

Sri Lanka Institute of Information Technology



Module: IE2012

Year 2, Semester 1

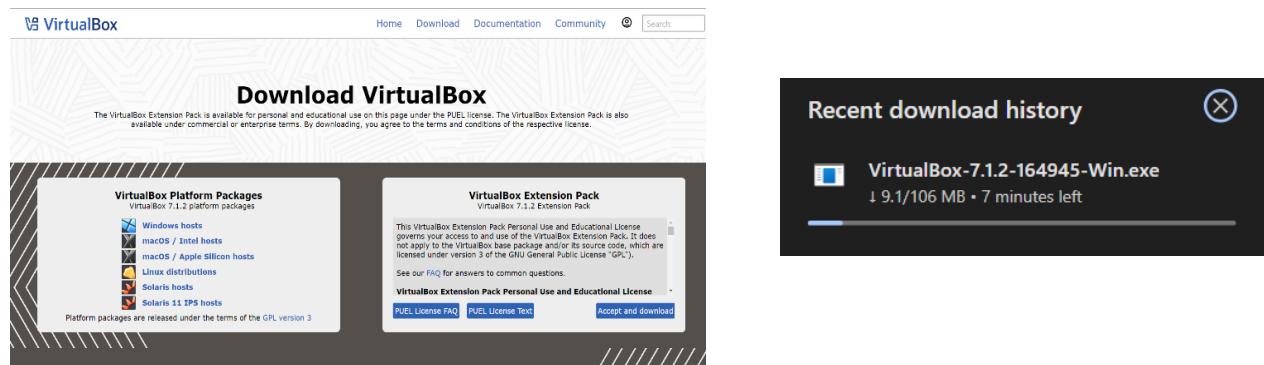
Systems and Networking Programming

S M F Hana IT23255142

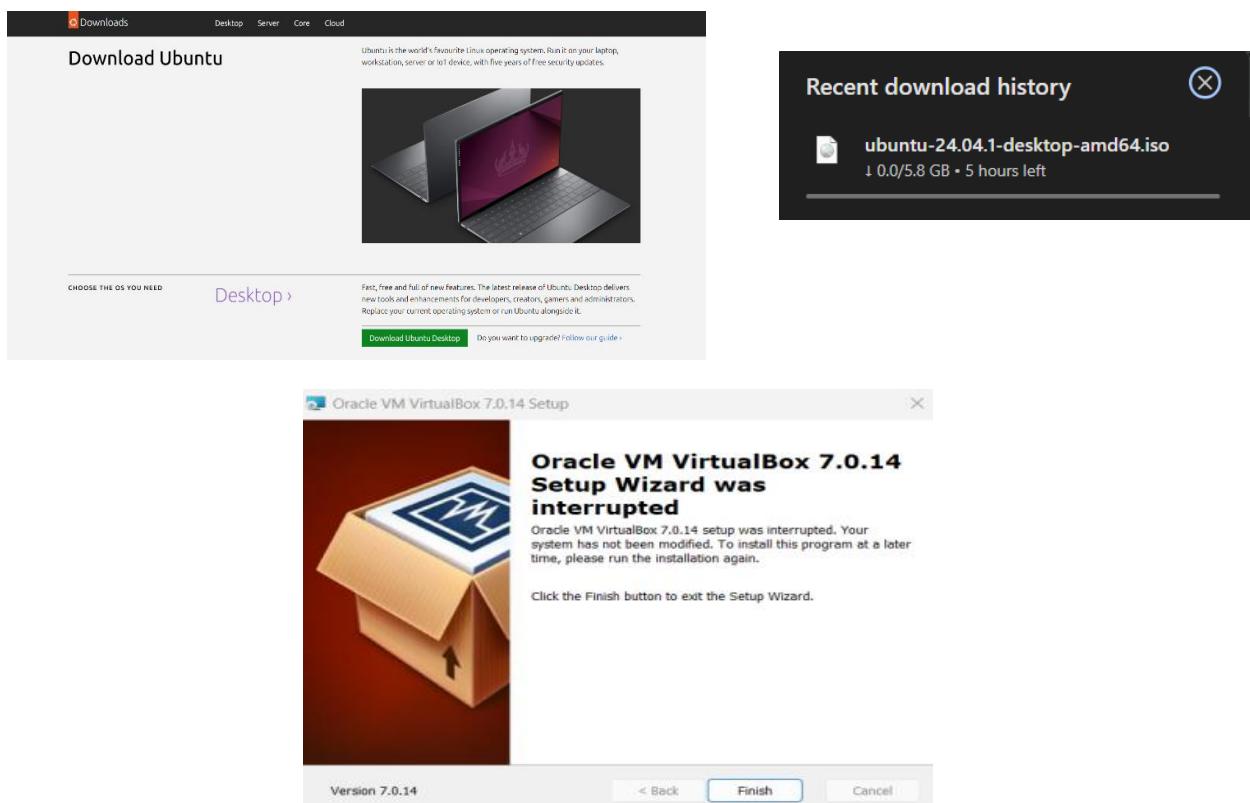
01.Basic Of Linux

01.1Virtual Machine

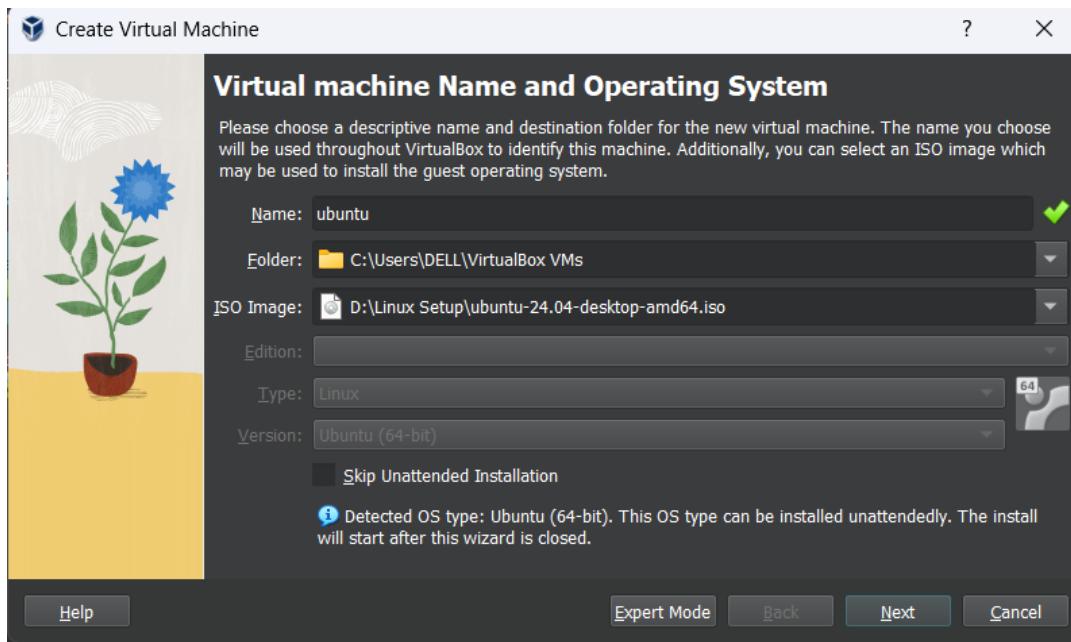
Downloaded the virtual machine after choosing the windows hosts version of the virtual platform package from the official VirtualBox website, <https://www.virtualbox.org/>.



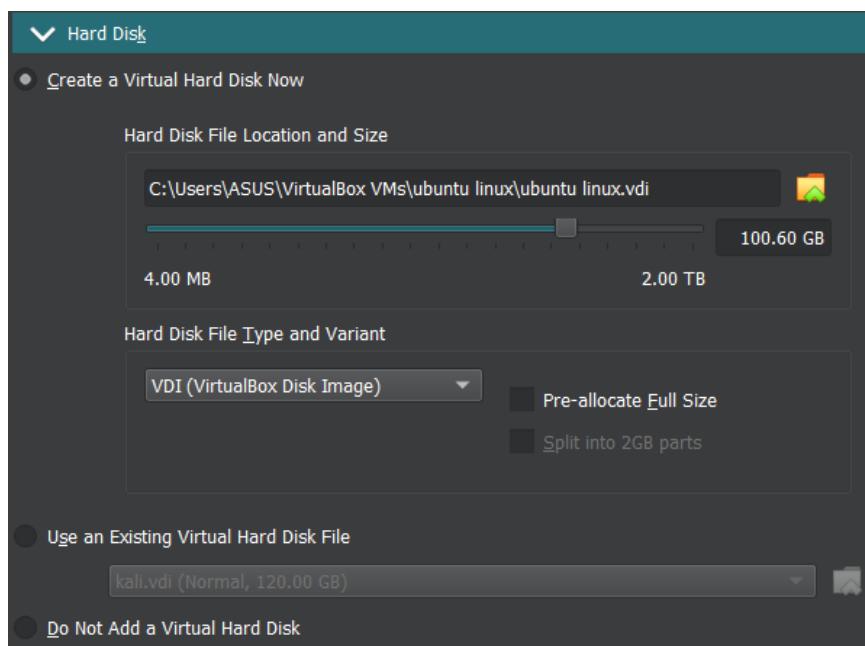
Downloaded the Linux-based system, Ubuntu desktop from <https://ubuntu.com/download>. Selected the version 24.04.1 LTS



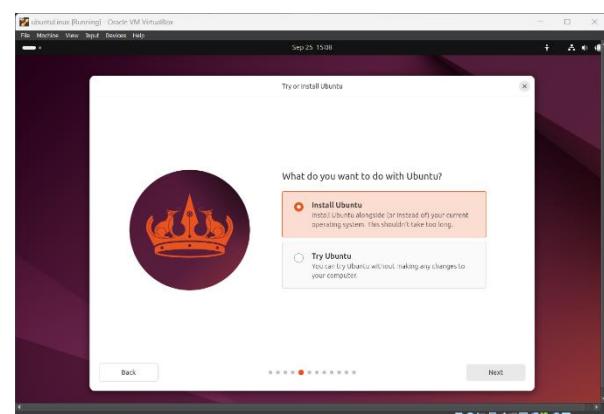
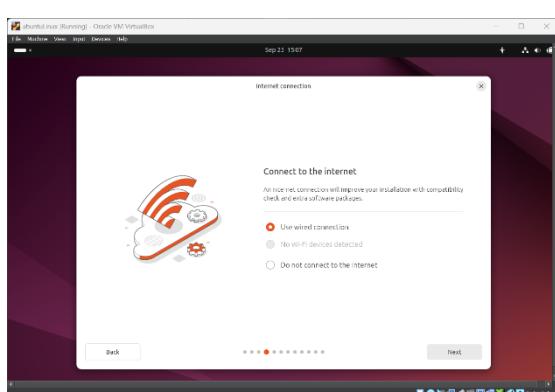
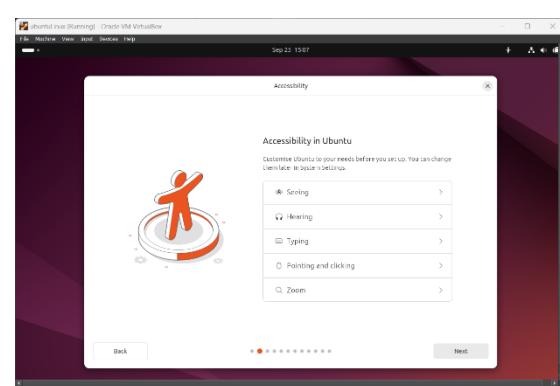
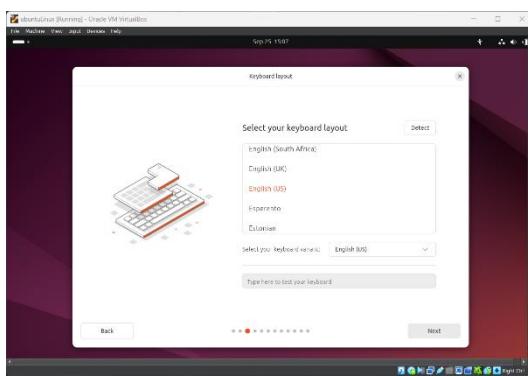
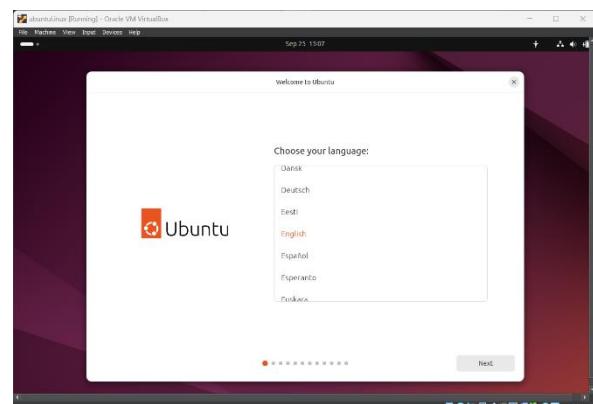
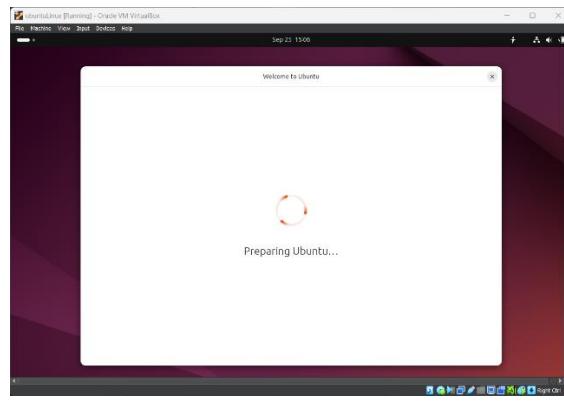
After the installation of virtual machine, it was opened and the **new** button was clicked. Then typed the name field as **ubuntu** as the VM name and assigned the path to store the VM files. For installation ISO image was selected by giving *other* and navigating to the previously mentioned ubuntu setup file

