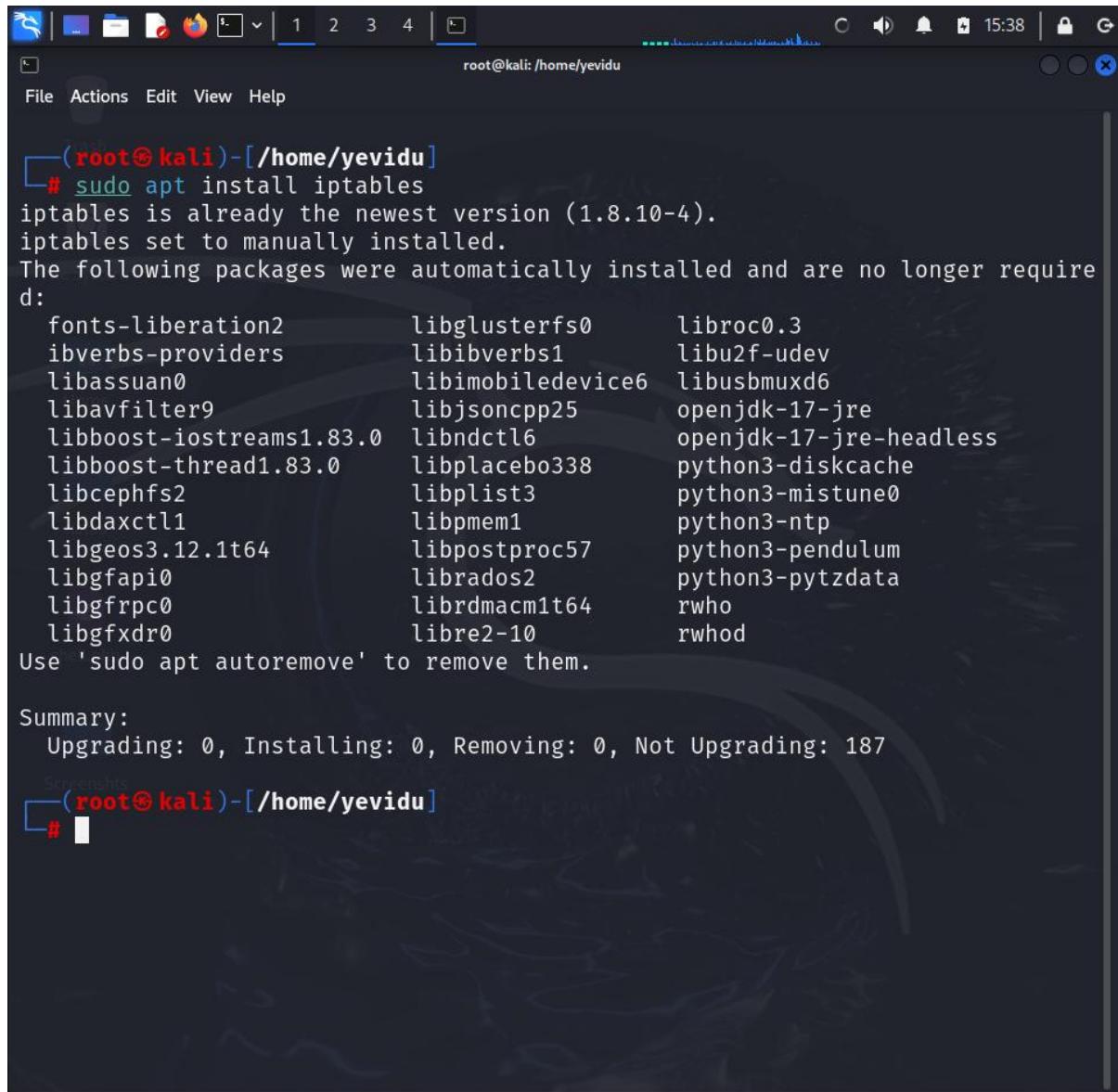
Oct 06 02:47:11 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell server.>
Oct 06 02:47:11 kali sshd[10485]: Server listening on 0.0.0.0 port 22.
Oct 06 02:47:11 kali sshd[10485]: Server listening on :: port 22.
Oct 06 02:47:11 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell server.

(yevidu㉿kali)-[~]
└$
```

3.3 Iptables and ACLs

First, you need to install iptables. Most Linux operating systems come with this package pre-installed.

Command **sudo apt install iptables**



```
root@kali: /home/yevidu
File Actions Edit View Help
└─(root@kali)-[/home/yevidu]
# sudo apt install iptables
iptables is already the newest version (1.8.10-4).
iptables set to manually installed.
The following packages were automatically installed and are no longer required:
  fonts-liberation2      libglusterfs0      libroc0.3
  ibverbs-providers     libibverbs1       libu2f-udev
  libassuan0            libimobiledevice6 libusbmuxd6
  libavfilter9          libjsoncpp25      openjdk-17-jre
  libboost-iostreams1.83.0 libndctl6        openjdk-17-jre-headless
  libboost-thread1.83.0  libplacebo338    python3-diskcache
  libcephdfs            libplist3         python3-mistune0
  libdaxctl1            libpmem1         python3-ntp
  libgeos3.12.1t64      libpostproc57   python3-pendulum
  libgfapi0              librados2        python3-pytzdata
  libgfrpc0              librdmacm1t64   rwho
  libgfxdr0              libre2-10       rwhod
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 187
Screenshots
└─(root@kali)-[/home/yevidu]
#
```

Check the status of your current iptables configuration.

Command **iptables -L -v**

```
(root@kali)-[~/home/yevidu]
# iptables -L -v
Chain INPUT (policy ACCEPT 9 packets, 2844 bytes)
 pkts bytes target     prot opt in     out     source          destination
on
      0     0 ACCEPT     tcp  --  any    any    anywhere       anywhere
      0     0           tcp dpt:http

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source          destination
on

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source          destination

#
```

Web Server Security: Allow incoming traffic only on port 80 (HTTP) and port 443 (HTTPS) for your web server. Block all other incoming traffic by default.

Commands

iptables -A INPUT -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -p tcp --dport 443 -j ACCEPT

iptables -A INPUT -j DROP

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal window title is 'root@kali: /home/yevidu'. The window has tabs labeled 1, 2, 3, and 4. The status bar at the top right shows 'CPU usage: 0.8%' and the time '18:52'. The terminal content displays the following commands and their output:

```
(root@kali)-[~/home/yevidu]
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT

(root@kali)-[~/home/yevidu]
# iptables -A INPUT -p tcp --dport 443 -j ACCEPT

(root@kali)-[~/home/yevidu]
# iptables -A INPUT -j DROP

(root@kali)-[~/home/yevidu]
# iptables -L -v
Chain INPUT (policy ACCEPT 48 packets, 15168 bytes)
 pkts bytes target     prot opt in     out     source          destination
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
                tcp dpt:http
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
                tcp dpt:http
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
                tcp dpt:https
  0     0 DROP        all  --  any    any    anywhere       anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source          destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source          destination

(root@kali)-[~/home/yevidu]
#
```

Remote Administration Access: Allow SSH access (port 22) only from specific IP addresses of your trusted machines used for administration. This restricts remote access attempts to authorized sources.

Commands **iptables -A INPUT -p tcp --dport 22 -j ACCEPT -s 192.168.1.151**

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal is running as root, indicated by the red text '(root@kali)'. The user has run several commands related to iptables:

- # iptables -A INPUT -p tcp --dport 22 -j ACCEPT -s 192.168.1.151
- # iptables -L -v

The output of the -L -v command shows the current iptables rules:

Chain	Policy	Action	Protocols	In	Out	Source	Destination
INPUT	ACCEPT	ACCEPT	tcp	--	any	anywhere	anywhere
			tcp	dpt:http			
			ACCEPT	tcp	--	any	anywhere
			tcp	dpt:http			
			ACCEPT	tcp	--	any	anywhere
			tcp	dpt:https			
			DROP	all	--	any	anywhere
			ACCEPT	tcp	--	any	anywhere
			tcp	dpt:ssh			
FORWARD	ACCEPT	ACCEPT	0	packets, 0 bytes			
OUTPUT	ACCEPT	ACCEPT	0	packets, 0 bytes			

At the bottom of the terminal, there is a small white box containing a black icon.

Allow Specific Applications: If you know the port numbers used by specific applications you want to allow (like a video conferencing app using port 443), you can create an ACL rule to permit traffic only for those ports.

Commands **iptables -A INPUT -p tcp --dport 53 -j ACCEPT**

The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title is 'root@kali: /home/yevidu'. The user has run the command 'iptables -A INPUT -p tcp --dport 53 -j ACCEPT' to allow DNS traffic. Then, they ran 'iptables -L -v' to list the current rules, which shows the new rule added to the INPUT chain. The terminal also displays the FORWARD and OUTPUT chains, both with their default policy set to ACCEPT.

```
(root㉿kali)-[~/home/yevidu]
# iptables -A INPUT -p tcp --dport 53 -j ACCEPT

(root㉿kali)-[~/home/yevidu]
# iptables -L -v
Chain INPUT (policy ACCEPT 48 packets, 15168 bytes)
 pkts bytes target     prot opt in     out     source          destination
on
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
      0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
      0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
      0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
      2   144 DROP       all  --  any    any    anywhere       anywhere
      0     0 ACCEPT      tcp  --  any    any    192.168.1.151  anywhere
      0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
      0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source          destination
on

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source          destination

(root㉿kali)-[~/home/yevidu]
#
```

Allow Pings (ICMP Echo Request): This basic rule allows ping requests (ICMP Echo Request) to your machine, which can be helpful for troubleshooting network connectivity.

Commands

iptables -A OUTPUT -p icmp -j ACCEPT

The screenshot shows a terminal window with a dark theme. At the top, there are icons for file operations like copy, paste, and save, followed by tabs labeled 1, 2, 3, 4. The title bar says "root@kali: /home/yevidu". The status bar at the bottom right shows the time as 18:58. The terminal content is as follows:

```
# iptables -A OUTPUT -p icmp -j ACCEPT
[root@kali ~]#
[root@kali ~]# iptables -L -v
Chain INPUT (policy ACCEPT 48 packets, 15168 bytes)
pkts bytes target     prot opt in     out    source          destination
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
        0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
        0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
        0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
        0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
  14   3448 DROP       all  --  any    any    anywhere       anywhere
  0     0 ACCEPT      tcp  --  any    any    192.168.1.151  anywhere
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
  0     0 ACCEPT      tcp  --  any    any    anywhere       anywhere
  0     0 ACCEPT      icmp --  any   any    anywhere       anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out    source          destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out    source          destination
  0     0 ACCEPT      icmp --  any   any    anywhere       anywhere

[root@kali ~]#

```

Printer Server Access: For a printer server, allow printing traffic (port 9100) only from specific IP addresses within your local network. Block all external access to the printer server to prevent unauthorized printing.

Commands

iptables -A INPUT -p tcp --dport 9100 -s 192.168.1.151 -j ACCEPT

iptables -A INPUT -p tcp --dport 9100! -s 192.168.1.0/24 -j DROP

```
Oct 6 04:27
root@kali:/home/yevidu

[root@kali]# iptables -A INPUT -p tcp --dport 9100 -s 192.168.1.151 -j ACCEPT
[root@kali]# iptables -A INPUT -p tcp --dport 9100 ! -s 192.168.1.0/24 -j DROP
[root@kali]# iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target     prot opt in     out      source          destination
12356 1198K ufw-before-logging-input  all    --    any    any    anywhere        anywhere
12356 1198K ufw-before-input   all    --    any    any    anywhere        anywhere
187 16664 ufw-after-input   all    --    any    any    anywhere        anywhere
8 608 ufw-after-logging-input all    --    any    any    anywhere        anywhere
8 608 ufw-reject-input   all    --    any    any    anywhere        anywhere
8 608 ufw-track-input   all    --    any    any    anywhere        anywhere
0 0 ACCEPT    all    --    any    any    anywhere        anywhere
ctstate RELATED,ESTABLISHED
0 0 ACCEPT    tcp    --    any    any    anywhere        anywhere
tcp dpt:http
0 0 ACCEPT    tcp    --    any    any    anywhere        anywhere
tcp dpt:https
0 0 ACCEPT    tcp    --    any    any    192.168.1.10    anywhere
tcp dpt:ssh
0 0 ACCEPT    tcp    --    any    any    192.168.1.0/24  anywhere
tcp dpt:ssh
0 0 ACCEPT    tcp    --    any    any    192.168.1.10    anywhere
tcp dpt:ssh
0 0 ACCEPT    tcp    --    any    any    anywhere        anywhere
```

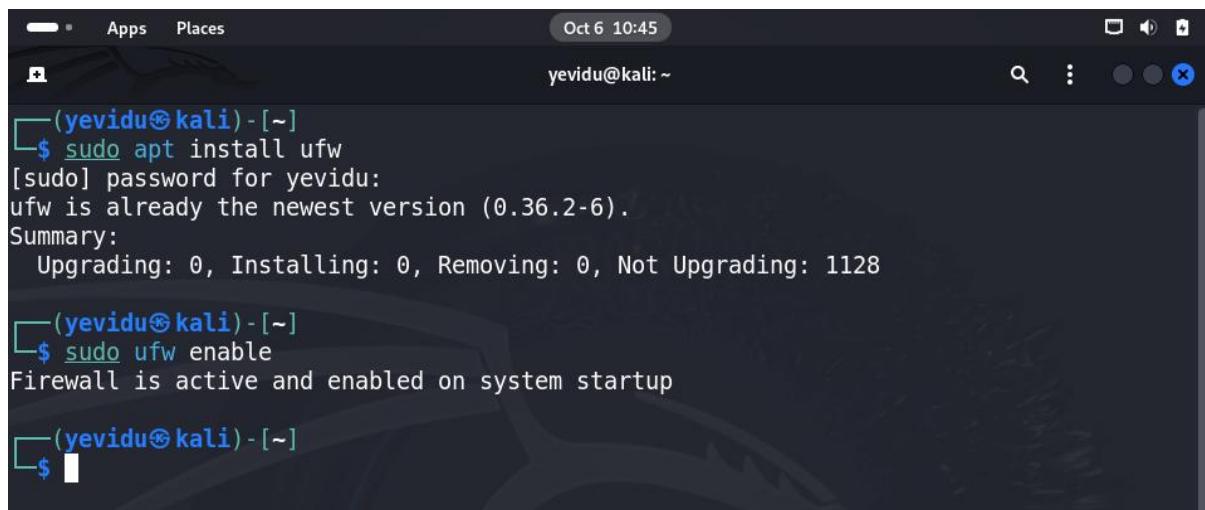
4.Best Practices.

4.1). Implementing a Firewall in Kali Linux.

Installing a firewall in Kali Linux is essential for protecting your system from threats and illegal access. This manual offers a methodical way to putting the Uncomplicated Firewall (UFW), which streamlines firewall rule management, into practice.

First need to install UFW if it's not already present on Linux.

After installed, enable UFW to start protecting Linux.



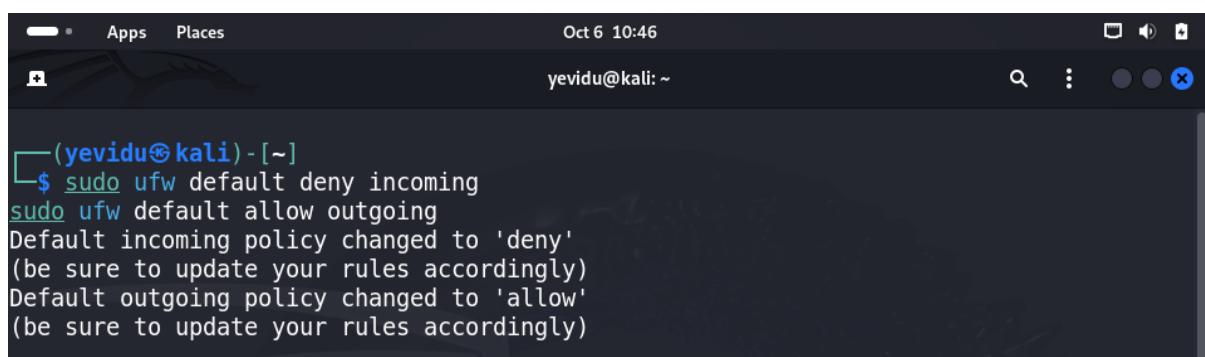
The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:

```
(yevidu㉿kali)-[~]
$ sudo apt install ufw
[sudo] password for yevidu:
ufw is already the newest version (0.36.2-6).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1128

(yevidu㉿kali)-[~]
$ sudo ufw enable
Firewall is active and enabled on system startup

(yevidu㉿kali)-[~]
$
```

To improve security, set default rules for all incoming and outgoing traffic.

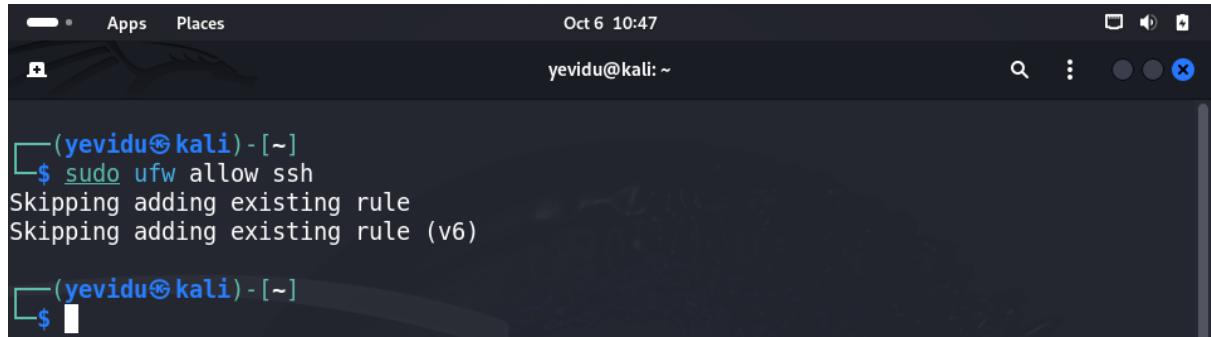


The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output is as follows:

```
(yevidu㉿kali)-[~]
$ sudo ufw default deny incoming
sudo ufw default allow outgoing
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

- **Deny Incoming:** This prevents all incoming connections unless explicitly allowed, reducing the risk of unauthorized access.
- **Allow Outgoing:** This permits all outgoing connections, which is generally safe for applications that need internet access.

Then allow specific services through the firewall.

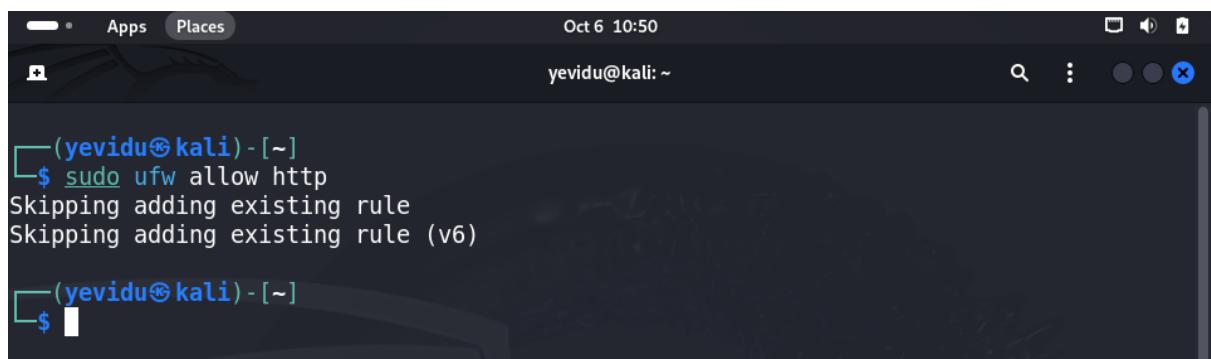


```
Oct 6 10:47
yevidu@kali: ~

(yevidu㉿kali)-[~]
$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)

(yevidu㉿kali)-[~]
$
```

Allow HTTP traffic.



```
Oct 6 10:50
yevidu@kali: ~

(yevidu㉿kali)-[~]
$ sudo ufw allow http
Skipping adding existing rule
Skipping adding existing rule (v6)

(yevidu㉿kali)-[~]
$
```

Check and verify the status of UFW and see whether which rules are active.

```
(yevidu㉿kali)-[~]
└$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To          Action    From
--          ----
22/tcp      ALLOW IN  Anywhere
22          ALLOW IN  Anywhere
80/tcp      ALLOW IN  Anywhere
22/tcp (v6) ALLOW IN  Anywhere (v6)
22 (v6)    ALLOW IN  Anywhere (v6)
80/tcp (v6) ALLOW IN  Anywhere (v6)

(yevidu㉿kali)-[~]
└$
```

Enable logging.

It monitors traffic and detect unauthorized access attempts.

```
(yevidu㉿kali)-[~]
└$ sudo ufw logging on
Logging enabled

(yevidu㉿kali)-[~]
└$
```

Your firewall can be further customized by permitting or prohibiting IP addresses or ranges. To permit access, for example, from a particular IP address

```
(yevidu㉿kali)-[~]
└$ sudo ufw allow from 192.168.1.0/24 to any port 80
Rule added

(yevidu㉿kali)-[~]
└$
```

Deny access from a specific IP address.