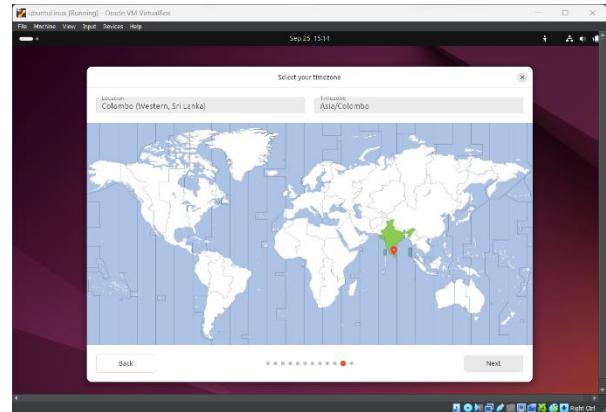
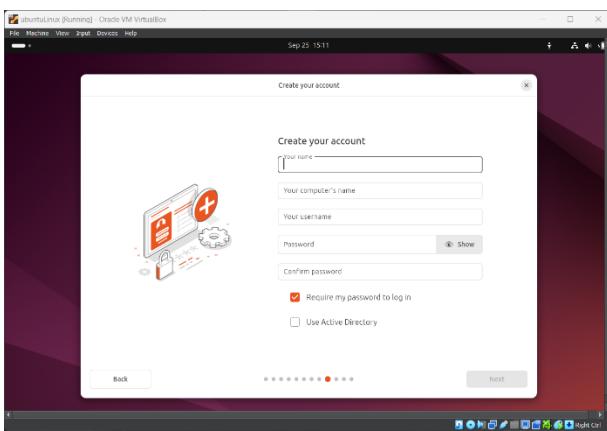
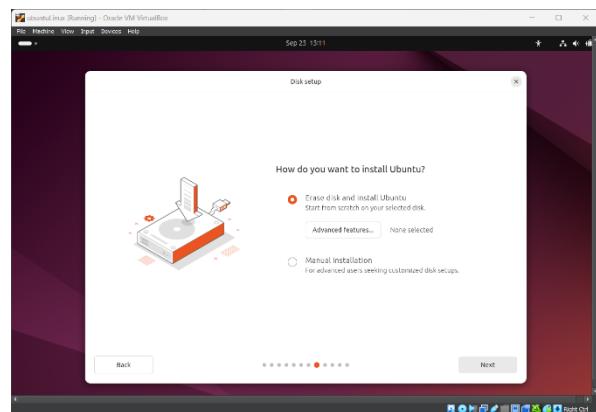
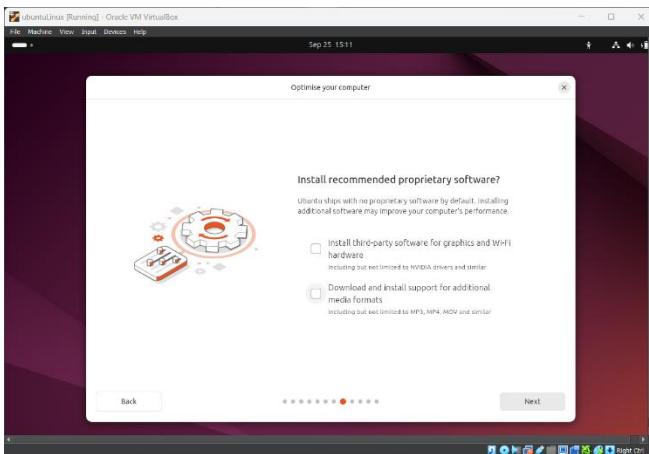
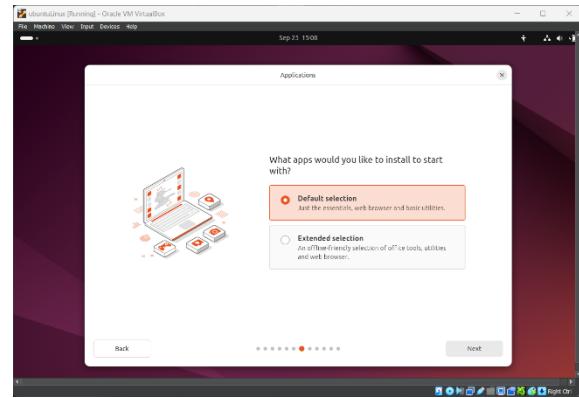
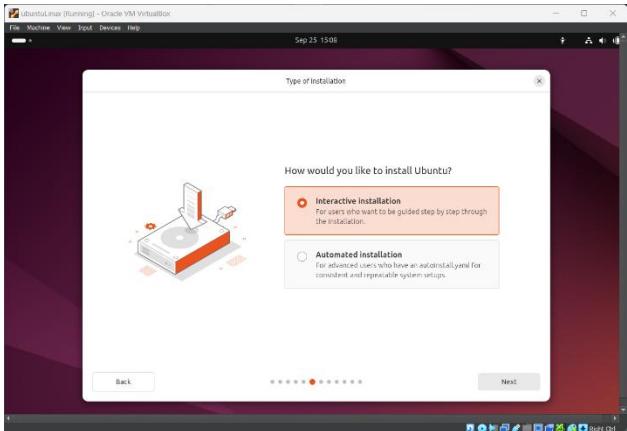


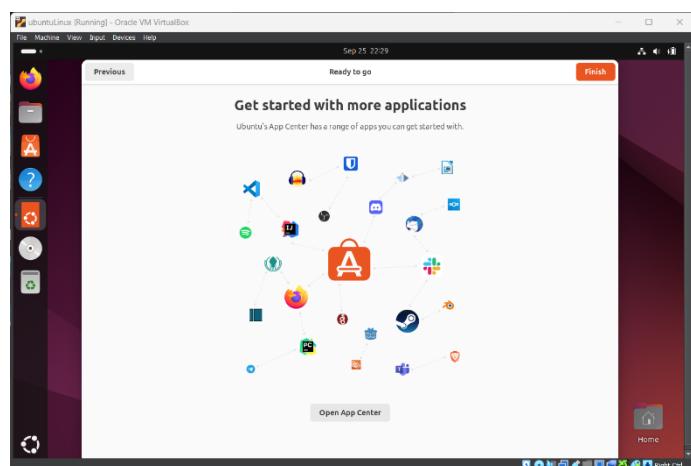
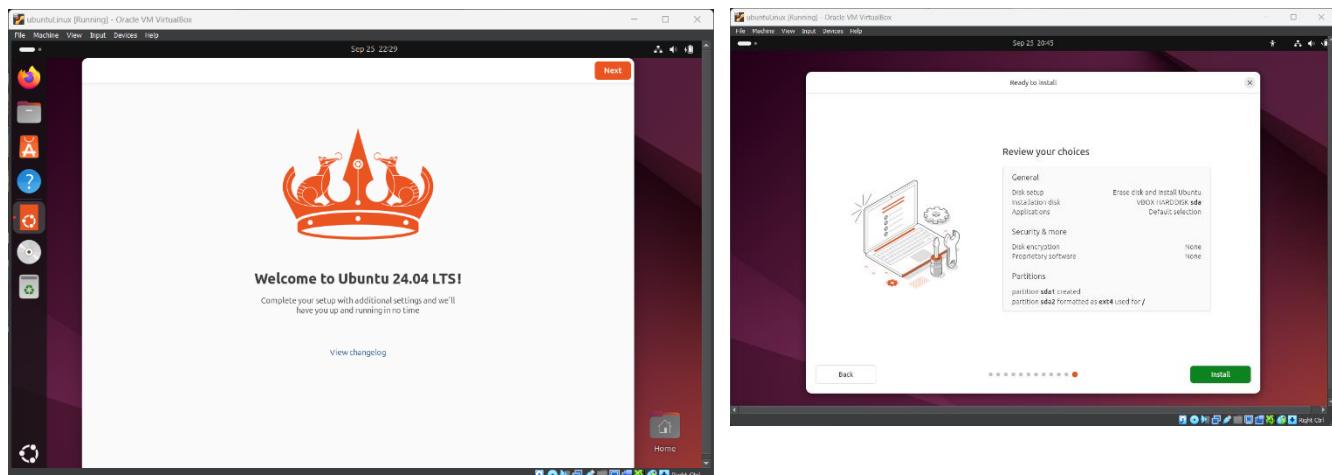
After clicking **Next**, **Base Memory** was set to 4000 MB and **Processors** was set to 7. Set the size as follows and clicked finish



Thereafter, followed the following steps







02.2 Command Line Introduction

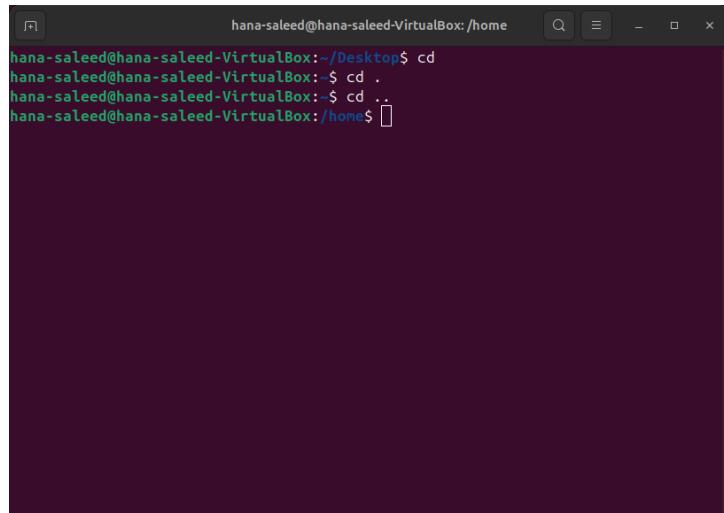
I. Basic Navigation Commands

a. cd (Change Directory) – Used to navigate between directories in the file system

`cd` : Navigates to home directory

`cd .` : Stays in the current directory

`cd ..` : Moves up one directory level



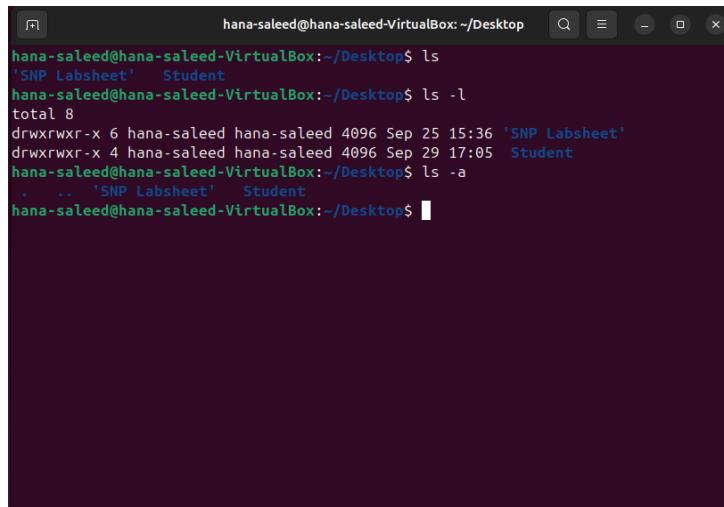
A terminal window titled "hana-saleed@hana-saleed-VirtualBox: ~/Desktop". The user enters three commands: "cd", "cd .", and "cd ..". The output shows the user's path changing from ~/Desktop to ~/home.

```
hana-saleed@hana-saleed-VirtualBox:~/Desktop$ cd
hana-saleed@hana-saleed-VirtualBox:~$ cd .
hana-saleed@hana-saleed-VirtualBox:~$ cd ..
hana-saleed@hana-saleed-VirtualBox:~/home$
```

b. ls (list) – Lists the files and the directories in the current directory

i. `ls -l` : Display a detailed list

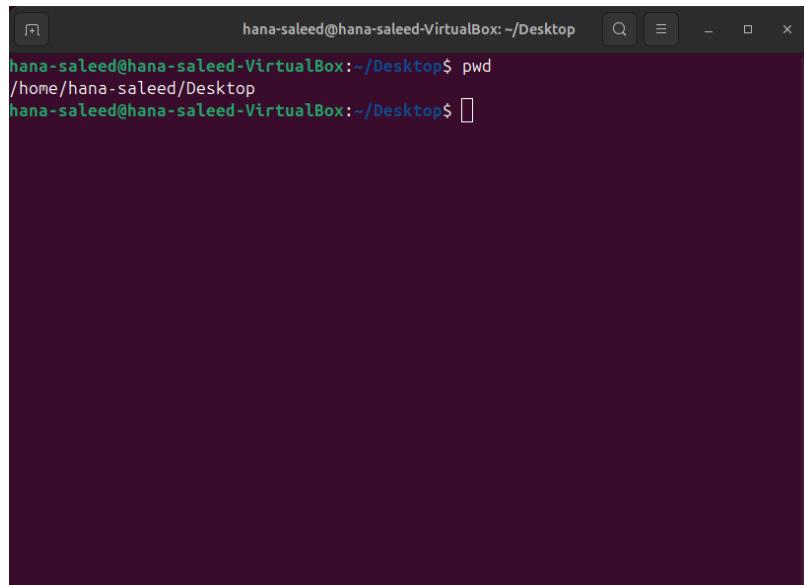
ii. `ls -a` : Display files including the hidden files



A terminal window titled "hana-saleed@hana-saleed-VirtualBox: ~/Desktop". The user runs two `ls` commands: one without options and one with `-l`, followed by one with `-a`. The output lists files and directories including hidden ones.

```
hana-saleed@hana-saleed-VirtualBox:~/Desktop$ ls
'SNP Labsheet'  Student
hana-saleed@hana-saleed-VirtualBox:~/Desktop$ ls -l
total 8
drwxrwxr-x 6 hana-saleed hana-saleed 4096 Sep 25 15:36 'SNP Labsheet'
drwxrwxr-x 4 hana-saleed hana-saleed 4096 Sep 29 17:05 Student
hana-saleed@hana-saleed-VirtualBox:~/Desktop$ ls -a
.  ..  'SNP Labsheet'  Student
hana-saleed@hana-saleed-VirtualBox:~/Desktop$
```

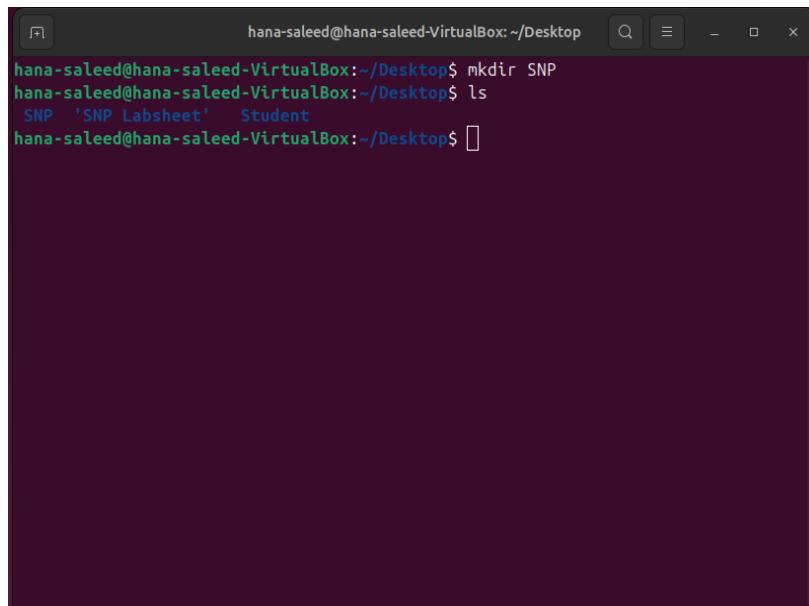
c. pwd (print work directory) – Displays the full path of the current directory



```
hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop$ pwd
/home/hanna-saleed/Desktop
hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop$
```

A screenshot of a terminal window titled "hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop". The window has a dark background and light-colored text. It shows the user's name, the host name, the IP address, and the current working directory. The user then types the command "pwd" and presses Enter. The terminal displays the full path of the current directory, which is "/home/hanna-saleed/Desktop". The user then types another command and presses Enter again.

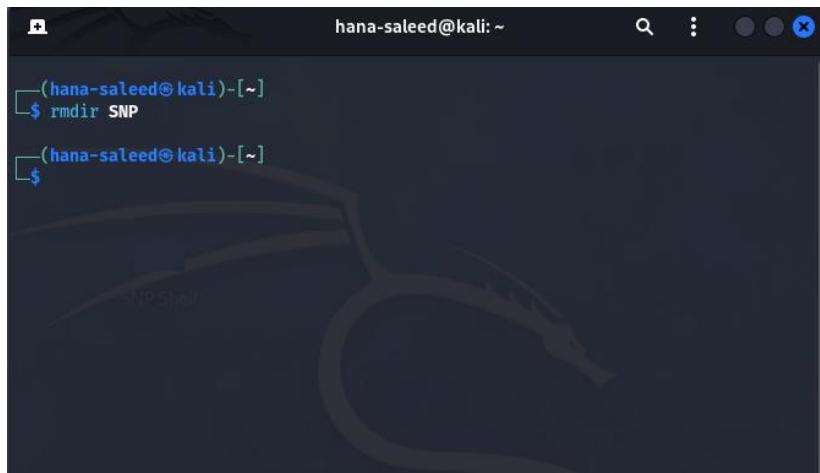
d. mkdir (make directory) – Creates a new directory in the current location. We can verify the creation of the directory with **mkdir** after it appears in the list when the **ls** command is run



```
hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop$ mkdir SNP
hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop$ ls
SNP 'SNP Labsheet' Student
hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop$
```

A screenshot of a terminal window titled "hanna-saleed@hanna-saleed-VirtualBox: ~/Desktop". The user types the command "mkdir SNP" and presses Enter. Then, the user types "ls" and presses Enter. The terminal lists the contents of the current directory, which include the newly created directory "SNP", a file named "SNP Labsheet", and a file named "Student". The user then types another command and presses Enter again.

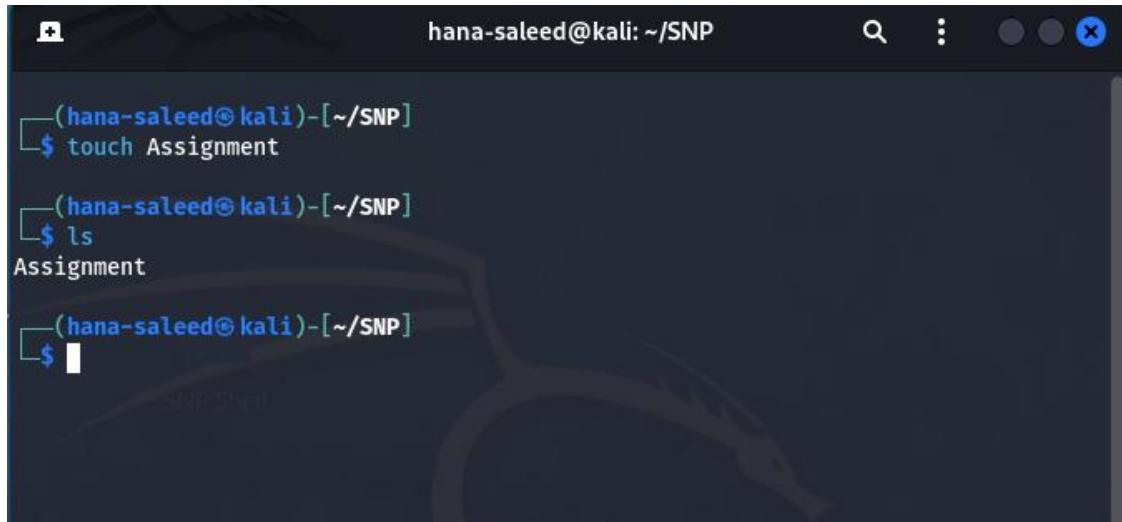
d. rmdir (remover directory) - Removes or deletes an empty directory from the file system.



A screenshot of a terminal window titled "hana-saleed@kali: ~". The window shows a command-line session where the user runs the command "rmdir SNP". The terminal has a dark background with a faint dragon logo watermark. The command is entered at the prompt, followed by a confirmation message and a new prompt.

```
(hana-saleed㉿kali)-[~]
$ rmdir SNP
(hana-saleed㉿kali)-[~]
$
```

e. touch – Creates a new empty file or update timestamps. The filename has to be specified after the command. The syntax follows as **touch *filename***



A screenshot of a terminal window titled "hana-saleed@kali: ~/SNP". The window shows a command-line session where the user runs the command "touch Assignment". They then run "ls" to list the contents of the directory, which shows the newly created file "Assignment". The terminal has a dark background with a faint dragon logo watermark. The commands are entered at the prompt, followed by their respective outputs and a new prompt.

```
(hana-saleed㉿kali)-[~/SNP]
$ touch Assignment
(hana-saleed㉿kali)-[~/SNP]
$ ls
Assignment
(hana-saleed㉿kali)-[~/SNP]
$
```

f. cp (copy) – Copies files or directories from one location to another. The syntax of the command is **cp *source destination***, where **source** is the file or directory that has to be copied and **destination** is the destination of the file or directory where the file needs to be copied.

```
hanna-saleed@kali: ~/SNP
(hanna-saleed@kali)-[~/SNP]
$ cp Assignment Assignment2

(hanna-saleed@kali)-[~/SNP]
$ ls
Assignment Assignment2

(hanna-saleed@kali)-[~/SNP]
$
```

g. mv (move) – Moves or renames files and directories. To move the **mv source path_to_new_destination**, where **source the** is the file or directory that has to be moved and **path_to_new_destination** is the new location of the file. To rename the file **mv filename new_filename**, where **filename** renames to **new_filename**.

```
hanna-saleed@Kali: ~/Desktop/Assignment
(hanna-saleed@kali)-[~/Desktop/Assignment]
$ mv Assignment.txt Assignment2.txt

(hanna-saleed@kali)-[~/Desktop/Assignment]
$
```

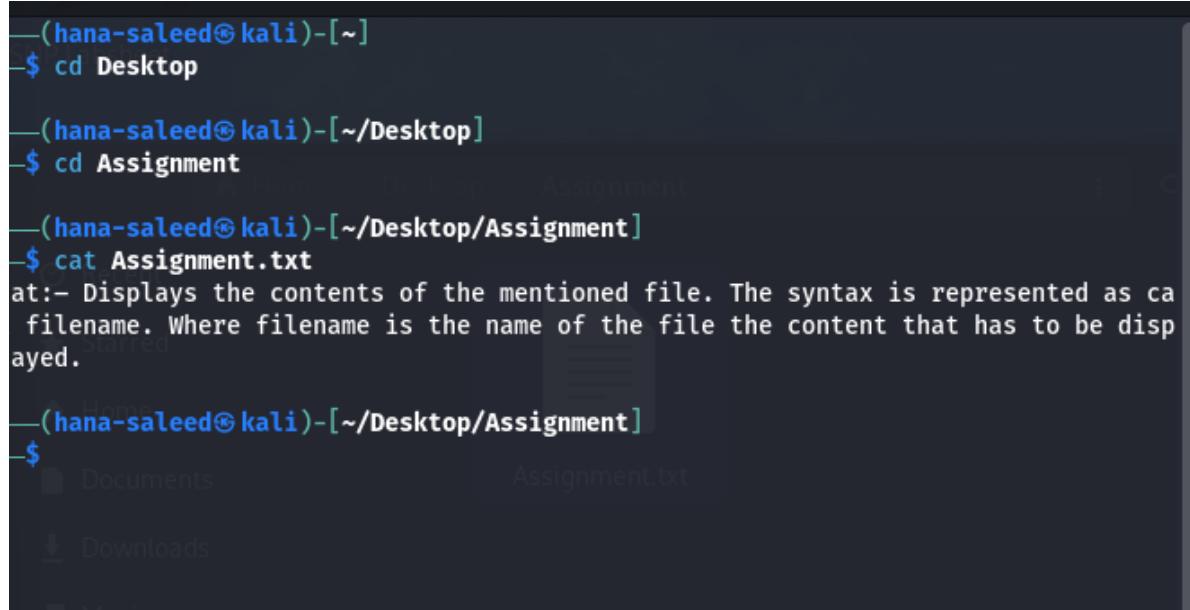
h.cat (concatenate) – Displays the contents of the mentioned file. The syntax is represented as **cat filename**. Where **filename** is the name of the file the content that has to be displayed.

```
—(hana-saleed㉿kali)-[~]
$ cd Desktop

—(hana-saleed㉿kali)-[~/Desktop]
$ cd Assignment

—(hana-saleed㉿kali)-[~/Desktop/Assignment]
$ cat Assignment.txt
at:- Displays the contents of the mentioned file. The syntax is represented as ca
filename. Where filename is the name of the file the content that has to be disp
ayed.

—(hana-saleed㉿kali)-[~/Desktop/Assignment]
$
```



i. **more** – Displays the contents of a file one screen at a time

j. **head** – Displays the first few lines in a file

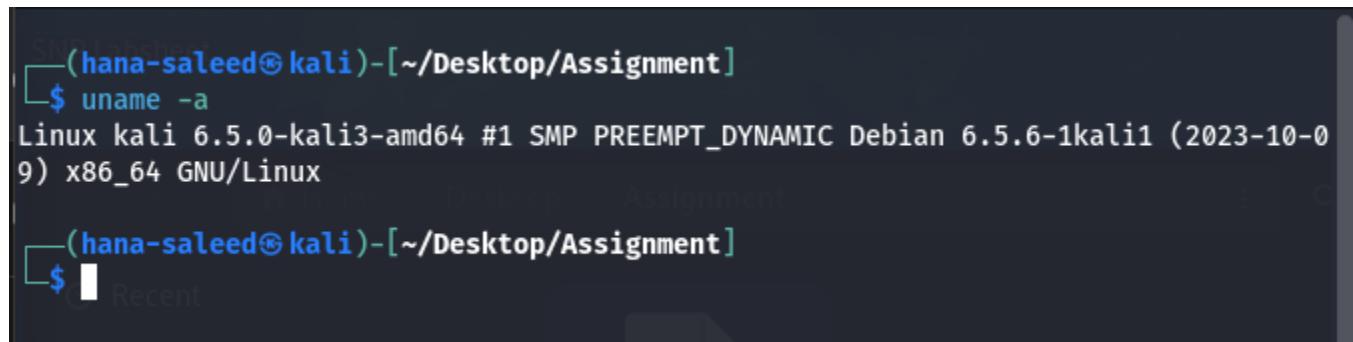
k. **tail** – Displays the last few lines in a file

02.3 System Information and User Management

I. **uname -a** – Displays details of the system and kernel. The **-a** stands for all which provides a summary of the system's configuration and environment.

```
—(hana-saleed㉿kali)-[~/Desktop/Assignment]
$ uname -a
Linux kali 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1 (2023-10-0
9) x86_64 GNU/Linux

—(hana-saleed㉿kali)-[~/Desktop/Assignment]
$
```



m. cat /proc/version - Displays the information about the currently running kernel and the system. **/proc/version** file contains details about the Linux kernel version.

```
(hana-saleed㉿kali)-[~]
$ sudo cat /proc/version

[sudo] password for hana-saleed:
Linux version 6.5.0-kali3-amd64 (devel@kali.org) (gcc-13 (Debian 13.2.0-4) 13.2.0,
GNU ld (GNU Binutils for Debian) 2.41) #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1
(2023-10-09)
```

The terminal window shows the command `sudo cat /proc/version` being run. The output displays the kernel version and configuration details. A tooltip provides a brief explanation of the `cat` command.

n. df -h (disk filesystem) – Displays the disk space usage for all mounted filesystems in human readable format. **df** reports the amount of disk space used and available in the filesystem while **-h** stands for human-readable which helps to display the output in a more understandable way.

```
(hana-saleed㉿kali)-[~]
$ df -h

Filesystem      Size  Used Avail Use% Mounted on
udev            1.5G    0   1.5G  0% /dev
tmpfs           303M  1.3M  302M  1% /run
/dev/sda1        58G  15G   41G  27% /
tmpfs           1.5G    0   1.5G  0% /dev/shm
tmpfs           5.0M    0   5.0M  0% /run/lock
tmpfs           1.0M    0   1.0M  0% /run/credentials/systemd-journald.service
tmpfs           1.0M    0   1.0M  0% /run/credentials/systemd-udev-load-credentia
ls.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-tmpfiles-setup-dev.
service
tmpfs           1.5G    0   1.5G  0% /tmp
tmpfs           1.0M    0   1.0M  0% /run/credentials/systemd-tmpfiles-setup.serv
ice
tmpfs           303M  156K  303M  1% /run/user/1000

(hana-saleed㉿kali)-[~]
```

o. free -m – Displays information about total, used and available space of physical memory and swap memory. By default, **free** command will print the memory details in kilo bytes. **-m** stands for megabytes. This will print the details in megabytes.

```
(hana-saleed㉿kali)-[~]
$ free -m
      total        used        free      shared  buff/cache   available
Mem:       3022       1650        669          20       898       1372
Swap:        974          0        974

(hana-saleed㉿kali)-[~]
$
```

p. id – Displays the user and group details for mentioned user as **id username**. When username is not specified it displays the details of the current logged in user.

```
(hana-saleed㉿kali)-[~]
$ id
uid=1000(hana-saleed) gid=1000(hana-saleed) groups=1000(hana-saleed),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),100(users),101(netdev),113(wireshark),116(bluetooth),127(scanner),135(vboxsf),136(kabober)
Recent
Starred

(hana-saleed㉿kali)-[~]
$
```

q. whoami – Displays the username of the user that has currently logged in.

```
SNDalsa
(hana-saleed㉿kali)-[~]
$ whoami
hana-saleed

(hana-saleed㉿kali)-[~] Desktop / Assignment
$ |
Recent
```

r. passwd – This command is used to change the user password. By default, the system will prompt to enter the current password subsequently prompting to enter the new password

```
SN-Labshare:(hana-saleed㉿kali)-[~]
$ passwd
Changing password for hana-saleed.
Current password:
New password:
```