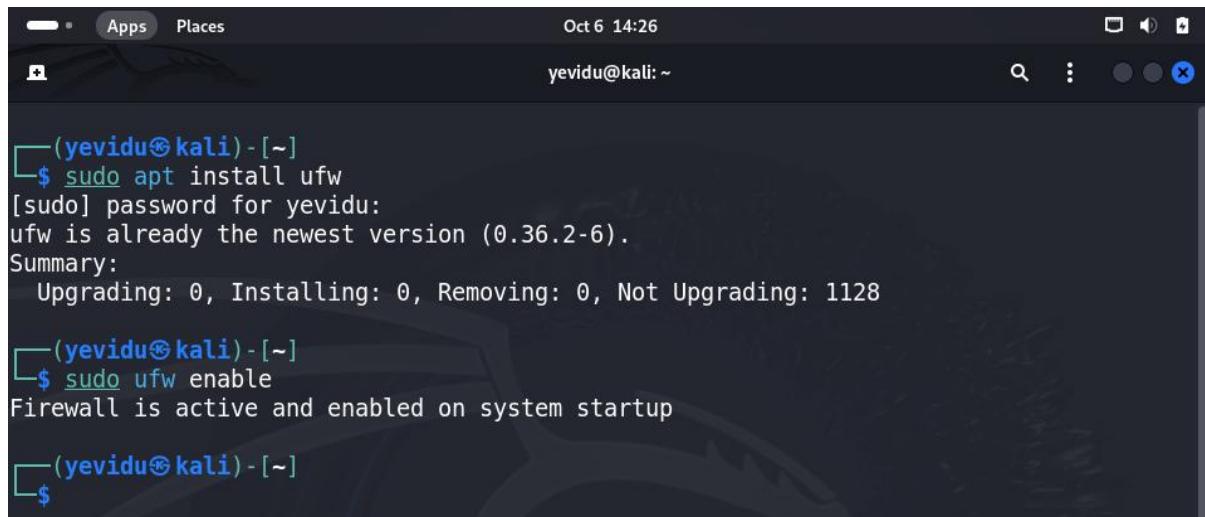
```
Oct 6 10:51  
yevidu@kali: ~  
└─(yevidu㉿kali)-[~]  
└$ sudo ufw deny from 192.168.1.10  
Rule added  
└─(yevidu㉿kali)-[~]  
└$
```

Now check active services using.

```
Oct 6 10:52  
yevidu@kali: ~  
└─(yevidu㉿kali)-[~]  
└$ sudo systemctl list-units --type=service --state=running  
UNIT LOAD ACTIVE SUB DESCRIPTION  
accounts-daemon.service loaded active running Accounts Service  
colord.service loaded active running Manage, Install and Generate >  
cron.service loaded active running Regular background program pr>  
dbus.service loaded active running D-Bus System Message Bus  
fwupd.service loaded active running Firmware update daemon  
gdm.service loaded active running GNOME Display Manager  
haveged.service loaded active running Entropy Daemon based on the H>  
ModemManager.service loaded active running Modem Manager  
named.service loaded active running BIND Domain Name Server  
NetworkManager.service loaded active running Network Manager  
polkit.service loaded active running Authorization Manager  
power-profiles-daemon.service loaded active running Power Profiles daemon  
rtkit-daemon.service loaded active running RealtimeKit Scheduling Policy>  
ssh.service loaded active running OpenBSD Secure Shell server  
systemd-journald.service loaded active running Journal Service  
systemd-logind.service loaded active running User Login Management  
systemd-udevd.service loaded active running Rule-based Manager for Device>  
udisks2.service loaded active running Disk Manager  
upower.service loaded active running Daemon for power management  
user@0.service loaded active running User Manager for UID 0  
user@1000.service loaded active running User Manager for UID 1000  
virtualbox-guest-utils.service loaded active running Virtualbox guest utils  
wpa_supplicant.service loaded active running WPA supplicant  
  
Legend: LOAD → Reflects whether the unit definition was properly loaded.  
ACTIVE → The high-level unit activation state, i.e. generalization of SUB.  
SUB → The low-level unit activation state, values depend on unit type.  
23 loaded units listed.  
└─(yevidu㉿kali)-[~]  
└$
```

4.2). Minimize Open Ports and Services.

First install and Enable UFW.



```
(yevidu㉿kali)-[~]
$ sudo apt install ufw
[sudo] password for yevidu:
ufw is already the newest version (0.36.2-6).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1128

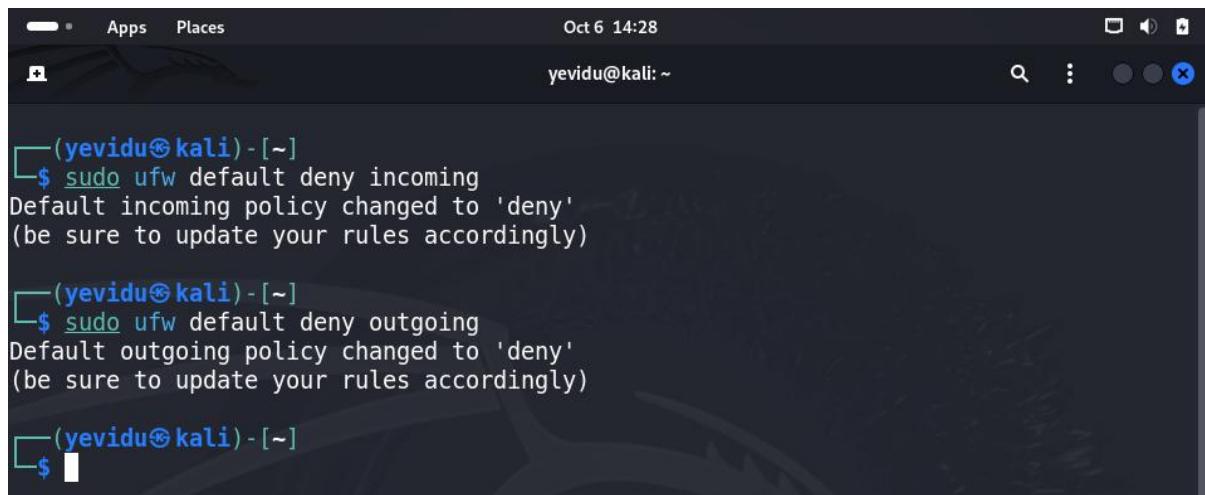
(yevidu㉿kali)-[~]
$ sudo ufw enable
Firewall is active and enabled on system startup

(yevidu㉿kali)-[~]
$
```

Then Set Default Policies.

Deny Incoming Connections: This prevents all incoming connections by default.

Allow Outgoing Connections: This allows all outgoing connections.



```
(yevidu㉿kali)-[~]
$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

(yevidu㉿kali)-[~]
$ sudo ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)

(yevidu㉿kali)-[~]
$
```

Allow SSH Access.

Then Open Other Specific Ports,

HTTP (port 80) ; **sudo ufw allow 80/tcp**

HTTPS (port 443) ; **sudo ufw allow 443/tcp**

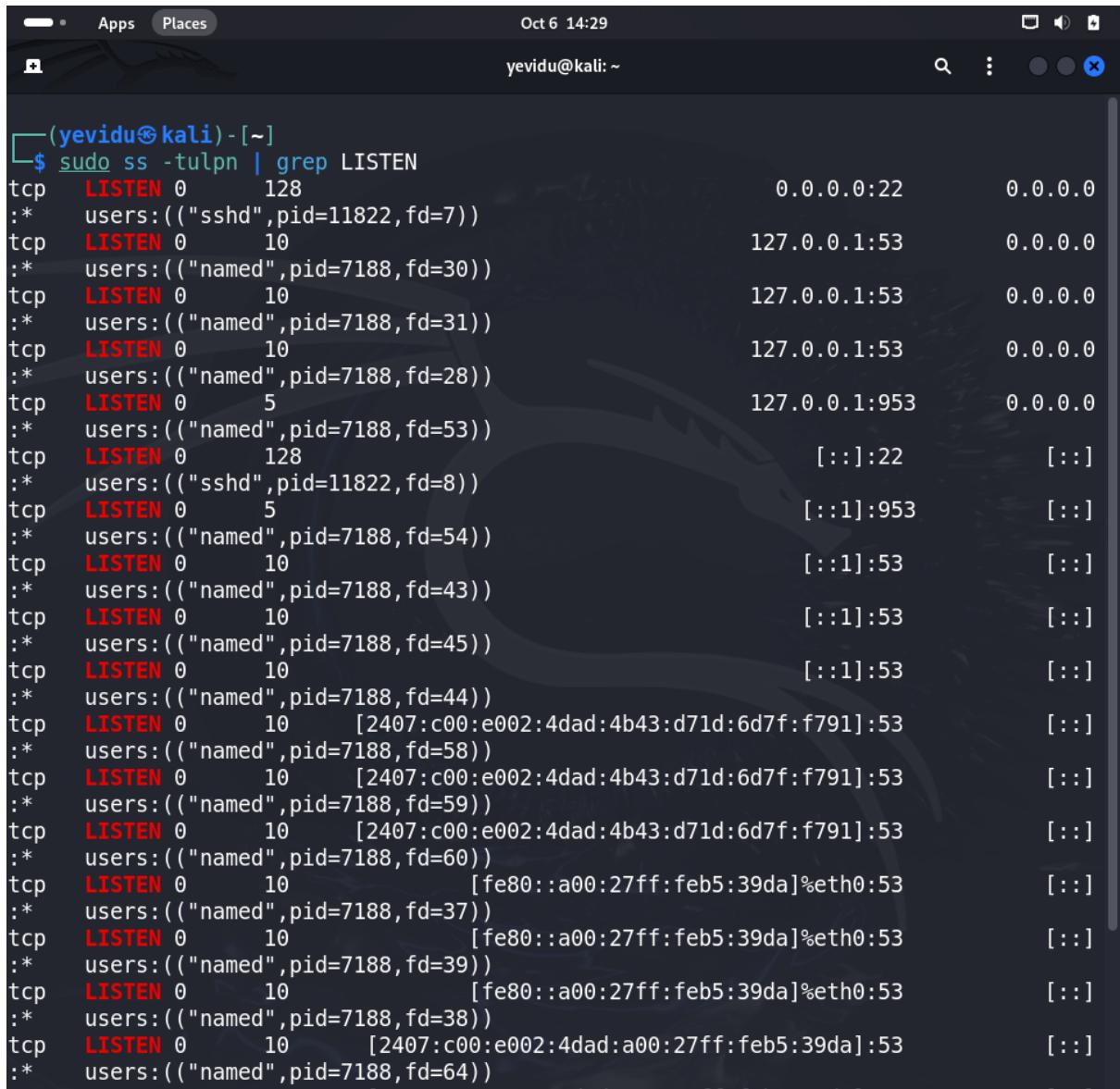


The screenshot shows a terminal window on a Kali Linux desktop. The terminal title is '(yevidu㉿kali)-[~]'. The user has run three commands to open ports:

- \$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
- \$ sudo ufw allow 80/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
- \$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)

The terminal prompt '\$' is visible at the bottom.