s. useradd – This command is used to create new user account

```
(hana-saleed㉿kali)-[~]
$ useradd
Usage: useradd [options] LOGIN
      useradd -D
      useradd -D [options]

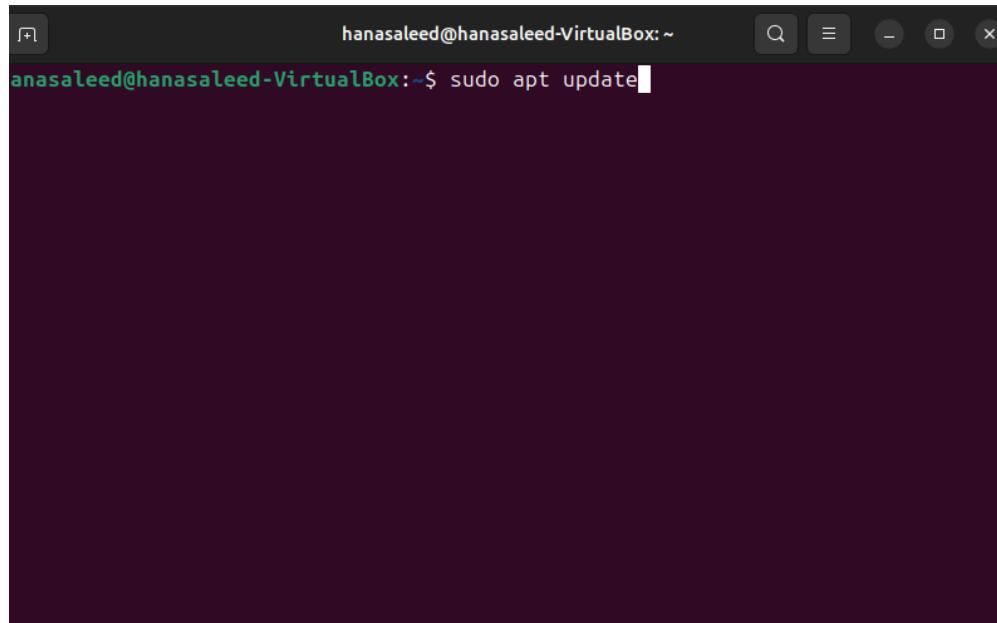
Options:
  --badname          do not check for bad names
  -b, --base-dir BASE_DIR   base directory for the home directory of the
                             new account
  --btrfs-subvolume-home    use BTRFS subvolume for home directory
  -c, --comment COMMENT     GECOS field of the new account
  -d, --home-dir HOME_DIR   home directory of the new account
  -D, --defaults           print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE expiration date of the new account
  -f, --inactive INACTIVE   password inactivity period of the new account
  -F, --add-subids-for-system
                            add entries to sub[ud]id even when adding a system
                            user
  -g, --gid GROUP          name or ID of the primary group of the new
                            account
  -G, --groups GROUPS      list of supplementary groups of the new
                            account
  -h, --help                display this help message and exit
  -k, --skel SKEL_DIR       use this alternative skeleton directory
  -K, --key KEY=VALUE       override /etc/login.defs defaults
```

02. DHCP, DNS and NTP Services

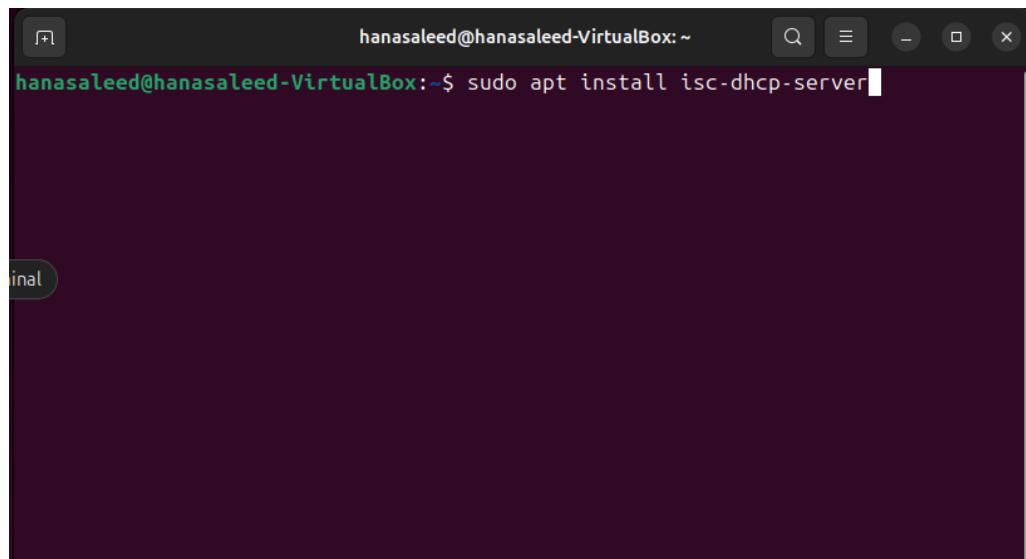
02.1 DHCP (Dynamic Host Configuration Protocol)

DHCP is a network protocol used to automatically assign IP addresses to devices on a network. This does not require a manual configuration. DHCP simplifies the process of managing IP address assignment, particularly in large networks helping to prevent IP address conflicts. First and foremost, to install DHCP the ubuntu has to be updated to patch vulnerabilities in the system, enhance or for better performances.

The DHCP server can be installed by the command **sudo apt install isc-dhcp-server**



A screenshot of a terminal window titled 'hanasaleed@hanasaleed-VirtualBox: ~'. The window shows the command 'sudo apt update' being typed into the terminal. The background of the terminal is dark, and the text is white.



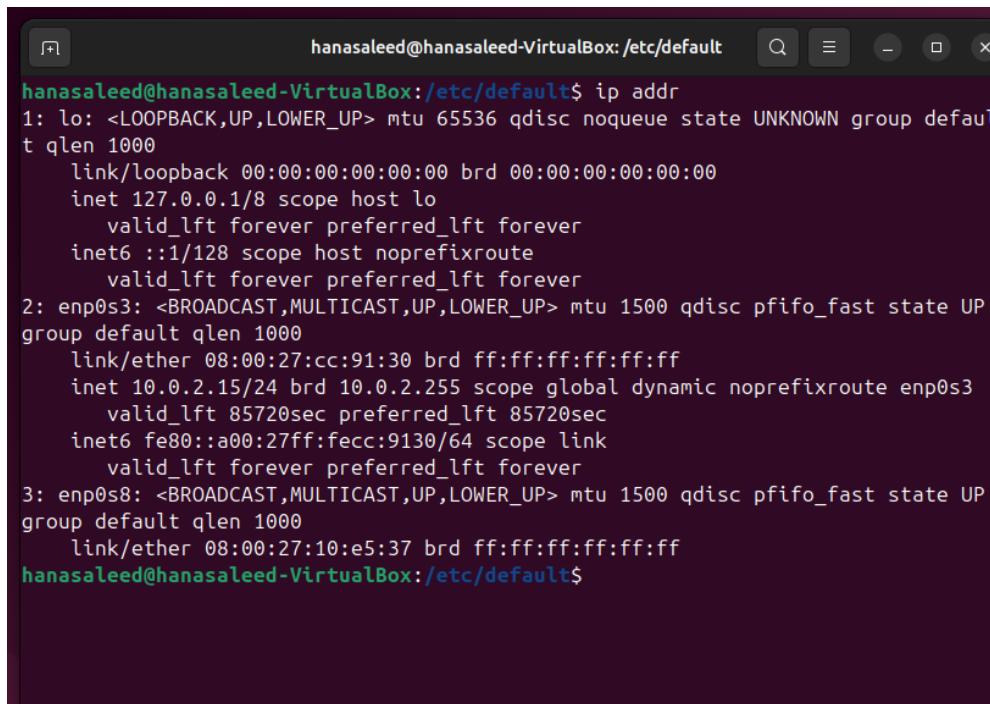
A screenshot of a terminal window titled 'hanasaleed@hanasaleed-VirtualBox: ~'. The window shows the command 'sudo apt install isc-dhcp-server' being typed into the terminal. The background of the terminal is dark, and the text is white.

Then, one must navigate to the directory where the configuration files are located by `cd /etc/dhcp/` and backup the default file in case of any problems by typing the command `sudo cp dhcpcd.conf dhcpcd.conf.backup`. Next, dhcp file should be created using `sudo touch dhcpcd.conf`

The image displays two terminal windows side-by-side. The top terminal window shows the user navigating to the /etc/dhcp directory and creating a backup of the dhcpcd.conf file. The bottom terminal window shows the user creating a new dhcpcd.conf file using sudo touch.

```
hanasaleed@hanasaleed-VirtualBox:~/Desktop$ cd /etc/dhcp/
hanasaleed@hanasaleed-VirtualBox:/etc/dhcp$ sudo cp dhcpcd.conf dhcpcd.conf.backup
hanasaleed@hanasaleed-VirtualBox:~/Desktop$ sudo touch dhcpcd.conf
```

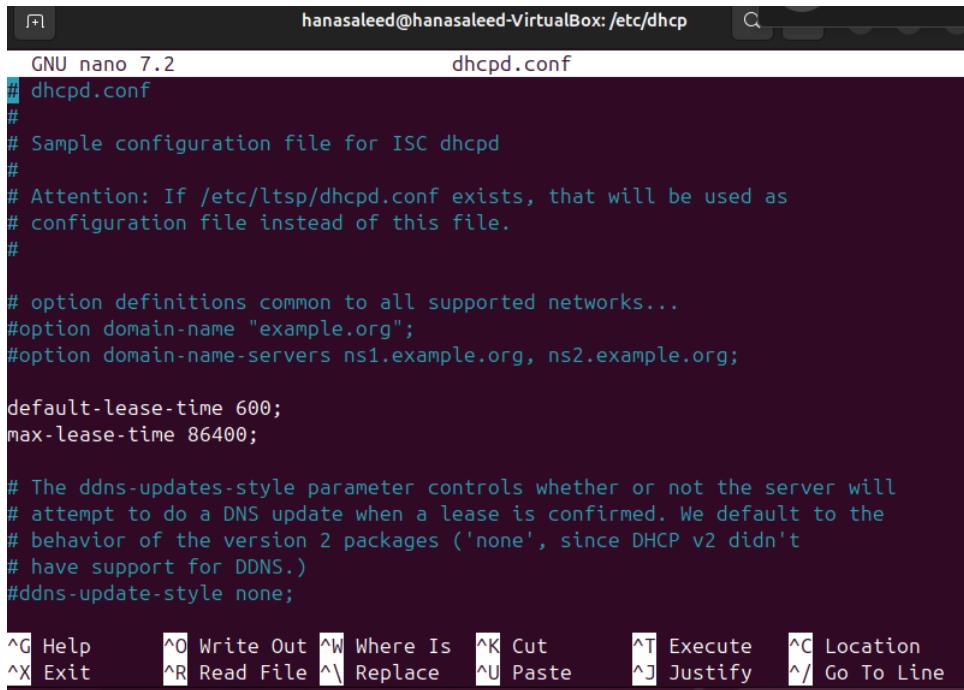
The ip address of the system has to be noted down. In order for that command **ip addr** has to be entered



```
hanasaleed@hanasaleed-VirtualBox:/etc/default$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:cc:91:30 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85720sec preferred_lft 85720sec
    inet6 fe80::a00:27ff:fecc:9130/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:10:e5:37 brd ff:ff:ff:ff:ff:ff
hanasaleed@hanasaleed-VirtualBox:/etc/default$
```

The Ip address here is **10.0.2.15/24** and **enp0s3**

The file has to be opened by the command **sudo nano dhcpcd.conf**. This will open the file **dhcpcd.conf**. The file has to be updated as follows



```
GNU nano 7.2                               dhcpcd.conf
# dhcpcd.conf
#
# Sample configuration file for ISC dhcpcd
#
# Attention: If /etc/ltsp/dhcpcd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 86400;

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
#ddns-update-style none;

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

The screenshot shows a terminal window titled "hanasaleed@hanasaleed-VirtualBox: /etc/dhcp". The file being edited is "dhcpd.conf". The content of the file is as follows:

```
GNU nano 7.2          dhcpd.conf
# range 10.0.29.10 10.0.29.230;
# }
#}
subnet 192.168.1.0 netmask 255.255.255.0 {
#    range 10.0.2.100 10.0.2.200;
#    option routers 10.0.2.1;
#    option subnet-mask 255.255.255.0;
#    option domain-name-servers 8.8.8.8, 8.8.4.4;
#}
option subnet-mask 255.255.255.0;
option broadcast-address 10.0.2.255;
option domain-name "server.local";
authoritative;

subnet 10.0.2.0 netmask 255.255.255.0 {
    range 10.0.2.20 10.0.2.50;
    option routers 10.0.2.1;
    option domain-name-servers 8.8.8.8;
}

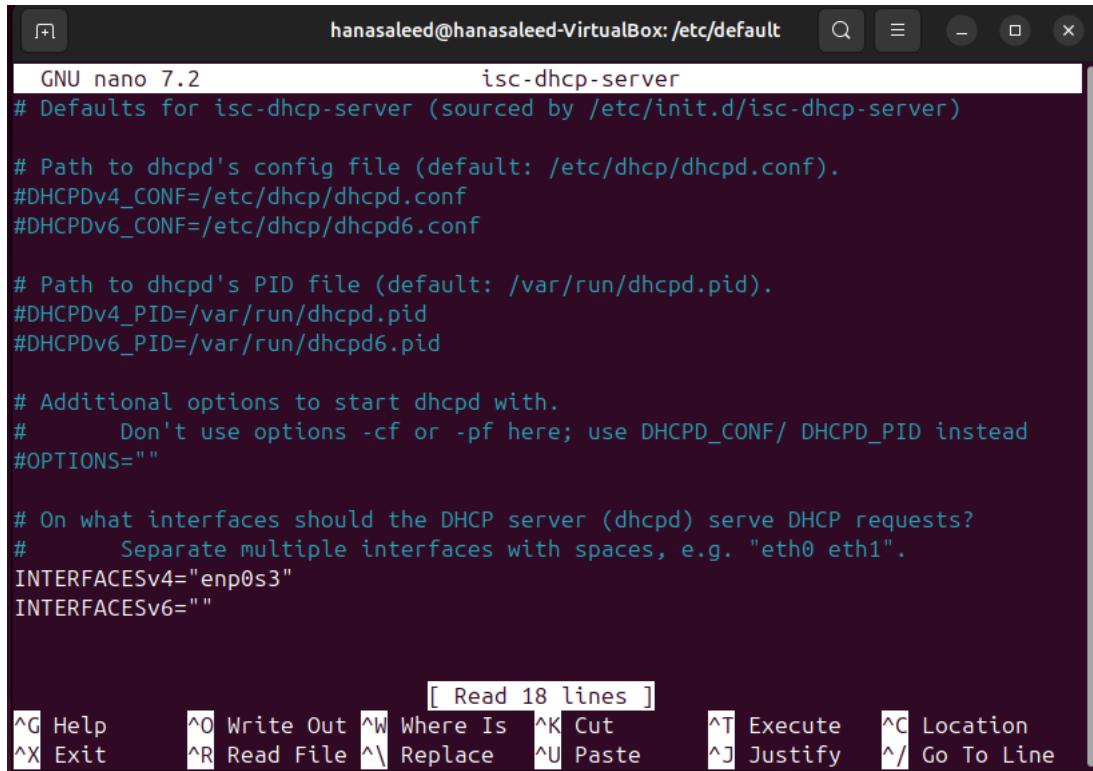
^G Help      ^O Write Out ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^D Read File ^L Read Line ^F Find      ^V Go To Line
```

To assign the DHCP server on a network interface that it will be running on the DHCP file has to be opened by navigating **cd /etc/default/** and **sudo nano isc-dhcp-server**

The screenshot shows a terminal window titled "hanasaleed@hanasaleed-VirtualBox: /etc/dhcp". The command entered is "sudo nano dhcpcd.conf". The output shows the user has permission to edit the file.

```
hanasaleed@hanasaleed-VirtualBox: /etc/dhcp$ sudo nano dhcpcd.conf
Center aleed@hanasaleed-VirtualBox: /etc/dhcp$
```

The empty ipv4 interface field should be filled with respect to the network interface available. Thereby, the network interface can be obtained by command **ip addr** and added it to the file. Here **enp0s3** has been chosen



```
hanasaleed@hanasaleed-VirtualBox: /etc/default
GNU nano 7.2          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/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

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

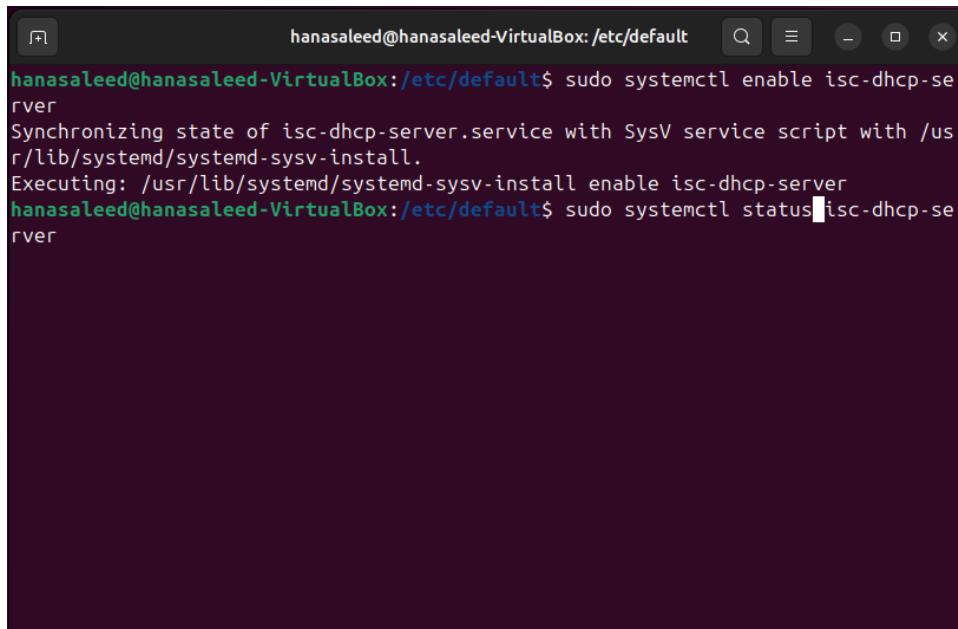
# Additional options to start dhcpcd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpcd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s3"
INTERFACESv6=""
```

[Read 18 lines]

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

Then DHCP server should be enabled by **sudo systemctl enable isc-dhcp-server** and then restart by **sudo systemctl restart isc-dhcp-server**. Finally, the activation of the DHCP command be verified by **sudo systemctl status isc-dhcp-server**.



```
hanasaleed@hanasaleed-VirtualBox: /etc/default$ sudo systemctl enable isc-dhcp-server
Synchronizing state of isc-dhcp-server.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable isc-dhcp-server
hanasaleed@hanasaleed-VirtualBox: /etc/default$ sudo systemctl status isc-dhcp-server
```

```
hanasaleed@hanasaleed-VirtualBox: /etc/default$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/usr/lib/systemd/system/isc-dhcp-server.service; enabled; v>
   Active: active (running) since Sat 2024-10-05 15:57:58 +0530; 11min ago
     Docs: man:dhcpd(8)
 Main PID: 2974 (dhcpd)
    Tasks: 1 (limit: 4498)
   Memory: 4.6M (peak: 4.8M)
      CPU: 320ms
    CGroup: /system.slice/isc-dhcp-server.service
            └─2974 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/d>

Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: PID file: /run/dhcp-server/d>
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Wrote 0 leases to leases fil>
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Wrote 0 leases to leases file.
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Listening on LPF/enp0s3/08:0>
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Listening on LPF/enp0s3/08:00:2>
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Sending on   LPF/enp0s3/08:00:2>
Oct 05 15:57:59 hanasaleed-VirtualBox sh[2974]: Sending on   Socket/fallback/fa>
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Sending on   LPF/enp0s3/08:0>
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Sending on   Socket/fallback>
Oct 05 15:57:59 hanasaleed-VirtualBox dhcpd[2974]: Server starting service.
[lines 1-21/21 (END)]
```

02.2 DNS (Domain Name System)

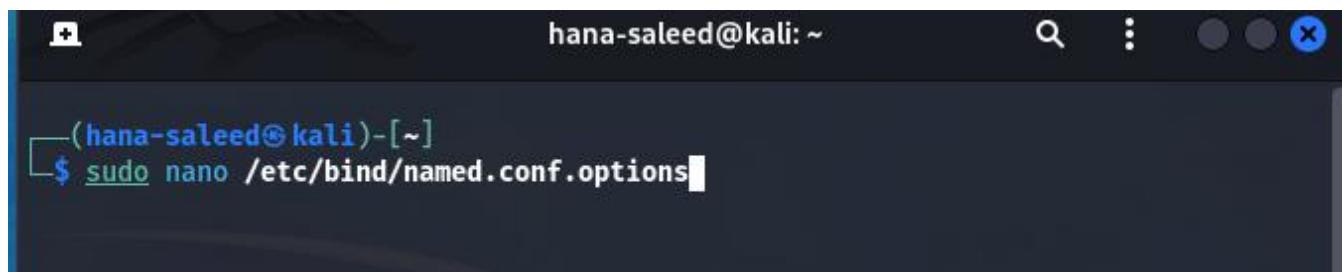
DNS is a naming system that translates human-readable domain names into IP addresses that computer identifies in the network. First, to install DNS server BIND (Berkeley Internet Name Domain) system has to be updated and BIND has to be installed.

```
(hana-saleed㉿kali)-[~]
$ sudo apt update
[sudo] password for hana-saleed:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2199 packages can be upgraded. Run 'apt list --upgradable' to see them.

(hana-saleed㉿kali)-[~]
$
```

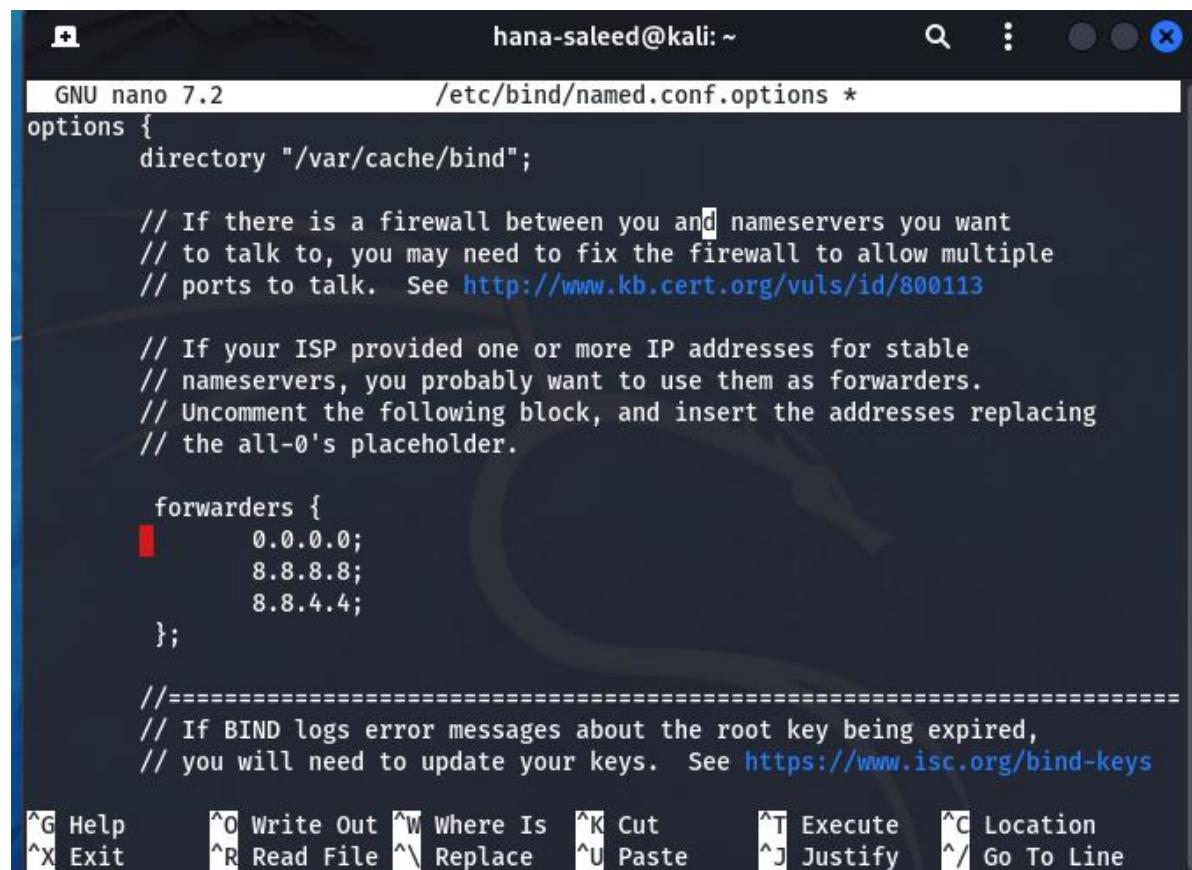
```
(hana-saleed㉿kali)-[~]
$ sudo apt install bind9 bind9utils bind9-doc
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bind9-dnsutils bind9-host bind9-libs bind9-utils liburcu8t64 libuv1-dev
    libuv1t64
Suggested packages:
  bind-doc resolvconf ufw libuv1-doc
The following packages will be REMOVED:
  liburcu8 libuv1
The following NEW packages will be installed:
  bind9 bind9-doc bind9-utils bind9utils liburcu8t64 libuv1t64
The following packages will be upgraded:
  bind9-dnsutils bind9-host bind9-libs libuv1-dev
4 upgraded, 6 newly installed, 2 to remove and 2195 not upgraded.
Need to get 7725 kB of archives.
After this operation, 10.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kali.download/kali kali-rolling/main amd64 bind9-dnsutils amd64 1:9.2
```

The main configuration file has to be edited by **sudo nano /etc/bind/named.conf.options**



```
hanna-saleed@kali: ~
(hanna-saleed@kali)-[~]
$ sudo nano /etc/bind/named.conf.options
```

Thereafter, forwarders should be uncommented and set up a public server like google which has the DNS 8.8.8.8 and 8.8.4.4.



```
GNU nano 7.2          /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 {
        0.0.0.0;
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // 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
}

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

The file should be saved and DNS zones should be edited in the file **named.conf** by the command **sudo nano /etc/bind/named.conf.local**

We should navigate to **/etc/bind/db.example.com** , edit the file as follow and check for syntax error

A terminal window titled "hana-saleed@kali: ~". The session starts with the user navigating to the directory `/etc/bind/db.example.com`. Then, they run the command `sudo named-checkconf`, which outputs the message "OK". Finally, they run `sudo named-checkzone example.com /etc/bind/db.example.com`, which outputs the message "zone example.com/IN: loaded serial 2".

```
(hana-saleed㉿kali)-[~]
$ sudo nano /etc/bind/db.example.com

(hana-saleed㉿kali)-[~]
$ sudo named-checkconf

(hana-saleed㉿kali)-[~]
$ sudo named-checkzone example.com /etc/bind/db.example.com
zone example.com/IN: loaded serial 2
OK

(hana-saleed㉿kali)-[~]
$
```

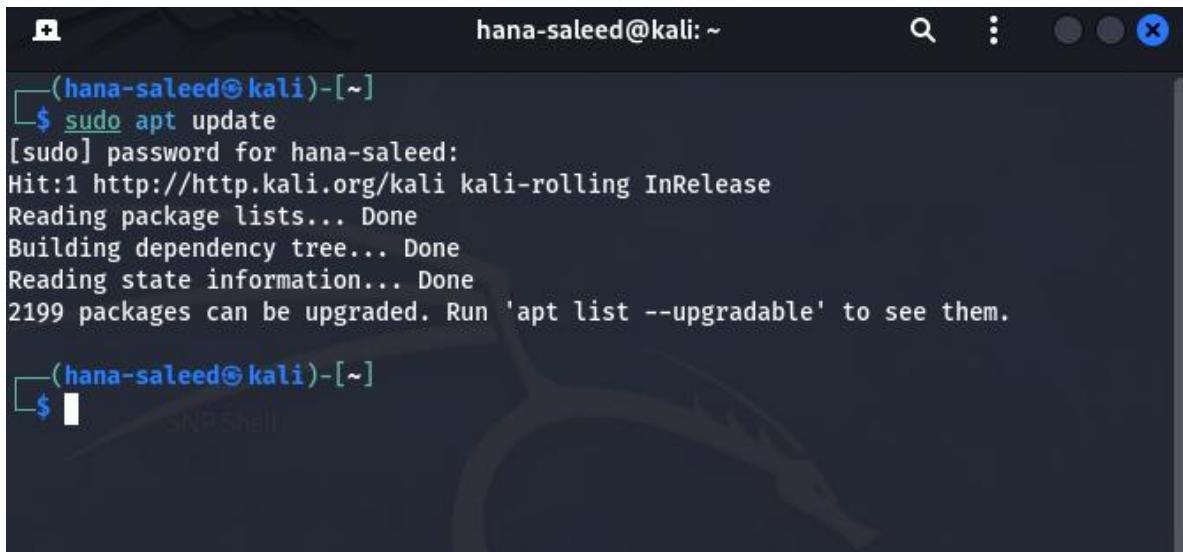
A terminal window titled "hana-saleed@kali: ~". The user is in the directory `/etc/bind/db.example.com` and has opened the file with `sudo nano`. The file contains the following DNS zone configuration:

```
GNU nano 7.2                               /etc/bind/db.example.com *
$TTL    604800
@       IN      SOA     ns1.example.com. admin.example.com. (
                          2           ; Serial
                          604800      ; Refresh
                          86400       ; Retry
                          2419200     ; Expire
                          604800 )     ; Negative Cache TTL
;
@       IN      NS      ns1.example.com.
@       IN      A       10.0.2.15
ns1    IN      A       10.0.2.15
www   IN      A       10.0.2.15
```

02.3 NTP (Network Time Protocol)

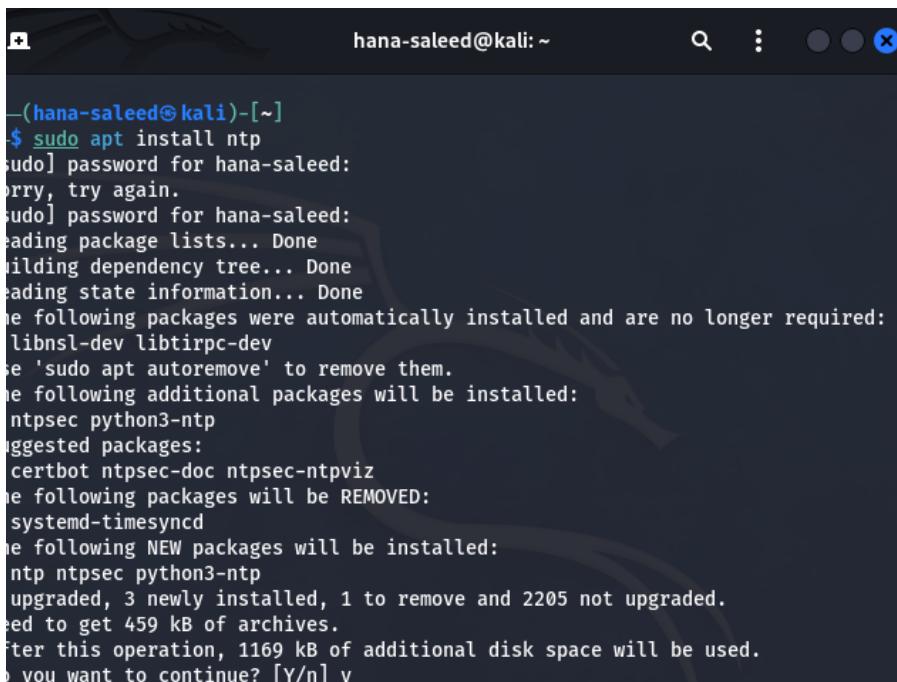
Network Time Protocol (NTP) is an internet protocol used to synchronize with computer clock time sources in a network. NTP helps ensure that all devices on a network maintain accurate and consistent time, which is essential for logging events, scheduling tasks, and coordinating activities across distributed systems.