Check Open Ports.



(yevidu㉿kali)-[~]\$ sudo ss -tulpn | grep LISTEN

Protocol	Port	Local Address	Foreign Address	User
tcp	LISTEN 0	128	0.0.0.0:22	0.0.0.0
:	*	users:(("sshd",pid=11822,fd=7))		
tcp	LISTEN 0	10	127.0.0.1:53	0.0.0.0
:	*	users:(("named",pid=7188,fd=30))		
tcp	LISTEN 0	10	127.0.0.1:53	0.0.0.0
:	*	users:(("named",pid=7188,fd=31))		
tcp	LISTEN 0	10	127.0.0.1:53	0.0.0.0
:	*	users:(("named",pid=7188,fd=28))		
tcp	LISTEN 0	5	127.0.0.1:953	0.0.0.0
:	*	users:(("named",pid=7188,fd=53))		
tcp	LISTEN 0	128	[::]:22	[::]
:	*	users:(("sshd",pid=11822,fd=8))		
tcp	LISTEN 0	5	[::1]:953	[::]
:	*	users:(("named",pid=7188,fd=54))		
tcp	LISTEN 0	10	[::1]:53	[::]
:	*	users:(("named",pid=7188,fd=43))		
tcp	LISTEN 0	10	[::1]:53	[::]
:	*	users:(("named",pid=7188,fd=45))		
tcp	LISTEN 0	10	[::1]:53	[::]
:	*	users:(("named",pid=7188,fd=44))		
tcp	LISTEN 0	10	[2407:c00:e002:4dad:4b43:d71d:6d7f:f791]:53	[::]
:	*	users:(("named",pid=7188,fd=58))		
tcp	LISTEN 0	10	[2407:c00:e002:4dad:4b43:d71d:6d7f:f791]:53	[::]
:	*	users:(("named",pid=7188,fd=59))		
tcp	LISTEN 0	10	[2407:c00:e002:4dad:4b43:d71d:6d7f:f791]:53	[::]
:	*	users:(("named",pid=7188,fd=60))		
tcp	LISTEN 0	10	[fe80::a00:27ff:feb5:39da]@eth0:53	[::]
:	*	users:(("named",pid=7188,fd=37))		
tcp	LISTEN 0	10	[fe80::a00:27ff:feb5:39da]@eth0:53	[::]
:	*	users:(("named",pid=7188,fd=39))		
tcp	LISTEN 0	10	[fe80::a00:27ff:feb5:39da]@eth0:53	[::]
:	*	users:(("named",pid=7188,fd=38))		
tcp	LISTEN 0	10	[2407:c00:e002:4dad:a00:27ff:feb5:39da]:53	[::]
:	*	users:(("named",pid=7188,fd=64))		

Identify Running Services and list all.

```
(yevidu㉿kali)-[~]
$ systemctl list-units --type=service --state=running
UNIT                                     LOAD   ACTIVE SUB   DESCRIPTION
accounts-daemon.service                 loaded  active running Accounts Service
color.service                           loaded  active running Manage, Install and Generate >
cron.service                            loaded  active running Regular background program pr>
dbus.service                            loaded  active running D-Bus System Message Bus
fwupd.service                           loaded  active running Firmware update daemon
gdm.service                             loaded  active running GNOME Display Manager
haveged.service                         loaded  active running Entropy Daemon based on the H>
ModemManager.service                   loaded  active running Modem Manager
named.service                           loaded  active running BIND Domain Name Server
NetworkManager.service                 loaded  active running Network Manager
polkit.service                          loaded  active running Authorization Manager
power-profiles-daemon.service          loaded  active running Power Profiles daemon
rtkit-daemon.service                  loaded  active running RealtimeKit Scheduling Policy>
ssh.service                            loaded  active running OpenBSD Secure Shell server
systemd-journald.service              loaded  active running Journal Service
systemd-logind.service                loaded  active running User Login Management
systemd-udevd.service                 loaded  active running Rule-based Manager for Device>
udisks2.service                         loaded  active running Disk Manager
upower.service                          loaded  active running Daemon for power management
user@0.service                          loaded  active running User Manager for UID 0
user@1000.service                      loaded  active running User Manager for UID 1000
virtualbox-guest-utils.service         loaded  active running Virtualbox guest utils
wpa_supplicant.service                 loaded  active running WPA supplicant

Legend: LOAD → Reflects whether the unit definition was properly loaded.
        ACTIVE → The high-level unit activation state, i.e. generalization of SUB.
        SUB     → The low-level unit activation state, values depend on unit type.

23 loaded units listed.

(yevidu㉿kali)-[~]
```

Disable Unused Services.

```
(yevidu㉿kali)-[~]
$ sudo systemctl disable cron.service
Synchronizing state of cron.service with SysV service script with /usr/lib/systemd/sysvinitctl.
Executing: /usr/lib/systemd/system-sysv-install disable cron
Removed '/etc/systemd/system/multi-user.target.wants/cron.service'.

(yevidu㉿kali)-[~]
```

Regularly review the firewall status and rules.

```
(yevidu㉿kali)-[~]
$ sudo ufw status numbered
Status: active

To                         Action      From
--                         --          --
[ 1] 22/tcp                ALLOW IN   Anywhere
[ 2] 22                     ALLOW IN   Anywhere
[ 3] 80/tcp                ALLOW IN   Anywhere
[ 4] 80                     ALLOW IN   192.168.1.0/24
[ 5] Anywhere              DENY IN    192.168.1.10
[ 6] 443/tcp               ALLOW IN   Anywhere
[ 7] 22/tcp (v6)           ALLOW IN   Anywhere (v6)
[ 8] 22 (v6)               ALLOW IN   Anywhere (v6)
[ 9] 80/tcp (v6)           ALLOW IN   Anywhere (v6)
[10] 443/tcp (v6)          ALLOW IN   Anywhere (v6)

(yevidu㉿kali)-[~]
$
```

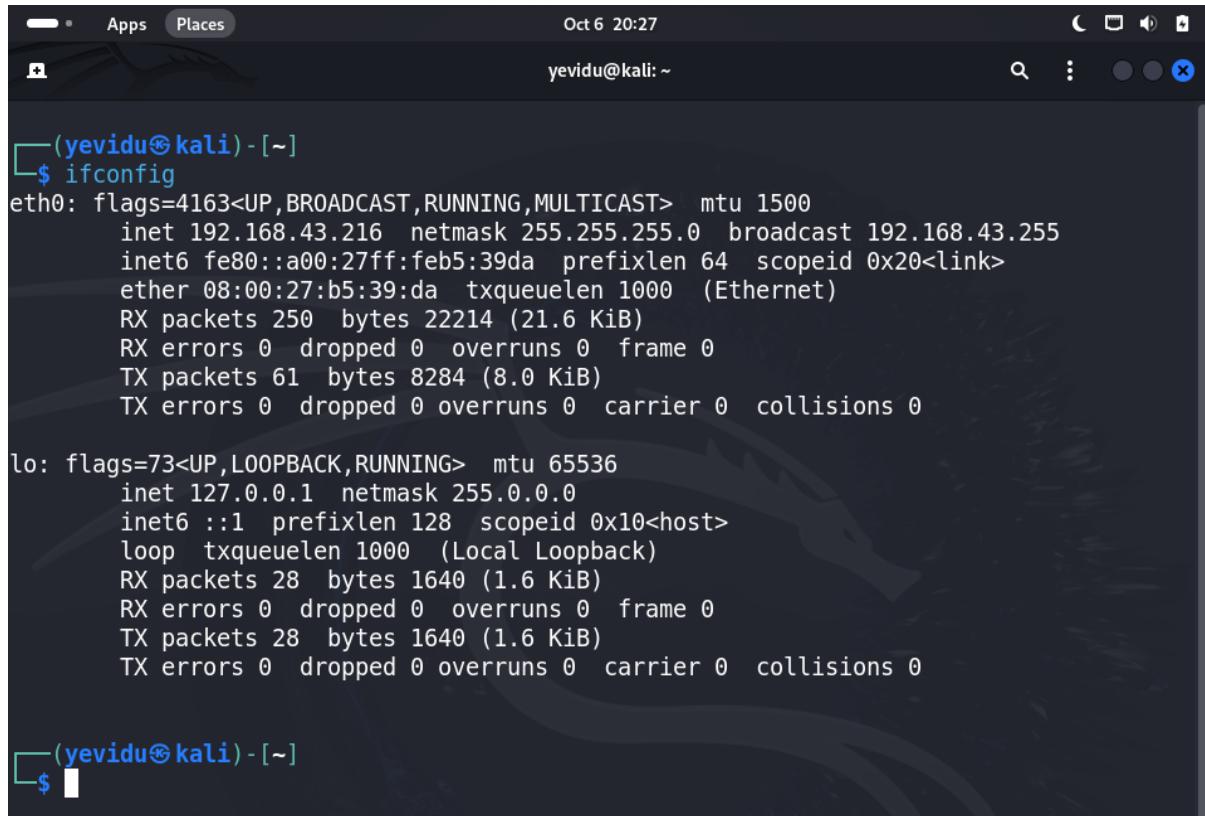
If you find any unnecessary rules, delete them by their number.

```
(yevidu㉿kali)-[~]
$ sudo ufw delete 1
Deleting:
 allow 22/tcp
Proceed with operation (y|n)? y
Rule deleted

(yevidu㉿kali)-[~]
$
```

4.3). Disable Unused Network Interface.

First view the list of all active network interfaces.