First update the package and install ntp



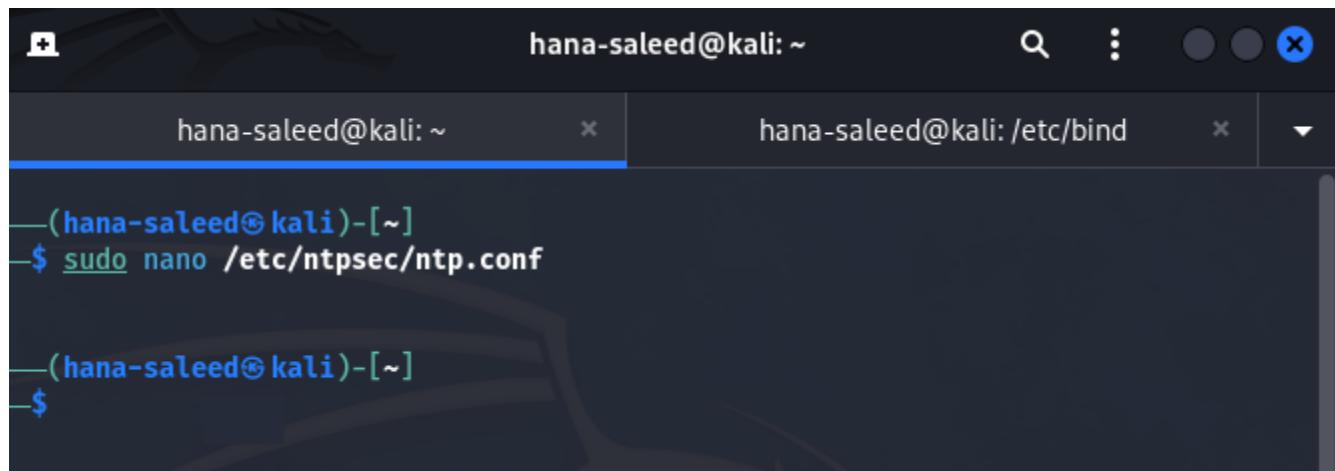
```
(hana-saleed㉿kali)-[~]
$ sudo apt update
[sudo] password for hana-saleed:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2199 packages can be upgraded. Run 'apt list --upgradable' to see them.

(hana-saleed㉿kali)-[~]
$
```



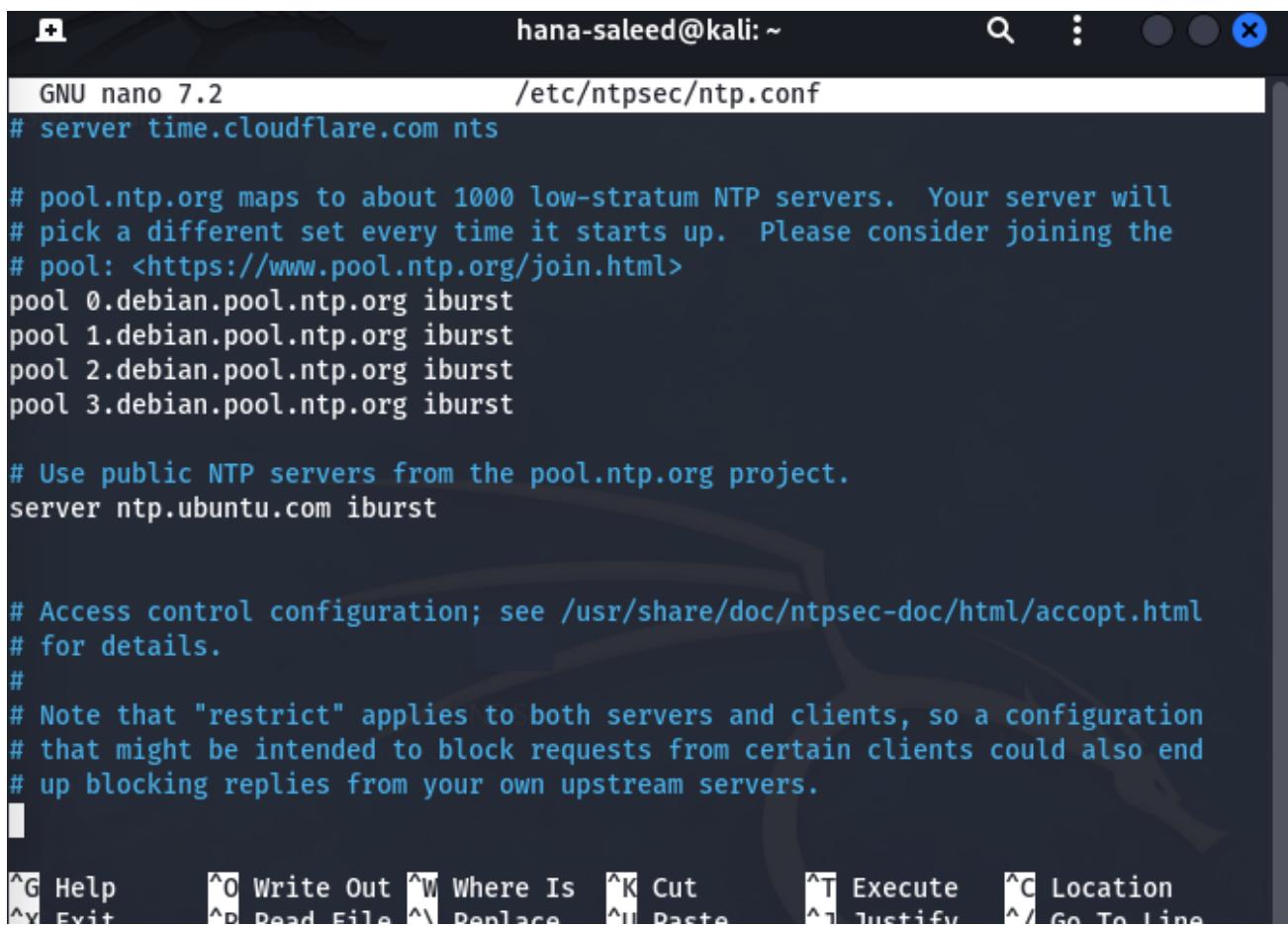
```
(hana-saleed㉿kali)-[~]
$ sudo apt install ntp
[sudo] password for hana-saleed:
Sorry, try again.
[sudo] password for hana-saleed:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
ntpsec python3-ntp
Suggested packages:
certbot ntpsec-doc ntpsec-ntpviz
The following packages will be REMOVED:
systemd-timesyncd
The following NEW packages will be installed:
ntp ntpsec python3-ntp
Upgraded, 3 newly installed, 1 to remove and 2205 not upgraded.
Need to get 459 kB of archives.
After this operation, 1169 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

The ntp file has to be opened using the nano command it has to be updated as follows



```
hanna-saleed@kali: ~
hanna-saleed@kali: ~
(hanna-saleed@kali)-[~]
$ sudo nano /etc/ntpsec/ntp.conf

(hanna-saleed@kali)-[~]
$
```



```
hanna-saleed@kali: ~
GNU nano 7.2          /etc/ntpsec/ntp.conf
# server time.cloudflare.com nts

# pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will
# pick a different set every time it starts up. Please consider joining the
# pool: <https://www.pool.ntp.org/join.html>
pool 0.debian.pool.ntp.org iburst
pool 1.debian.pool.ntp.org iburst
pool 2.debian.pool.ntp.org iburst
pool 3.debian.pool.ntp.org iburst

# Use public NTP servers from the pool.ntp.org project.
server ntp.ubuntu.com iburst

# Access control configuration; see /usr/share/doc/ntpsec-doc/html/accept.html
# for details.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

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

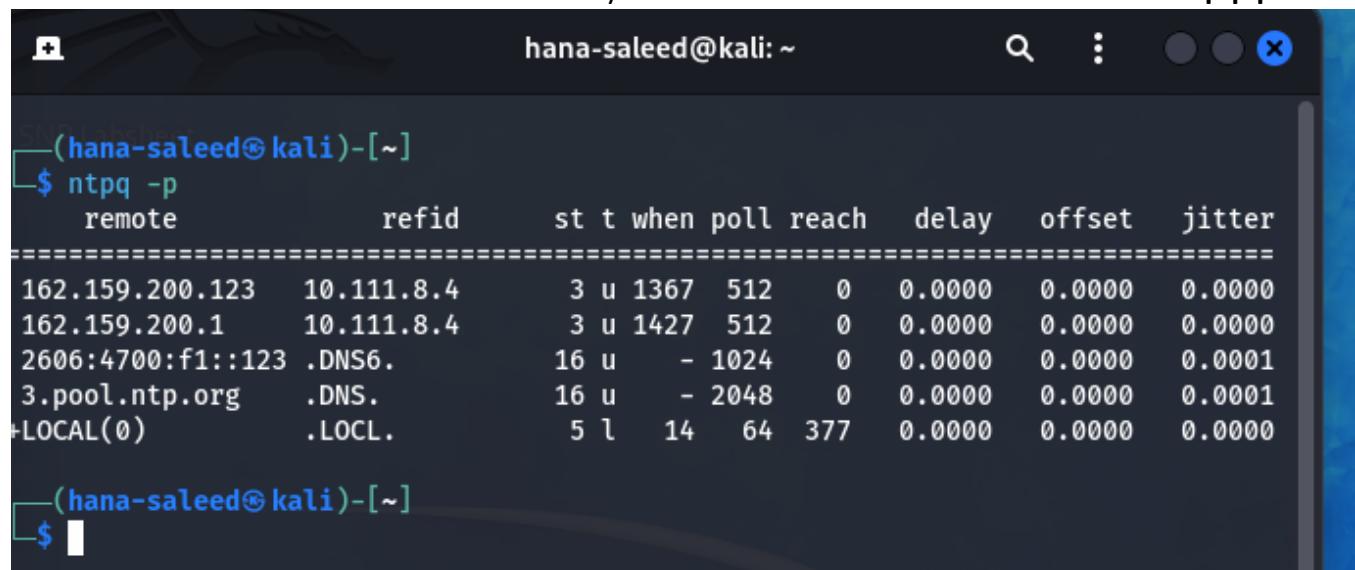
Then the service should be started and the enabled and then we can check the status

```
→ sudo systemctl status ntp
SNP Laptopsheet

● ntpsec.service - Network Time Service
  Loaded: loaded (/usr/lib/systemd/system/ntpsec.service; enabled; preset: disabled)
  Active: active (running) since Sun 2024-10-06 18:32:32 +0530; 2h 43min ago
    Invocation: b6d3d4dca0214d4fbb6fbabbedb5dbe8
      Docs: man:ntpd(8)
   Main PID: 4371 (ntpd)
     Tasks: 1 (limit: 3529)
    Memory: 12M (peak: 12.7M)
       CPU: 3.450s
      CGroup: /system.slice/ntpsec.service
              └─4371 /usr/sbin/ntpd -p /run/ntpd.pid -c /etc/ntpsec/ntp.conf -g -N

Oct 06 20:15:22 kali ntpd[4371]: DNS: dns_probe: 3.pool.ntp.org, cast_flags:1, fl>
Oct 06 20:15:22 kali ntpd[4371]: DNS: dns_check: processing 3.pool.ntp.org, 1, 20>
Oct 06 20:15:22 kali ntpd[4371]: DNS: Server skipping: 162.159.200.1
Oct 06 20:15:22 kali ntpd[4371]: DNS: Server skipping: 162.159.200.123
Oct 06 20:15:22 kali ntpd[4371]: DNS: dns_take_status: 3.pool.ntp.org=>good, 11
Oct 06 20:49:31 kali ntpd[4371]: DNS: dns_probe: 3.pool.ntp.org, cast_flags:1, fl>
Oct 06 20:49:31 kali ntpd[4371]: DNS: dns_check: processing 3.pool.ntp.org, 1, 20>
Oct 06 20:49:31 kali ntpd[4371]: DNS: Server skipping: 162.159.200.123
Oct 06 20:49:31 kali ntpd[4371]: DNS: Server skipping: 162.159.200.1
Oct 06 20:49:31 kali ntpd[4371]: DNS: dns_take_status: 3.pool.ntp.org=>good, 11
lines 1-22/22 (END)
```

To check the status and the servers we have synchronized with we can use the command `ntpq -p`



The screenshot shows a terminal window with the title bar "hana-saleed@kali: ~". The command `ntpq -p` is run, and the output is displayed. The output shows the synchronization status of various NTP servers:

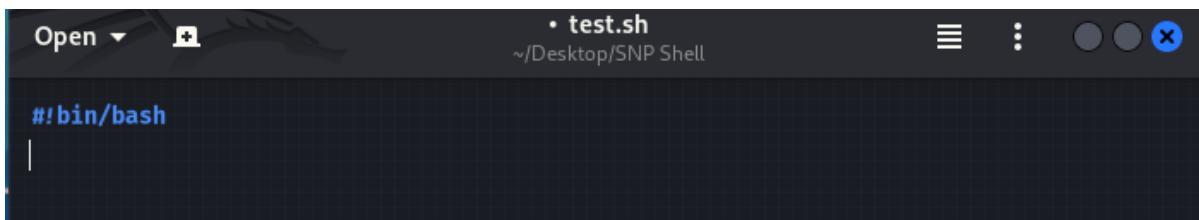
remote	refid	st	t	when	poll	reach	delay	offset	jitter
162.159.200.123	10.111.8.4	3	u	1367	512	0	0.0000	0.0000	0.0000
162.159.200.1	10.111.8.4	3	u	1427	512	0	0.0000	0.0000	0.0000
2606:4700:f1::123	.DNS6.	16	u	-	1024	0	0.0000	0.0000	0.0001
3.pool.ntp.org	.DNS.	16	u	-	2048	0	0.0000	0.0000	0.0001
+LOCAL(0)	.LOCL.	5	l	14	64	377	0.0000	0.0000	0.0000

03. Shell Scripting and Security

03.1 Shell Scripting

i. Basic Shell scripting syntax

Every shell script begins with a special character sequence **#! (shebang)**. This tells the system which interpreter should be used to run the script. That follows a specific path to the Bash Interpreter **/bin/bash**. Without shebang, the script will run using the default shell of the user executing it which may lead to unexpected behavior.



```
#!/bin/bash
```

Comments in shell scripts start with the **#** symbol. Anything after the **#** in the line is ignored by the shell interpreter. Multi-line comments are created by putting **#** at the start of each line



```
#!/bin/bash
#This is a comment
#This is
#a multi-line
#comment|
```

Variables in shell scripts are declared by assigning a value to a name without spaces around the equal sign. They are also case sensitive and every variable is treated as a string. In order to reference the variable a prefix **\$** sign is used for the variable name. Without the **\$** sign the variable name is treated as a literal text. Moreover, **echo** can be used to print output to the console.