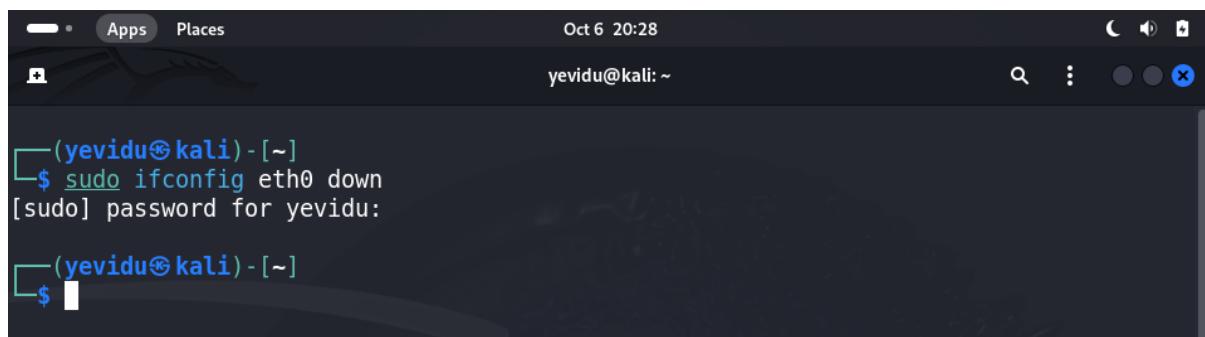


```
(yevidu㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.43.216 netmask 255.255.255.0 broadcast 192.168.43.255
        inet6 fe80::a00:27ff:feb5:39da prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:b5:39:da txqueuelen 1000 (Ethernet)
            RX packets 250 bytes 22214 (21.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 61 bytes 8284 (8.0 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 28 bytes 1640 (1.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 28 bytes 1640 (1.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(yevidu㉿kali)-[~]
$
```

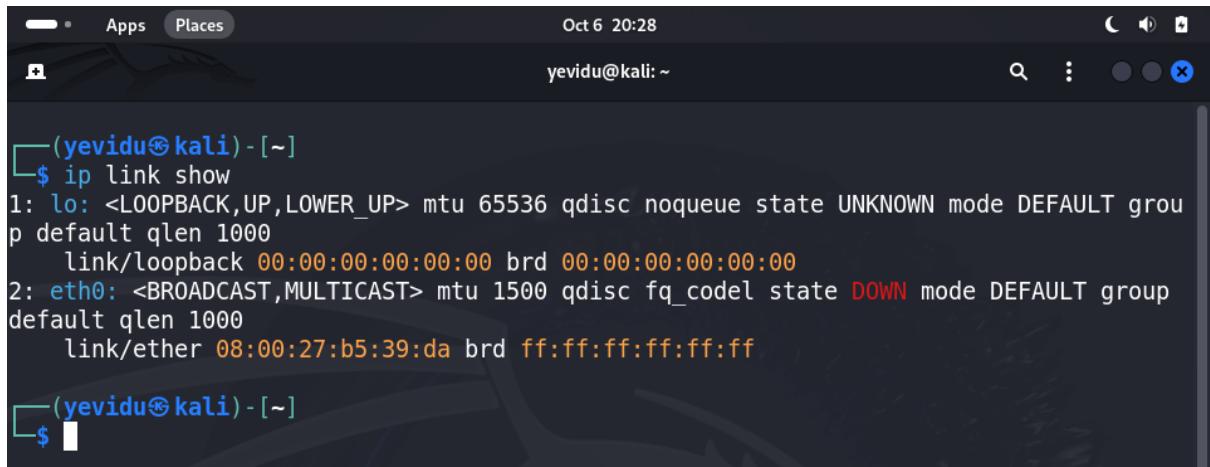
Disable the interface with name of the interface that you wish to disable.



```
(yevidu㉿kali)-[~]
$ sudo ifconfig eth0 down
[sudo] password for yevidu:

(yevidu㉿kali)-[~]
$
```

Check the current network interfaces with.

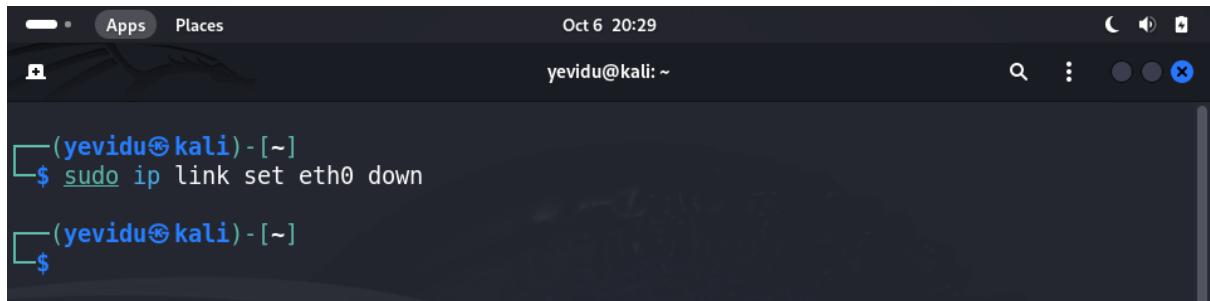


A screenshot of a terminal window on a Kali Linux desktop. The title bar shows 'Apps' and 'Places'. The status bar indicates the date as 'Oct 6 20:28' and the user as 'yevidu@kali: ~'. The terminal prompt is '(yevidu㉿kali)-[~]'. The user runs the command '\$ ip link show', which displays the following output:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN mode DEFAULT group default qlen 1000
    link/ether 08:00:27:b5:39:da brd ff:ff:ff:ff:ff:ff
```

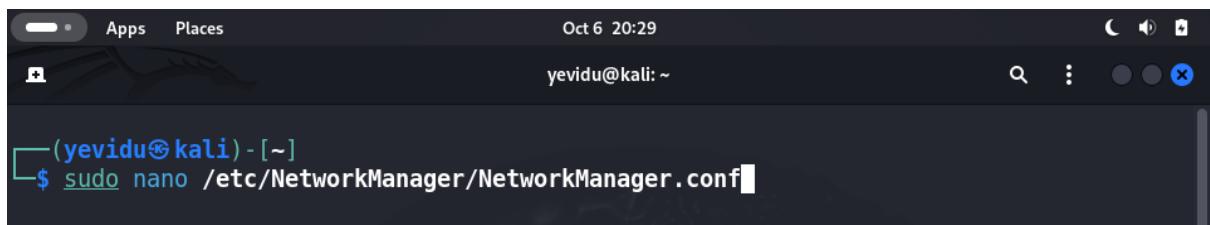
The terminal prompt then changes to '(yevidu㉿kali)-[~]' again.

Again, replace eth0 with your target interface name.



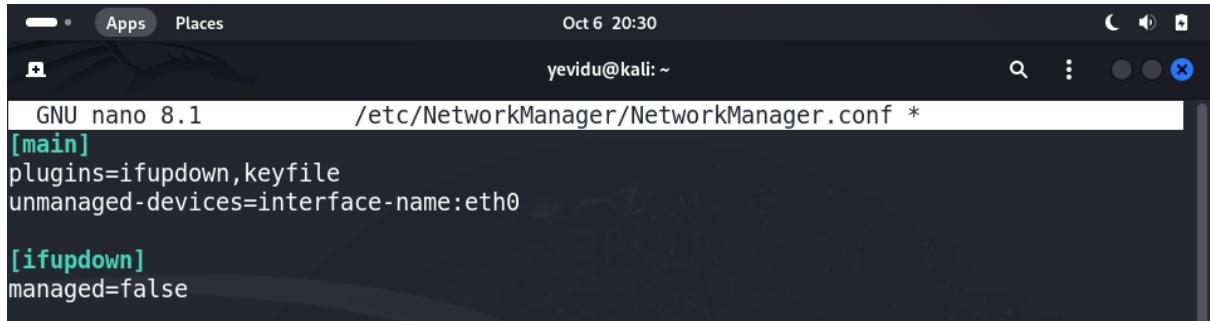
A screenshot of a terminal window on a Kali Linux desktop. The title bar shows 'Apps' and 'Places'. The status bar indicates the date as 'Oct 6 20:29' and the user as 'yevidu@kali: ~'. The terminal prompt is '(yevidu㉿kali)-[~]'. The user runs the command '\$ sudo ip link set eth0 down', which changes the interface state to 'DOWN'. The terminal prompt then changes to '(yevidu㉿kali)-[~]' again.

Open the configuration file with a text editor.



A screenshot of a terminal window on a Kali Linux desktop. The title bar shows 'Apps' and 'Places'. The status bar indicates the date as 'Oct 6 20:29' and the user as 'yevidu@kali: ~'. The terminal prompt is '(yevidu㉿kali)-[~]'. The user runs the command '\$ sudo nano /etc/NetworkManager/NetworkManager.conf', which opens the configuration file in a text editor. The terminal prompt then changes to '(yevidu㉿kali)-[~]' again.

Edit the file. Under the [main] section, add or modify the line to include your interface.

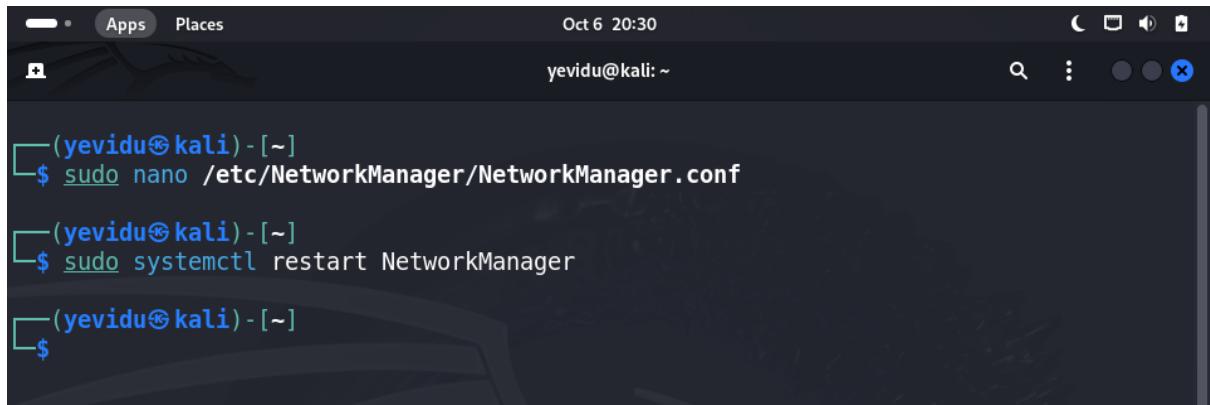


```
GNU nano 8.1          /etc/NetworkManager/NetworkManager.conf *

[main]
plugins=ifupdown,keyfile
unmanaged-devices-interface-name:eth0

[ifupdown]
managed=false
```

After saving changes, restart NetworkManager.



```
(yevidu㉿kali)-[~]
└─$ sudo nano /etc/NetworkManager/NetworkManager.conf

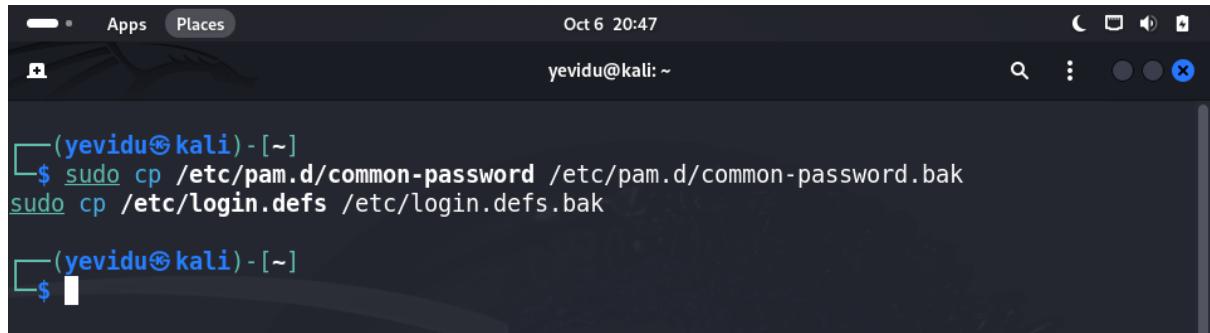
(yevidu㉿kali)-[~]
└─$ sudo systemctl restart NetworkManager

(yevidu㉿kali)-[~]
└─$
```

To re-enable a disabled interface, simply reverse the commands.

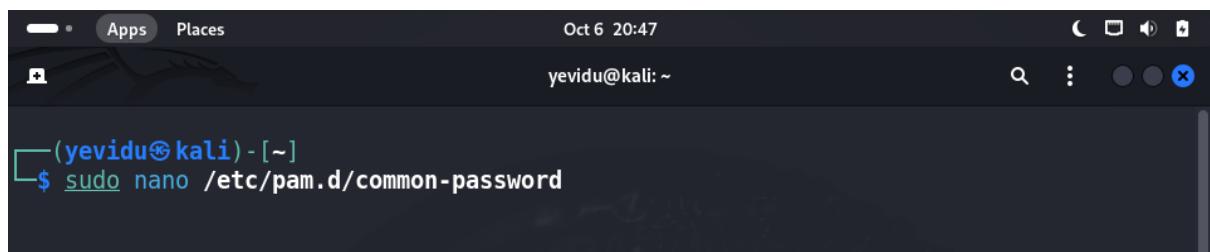
4.4). Modify the Default Password Policy Settings.

First, make backup relevant to the configuration file.



```
(yevidu㉿kali)-[~]
└─$ sudo cp /etc/pam.d/common-password /etc/pam.d/common-password.bak
└─$ sudo cp /etc/login.defs /etc/login.defs.bak
└─$ |
```

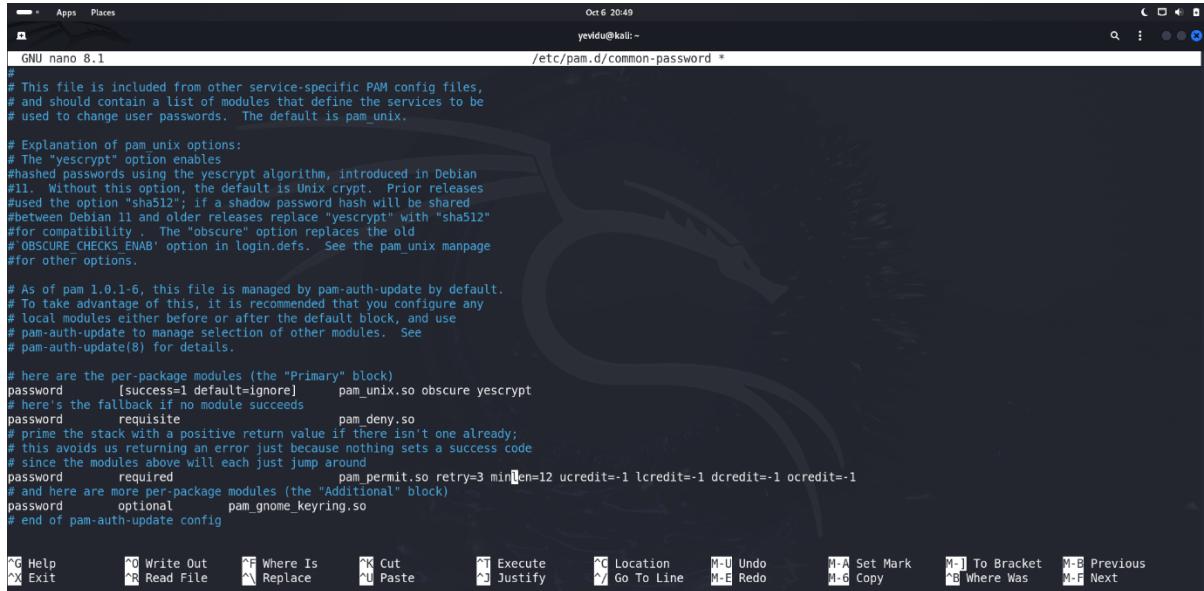
Open the file using a text editor.



```
(yevidu㉿kali)-[~]
└─$ sudo nano /etc/pam.d/common-password
```

find the line that starts with password requisite pam_pwquality.so or similar and append your desired settings.

Then edit it.



```
Oct 6 20:49
yevidu@kali:~ /etc/pam.d/common-password *

# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.

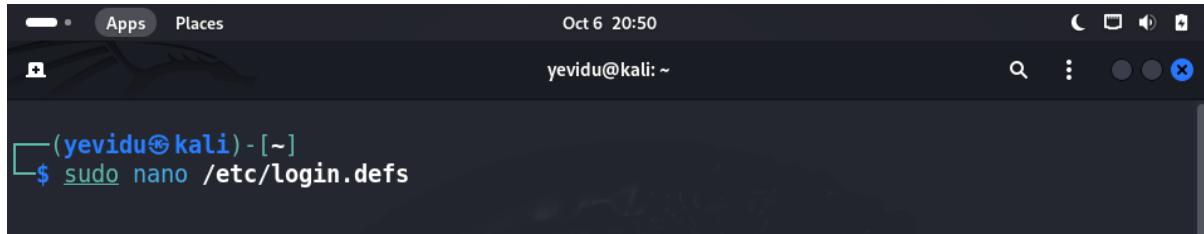
# Explanation of pam_unix options:
# The "yescrypt" option enables
# hashed passwords using the yescrypt algorithm, introduced in Debian
# #11. Without this option, the default is Unix crypt. Prior releases
# used the option "sha512"; if a shadow password hash will be shared
# between Debian 11 and older releases replace "yescrypt" with "sha512"
# for compatibility. The "obscure" option replaces the old
# "#OBSURE_CHECKS ENAB" option in login.defs. See the pam_unix manpage
# for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password [success=1 default=ignore] pam_unix.so obscure yescrypt
# here's the fallback if no module succeeds
password requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password required pam_permit.so retry=3 minlen=12 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1
# and here are more per-package modules (the "Additional" block)
password optional pam_gnome_keyring.so
# end of pam-auth-update config

PG Help      ^Q Write Out    ^F Where Is    ^K Cut        ^T Execute    ^C Location    M-U Undo    M-A Set Mark   M-J To Bracket  M-B Previous
^X Exit      ^R Read File    ^W Replace    ^U Paste     ^I Justify    ^G Go To Line  M-E Redo    M-B Copy      M-B Where Was  M-F Next
```

Open the login.defs file.



```
Oct 6 20:50
yevidu@kali:~
```

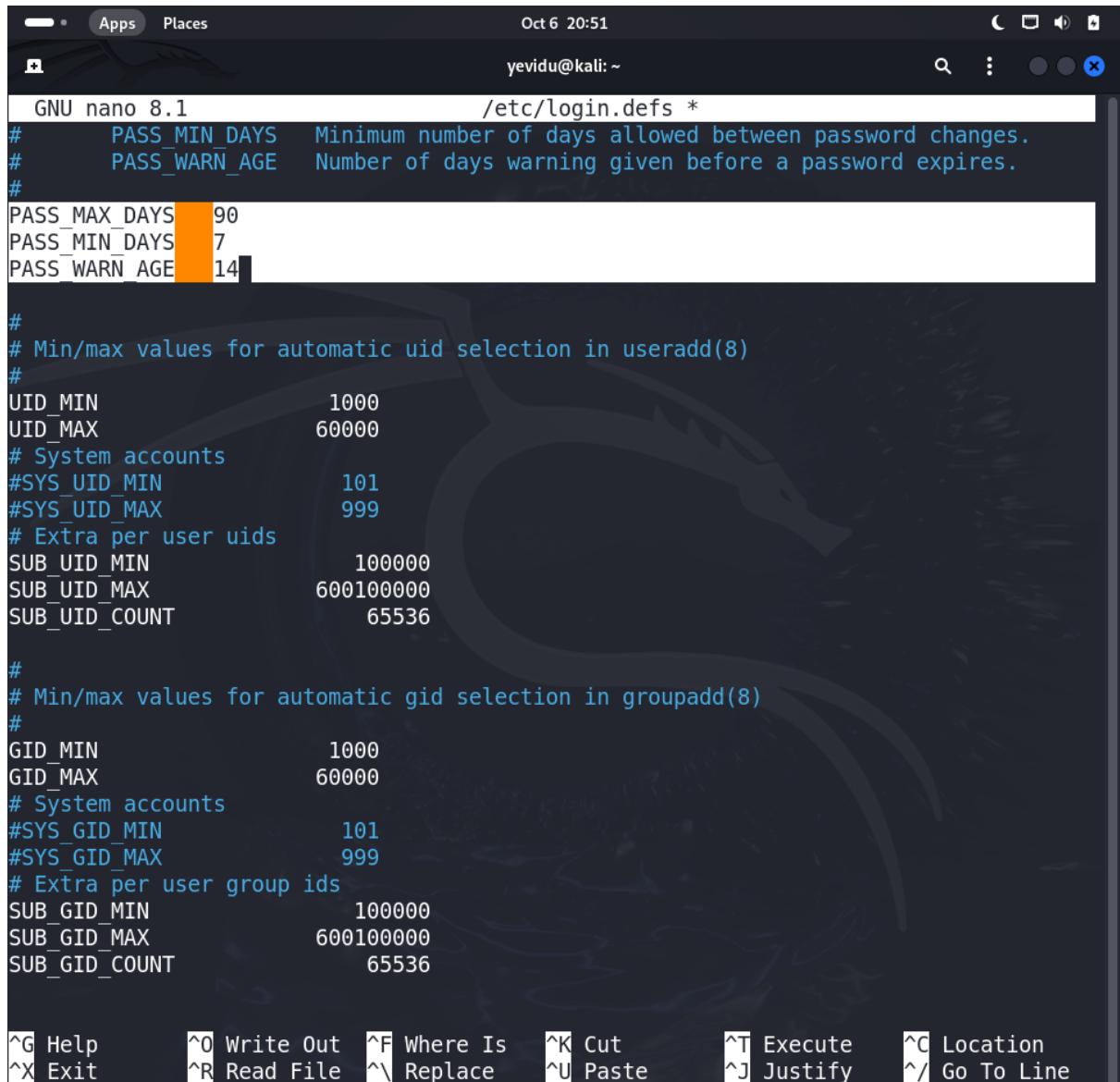
```
[yevidu@kali] - [~]
$ sudo nano /etc/login.defs
```

Modify or add the following lines according to your requirements.

PASS_MAX_DAYS 90 # Maximum days before password must be changed

PASS_MIN_DAYS 7 # Minimum days before a password can be changed

PASS_WARN_AGE 14 # Days before expiration to warn user