The screenshot shows a terminal window titled "test.sh" located at "~/Desktop/SNP Shell". The script content is:

```
#!/bin/bash
Name="Hana Saleed"
AGE="20"
echo "Hello, $Name"
```

The screenshot shows a terminal window titled "hanna-saleed@kali: ~/Desktop/SNP Shell". The command \$ bash test.sh is run, followed by the output "Hello, Hana Saleed".

```
(hanna-saleed@kali)-[~/Desktop/SNP Shell]
$ bash test.sh
Hello, Hana Saleed

(hanna-saleed@kali)-[~/Desktop/SNP Shell]
```

ii. Control flow statements

a. if command

The screenshot shows a terminal window titled "test.sh" located at "~/Desktop/SNP Shell". The script content is:

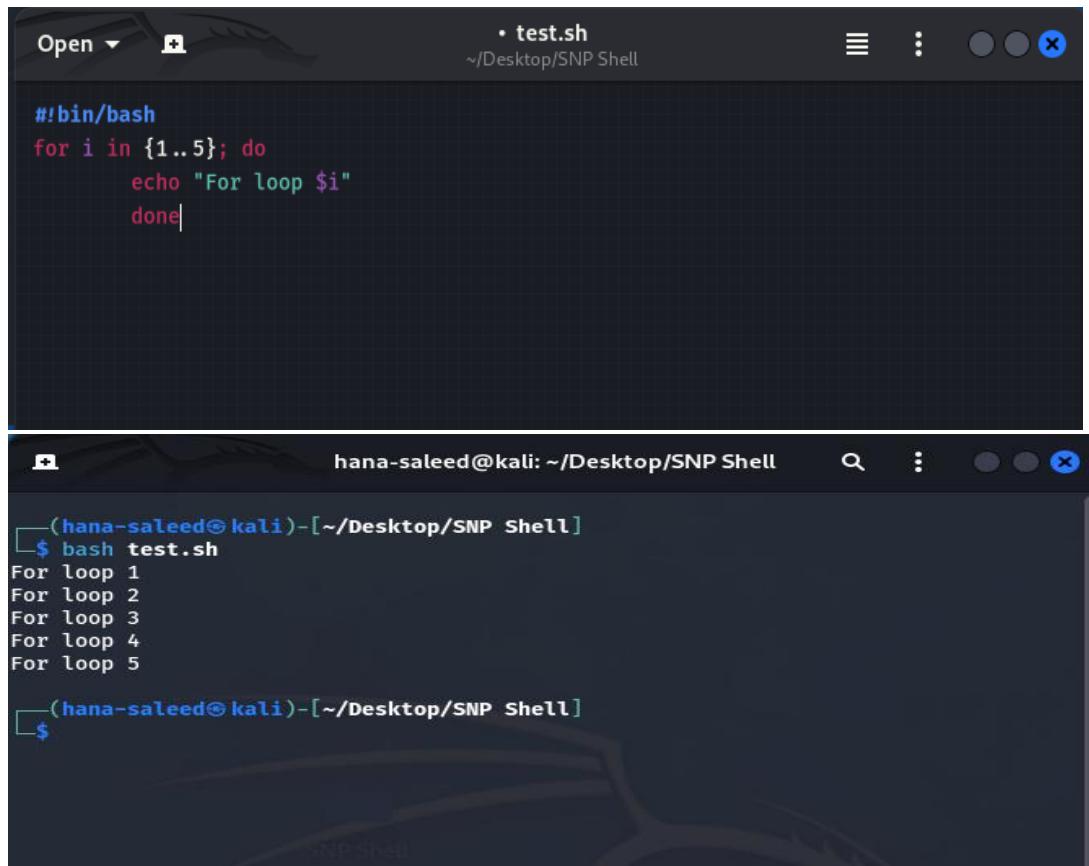
```
#!/bin/bash
AGE="20"
if [ $AGE -gt 18 ]; then
    echo "You can vote"
else
    echo "You cannot vote"
fi
```

The screenshot shows a terminal window titled "hanna-saleed@kali: ~/Desktop/SNP Shell". The command \$ bash test.sh is run, followed by the output "You can vote".

```
(hanna-saleed@kali)-[~/Desktop/SNP Shell]
$ bash test.sh
You can vote

(hanna-saleed@kali)-[~/Desktop/SNP Shell]
```

b. For command



The screenshot shows a terminal window with a dark theme. At the top, it says "test.sh" and "~/Desktop/SNP Shell". Below that is the script content:

```
#!/bin/bash
for i in {1..5}; do
    echo "For loop $i"
done
```

At the bottom, the terminal prompt shows the output of the script:

```
[hana-saleed@kali: ~/Desktop/SNP Shell]
$ bash test.sh
For loop 1
For loop 2
For loop 3
For loop 4
For loop 5
[hana-saleed@kali: ~/Desktop/SNP Shell]
$
```

C. While command



The screenshot shows a terminal window with a dark theme. At the top, it says "test.sh" and "~/Desktop/SNP Shell". Below that is the script content:

```
#!/bin/bash
count=1

while [ $count -le 5 ]; do
    echo "While loop: $count"
    ((count++))
done
```

```
(hana-saleed㉿kali)-[~/Desktop/SNP Shell]
$ bash test.sh
While loop: 1
While loop: 2
While loop: 3
While loop: 4
While loop: 5

(hana-saleed㉿kali)-[~/Desktop/SNP Shell]
$
```

d.Functions

```
hanna@kali: ~/Desktop/SNP Shell
(hana-saleed㉿kali)-[~/Desktop/SNP Shell]
$ bash test.sh
Hello, World!

(hana-saleed㉿kali)-[~/Desktop/SNP Shell]
$
```

```
Open ▾ test.sh
~/Desktop/SNP Shell

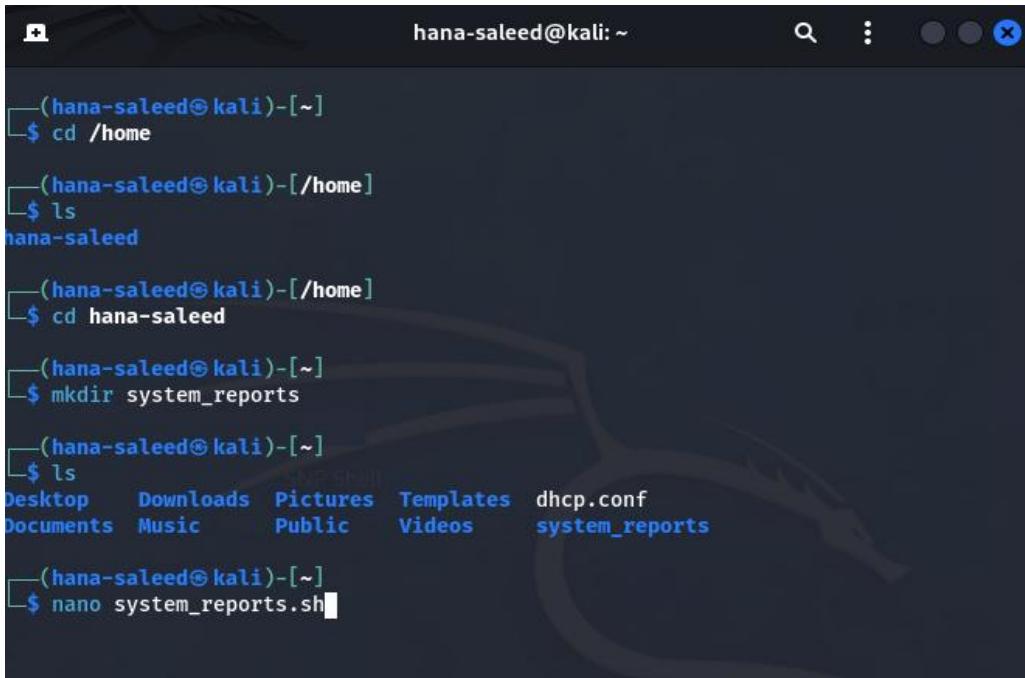
#!/bin/bash

greet() {
    echo "Hello, $1!"
}

greet "World"
```

iii. Automating Daily System Reports

Automating daily system reports involves creating a script which collects key system information such as date, uptime, memory usage and disk storage. The script automatically runs on scheduled at specific intervals using cron jobs. First, we have to navigate to the destination directory given **/home/user/system_reports** using the command **cd**. A directory has to be created named **system_reports** by the command **mkdir**. Thereafter, a shell script has to be created as **system_reports.sh** by using the command **nano** text editor.



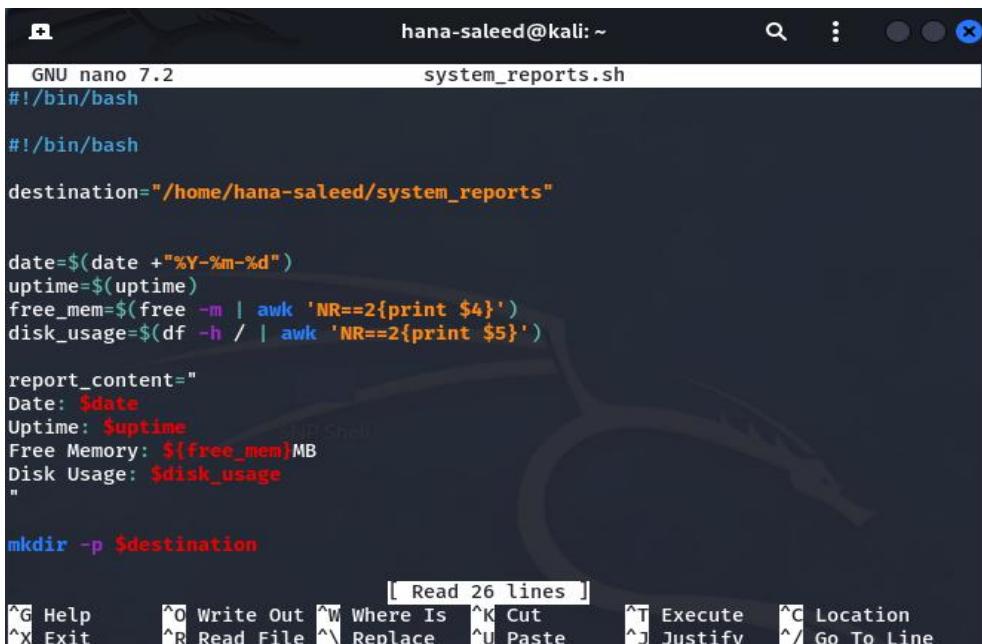
```
(hana-saleed㉿kali)-[~]
$ cd /home
(hana-saleed㉿kali)-[/home]
$ ls
hana-saleed

(hana-saleed㉿kali)-[/home]
$ cd hana-saleed

(hana-saleed㉿kali)-[~]
$ mkdir system_reports
(hana-saleed㉿kali)-[~]
$ ls
Desktop  Downloads  Pictures  Templates  dhcp.conf
Documents  Music      Public    Videos      system_reports

(hana-saleed㉿kali)-[~]
$ nano system_reports.sh
```

The command will create open a file and then the file should be edited as the following script



```
GNU nano 7.2
hanna-saleed@kali: ~
system_reports.sh

#!/bin/bash
#!/bin/bash

destination="/home/hana-saleed/system_reports"

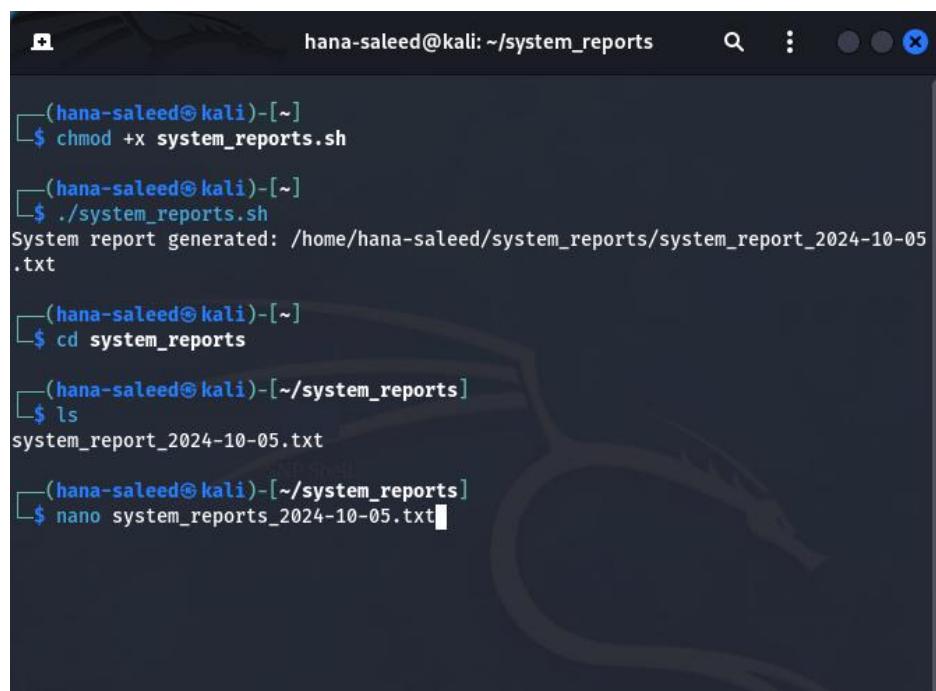
date=$(date +"%Y-%m-%d")
uptime=$(uptime)
free_mem=$(free -m | awk 'NR==2{print $4}')
disk_usage=$(df -h / | awk 'NR==2{print $5}')

report_content="
Date: $date
Uptime: $uptime
Free Memory: ${free_mem}MB
Disk Usage: $disk_usage
"

mkdir -p $destination
```

- **#!/bin/bash** : This tells the system to use the Bash shell to interpret the script
- **destination="/home/hana-saleed/system_reports"** : This sets the path where the report will be saved
- **date=\$(date +"Y-%m-%d")** : The **date** gets the current date in YYYY-MM-DD format. The \$ is a syntax used to execute the command and it stores its output in the variable **date**.
- **uptime=\$(uptime)** : This command captures how long the system has been running since the last boot
- **free_mem=\$(free -m | awk 'NR==2{print \$4}')** : The command **free -m** shows the memory usage in megabytes. **N==2** prints the fourth line column from the precessed output which shows the free memory available. This value will be stored in the variable **free_mem**
- **disk_usage=\$(df -h / | awk 'NR==2{print \$5}')** : **df -h** command outputs disk space usage in human readable form.
- **mkdir -p \$destination** : Creates the directory if it does not already exist. The **-p** ensures no error is shown if the directory exists
- **report_file="\$destination/system_report_\$date.txt"** : Creates a filename of the report with the date
- **echo "\$report_content" > \$report_file** : This will output the contents in the report to the file

After the script is edited it has to be saved and exited. Then, permission has to be enabled to make it executable and the system has to run by the command **./system_reports.sh**. This will display the directory of report generated. The file with date has to be opened by the command **nano**.