```
GNU nano 8.1          /etc/login.defs *

#      PASS_MIN_DAYS  Minimum number of days allowed between password changes.
#      PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS 90
PASS_MIN_DAYS 7
PASS_WARN_AGE 14

#
# Min/max values for automatic uid selection in useradd(8)
#
UID_MIN          1000
UID_MAX          60000
# System accounts
#SYS_UID_MIN     101
#SYS_UID_MAX     999
# Extra per user uids
SUB_UID_MIN      100000
SUB_UID_MAX      600100000
SUB_UID_COUNT    65536

#
# Min/max values for automatic gid selection in groupadd(8)
#
GID_MIN          1000
GID_MAX          60000
# System accounts
#SYS_GID_MIN     101
#SYS_GID_MAX     999
# Extra per user group ids
SUB_GID_MIN      100000
SUB_GID_MAX      600100000
SUB_GID_COUNT    65536

^G Help           ^O Write Out  ^F Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit          ^R Read File  ^L Replace    ^U Paste      ^J Justify   ^/ Go To Line
```

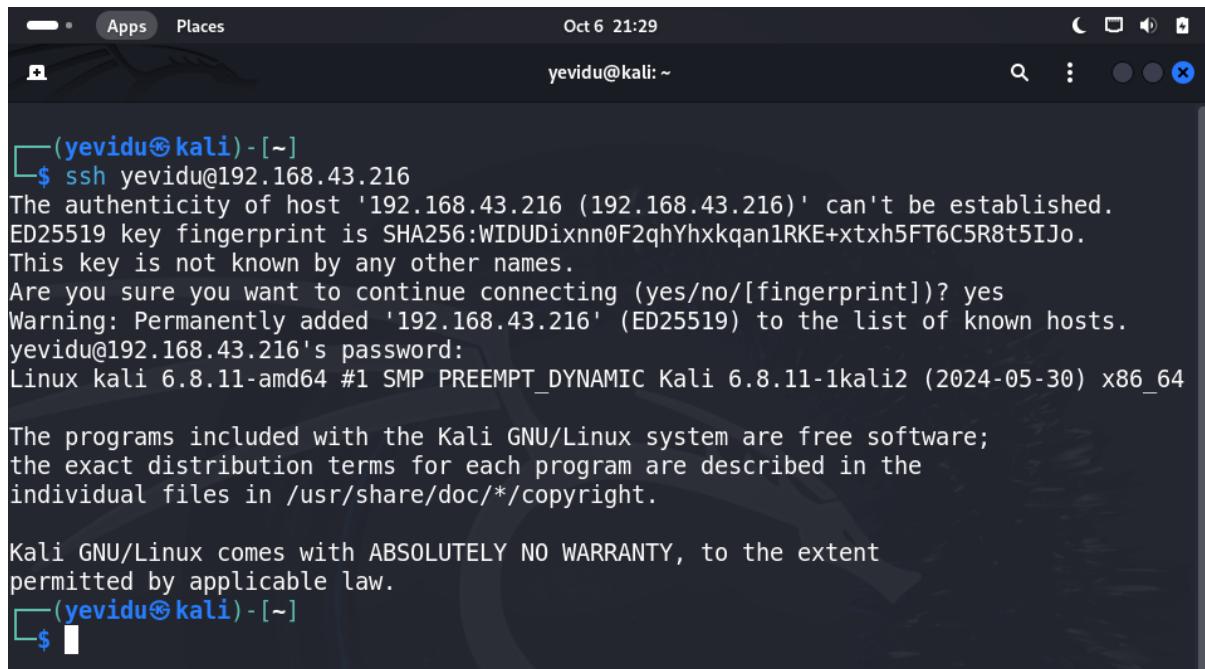
After making all necessary changes in both files, save and exit the text editor.

4.5). Change Default SSH Port.

First connect to your server using the following command.

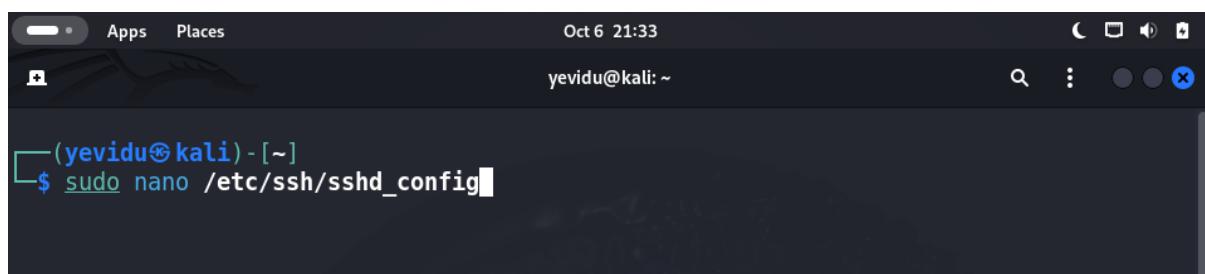
ssh username@server_ip

Replace username with your actual username and server_ip with the IP address of your server.



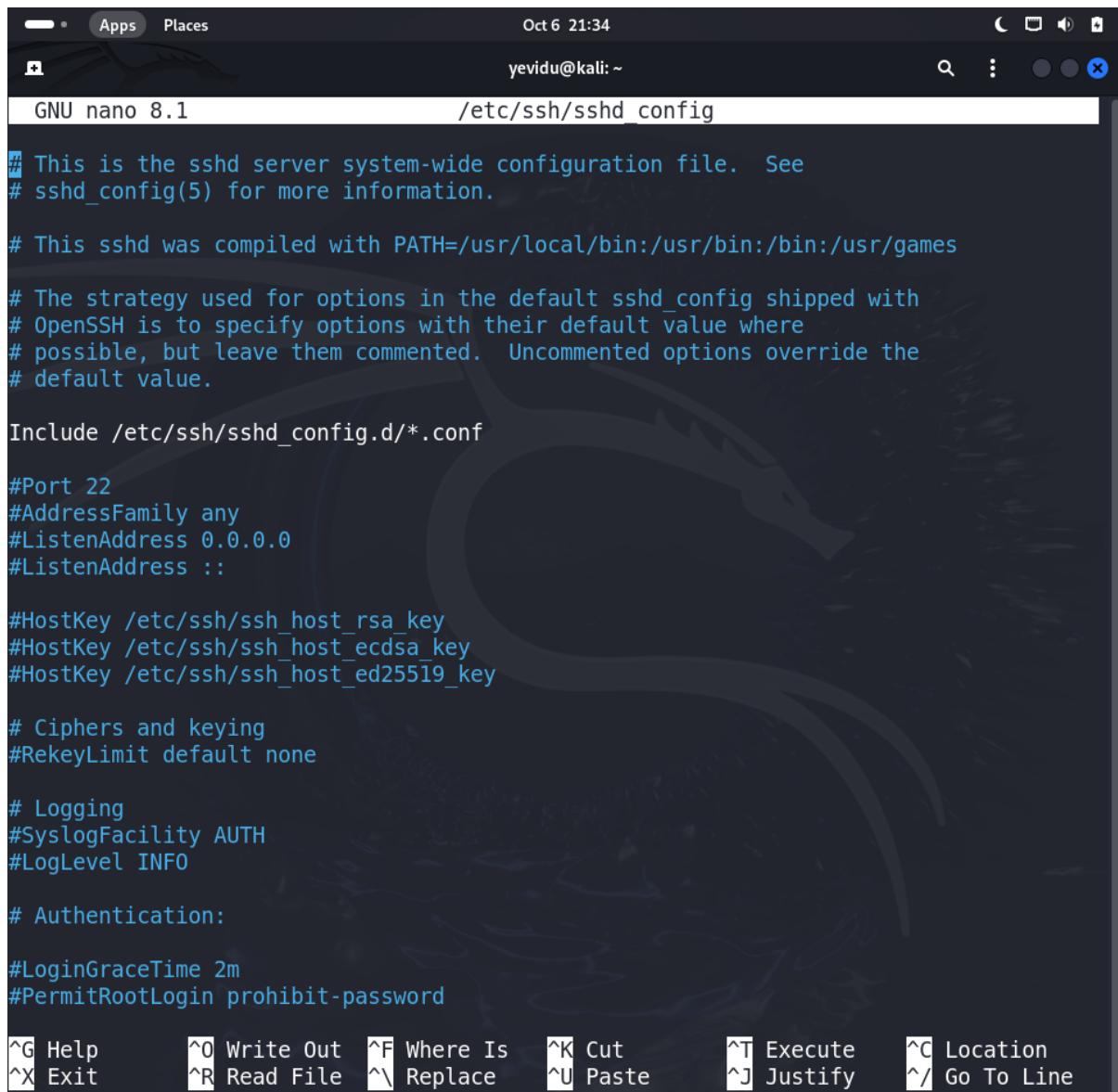
The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar indicates the user is 'yevidu' at 'kali' on 'Oct 6 21:29'. The terminal prompt is '(yevidu㉿kali)-[~]'. The user runs the command 'ssh yevidu@192.168.43.216'. The system prompts for confirmation about the host's fingerprint, which the user accepts ('yes'). It then asks for the password, which is also accepted. The terminal then displays the standard Kali Linux welcome message, including the kernel version 'Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64'. Finally, it shows the copyright notice for the software included in the distribution.

Open the SSH daemon configuration file using a text editor.



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar indicates the user is 'yevidu' at 'kali' on 'Oct 6 21:33'. The terminal prompt is '(yevidu㉿kali)-[~]'. The user runs the command 'sudo nano /etc/ssh/sshd_config'. The terminal shows the command being typed, with the cursor visible at the end of the path.

Uncomment this line by removing the # symbol and change 22 to your desired port number.

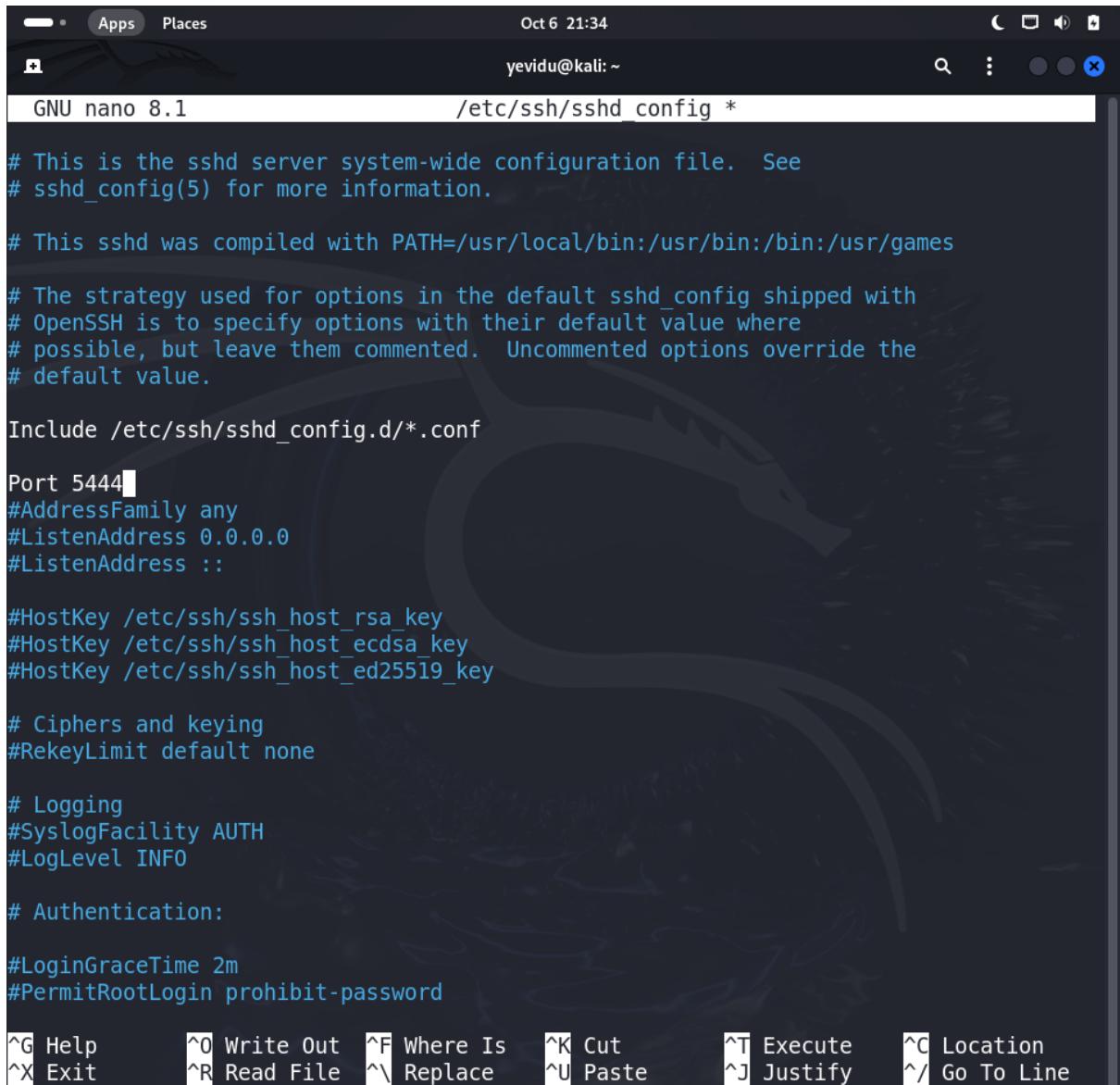


The screenshot shows a terminal window titled "GNU nano 8.1" with the file "/etc/ssh/sshd_config" open. The terminal interface includes a header bar with "Oct 6 21:34" and "yevidu@kali: ~". The window contains the configuration file for the sshd server. A specific line is highlighted with a red rectangle:

```
# Port 22
```

The rest of the configuration file includes comments about host keys, ciphers, logging, authentication, and root login settings. At the bottom of the terminal window, there is a menu of keyboard shortcuts:

$\wedge G$ Help	$\wedge O$ Write Out	$\wedge F$ Where Is	$\wedge K$ Cut	$\wedge T$ Execute	$\wedge C$ Location
$\wedge X$ Exit	$\wedge R$ Read File	$\wedge R$ Replace	$\wedge U$ Paste	$\wedge J$ Justify	$\wedge /$ Go To Line



```
GNU nano 8.1          /etc/ssh/sshd_config *

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 5444
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

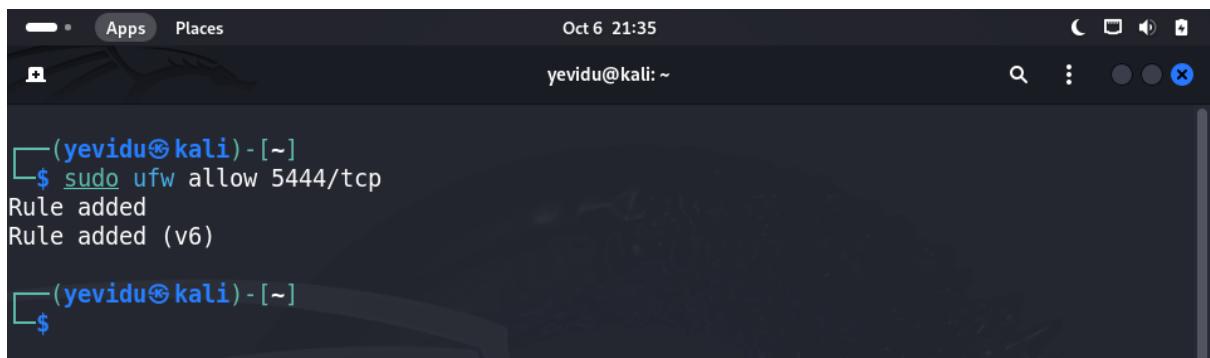
# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password

^G Help      ^O Write Out  ^F Where Is  ^K Cut      ^T Execute   ^C Location
^X Exit     ^R Read File  ^V Replace   ^U Paste    ^J Justify   ^Y Go To Line
```

Then save and exit from the editor.

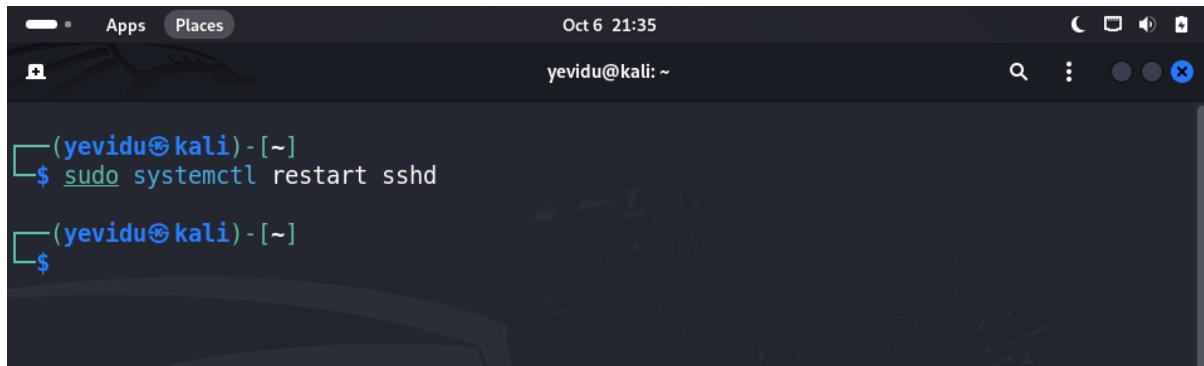
Configure Your Firewall.



```
(yevidu㉿kali)-[~]
└$ sudo ufw allow 5444/tcp
Rule added
Rule added (v6)

(yevidu㉿kali)-[~]
└$
```

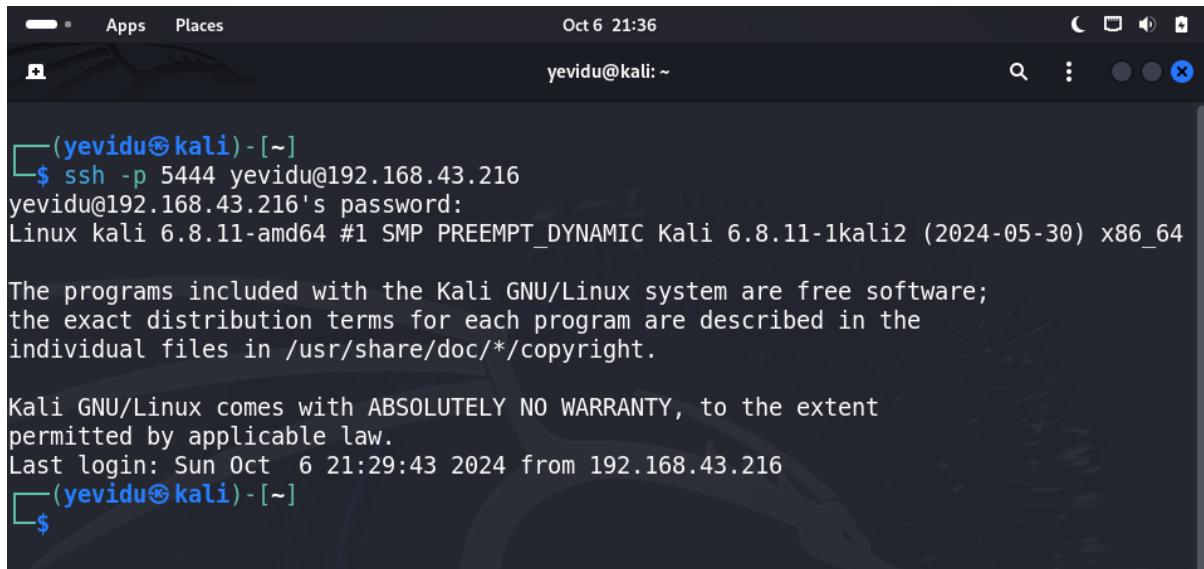
Restart the SSH Service.



```
(yevidu㉿kali)-[~]
$ sudo systemctl restart sshd

(yevidu㉿kali)-[~]
$
```

Test the New Port.



```
(yevidu㉿kali)-[~]
$ ssh -p 5444 yevidu@192.168.43.216
yevidu@192.168.43.216's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Oct  6 21:29:43 2024 from 192.168.43.216
(yevidu㉿kali)-[~]
$
```