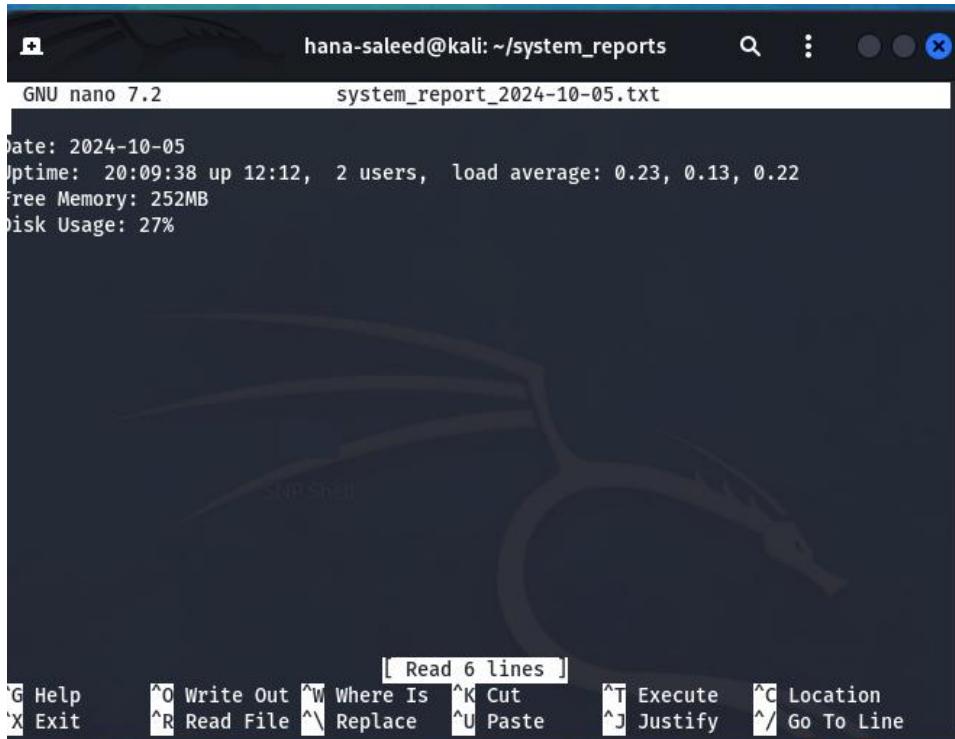


```

hanna@kali: ~/system_reports
(hanna@kali)-[~]
$ chmod +x system_reports.sh
(hanna@kali)-[~]
$ ./system_reports.sh
System report generated: /home/hanna-saleed/system_reports/system_report_2024-10-05.txt
(hanna@kali)-[~]
$ cd system_reports
(hanna@kali)-[~/system_reports]
$ ls
system_report_2024-10-05.txt
(hanna@kali)-[~/system_reports]
$ nano system_report_2024-10-05.txt

```

After the execution of the previous command, it will output the report as



The screenshot shows a terminal window titled "hana-saleed@kali: ~/system_reports". The file being edited is "system_report_2024-10-05.txt". The content of the file is as follows:

```
Date: 2024-10-05
Uptime: 20:09:38 up 12:12, 2 users, load average: 0.23, 0.13, 0.22
Free Memory: 252MB
Disk Usage: 27%
```

At the bottom of the terminal window, there is a menu bar with various options: Help, Write Out, Where Is, Cut, Execute, Location, Exit, Read File, Replace, Paste, Justify, and Go To Line. A message "[Read 6 lines]" is displayed above the menu bar.

To schedule it to run daily, cron can be used. It can be opened with the command

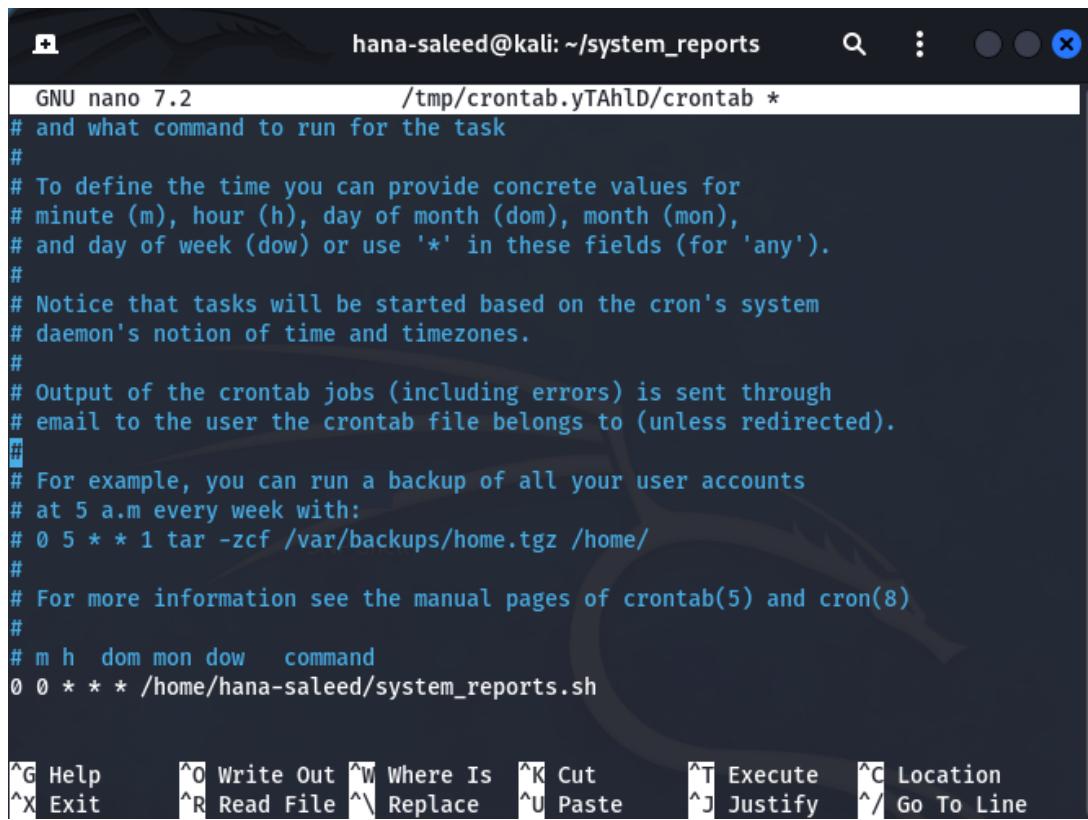


The screenshot shows a terminal window titled "hana-saleed@kali: ~/system_reports". The user runs the command "\$ crontab -e". The terminal displays the following text:

```
no crontab for hana-saleed - using an empty one
crontab: installing new crontab
```

At the bottom of the terminal window, there is a prompt "\$" indicating where the user can enter their next command.

The above command will open a file and it has to be edited in a way that it execute the script every midnight, generating a report with system details in the specific directory.



A screenshot of a terminal window titled "hana-saleed@kali: ~/system_reports". The window shows the "GNU nano 7.2" text editor with a crontab file open. The file contains the following content:

```
GNU nano 7.2          /tmp/crontab.yTAhLD/crontab *
# 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 0 * * * /home/hana-saleed/system_reports.sh
```

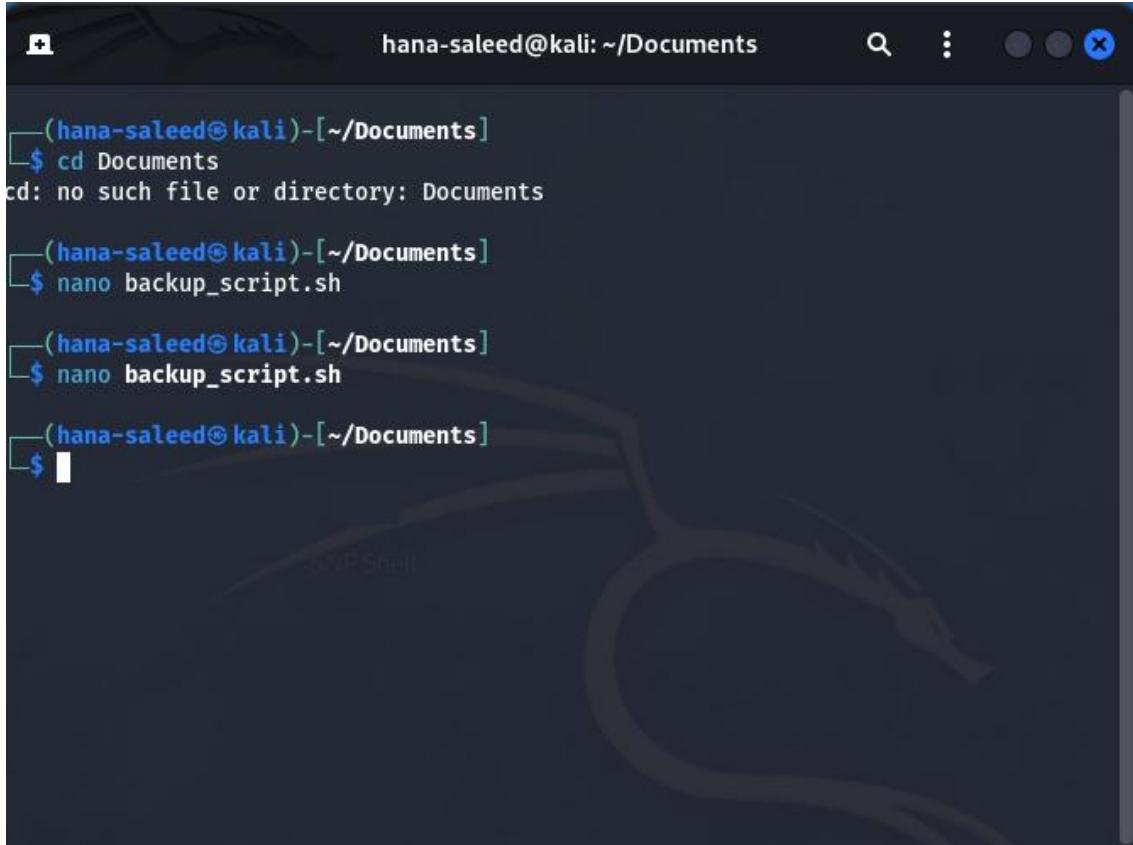
At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for navigating and editing the file.

The line **0 0 * * * /homr/hana-saleed/system_reports.sh** has to be added to the file

iii. Automating Backup Of Critical Files

This shell scripting is to ensure that important files are regularly copied and stored safely. In this case it will be stored in the directory **/home/user/documents**. The script can be scheduled to run periodically by using cron jobs. This helps with data loss.

The directory has to be changed using the **cd** command and open a new script under the name **backup_scrip.sh** by the command **nano**



A screenshot of a terminal window titled "hana-saleed@kali: ~/Documents". The terminal shows the following sequence of commands:

```
(hana-saleed㉿kali)-[~/Documents]
$ cd Documents
cd: no such file or directory: Documents

(hana-saleed㉿kali)-[~/Documents]
$ nano backup_script.sh

(hana-saleed㉿kali)-[~/Documents]
$ nano backup_script.sh

(hana-saleed㉿kali)-[~/Documents]
$
```

This will open the new script file and it has to be edited as such:

- **DATE=\$(date '+%Y-%m-%d')** : This command will get the date in YYYY-MM-DD format
- **BACKUP_DIR="/home/user/documents"**: Variable assigned to get the directory that needs to be backed up.
- **DEST_DIR="/home/user/backups"**: This command is to assign the destination directory where the backp up will be save
- **BACKUP_FILE="backup_\$DATE.tar.gz"**: Creates the file name of the backup file with the current date
- **mkdir -p \$DEST_DIR**: Creates the destination directory
- **tar -czvf \$DEST_DIR/\$BACKUP_FILE \$BACKUP_DIR**: Command that creates the backup

A screenshot of a terminal window titled "hanna-saleed@kali: ~/Documents". The window shows a file named "backup_script.sh" being edited with the GNU nano 7.2 text editor. The script content is as follows:

```
GNU nano 7.2          backup_script.sh
#!/bin/bash

source_dir="/home/hanna-saleed/Documents"
destination_dir="/home/hanna-saleed/backup/Documents"

mkdir -p $destination_dir

backup_file="$destination_dir/Documents_$(date +%Y-%m-%d).tar.gz"

tar -czf $backup_file $source_dir
echo "Backup Completed: $backup_file"
```

The file has to be given permission to make it executable and run using `./backup_script.sh` command. Then the backup path should be followed

A screenshot of a terminal window titled "hanna-saleed@kali: ~/backup/Documents". The session shows the following commands and their output:

```
(hanna-saleed㉿kali)-[~]
$ ls
Desktop  Downloads  Pictures  Templates  backup    system_reports
Documents  Music      Public     Videos      dhcp.conf  system_reports.sh

(hanna-saleed㉿kali)-[~]
$ cd backup

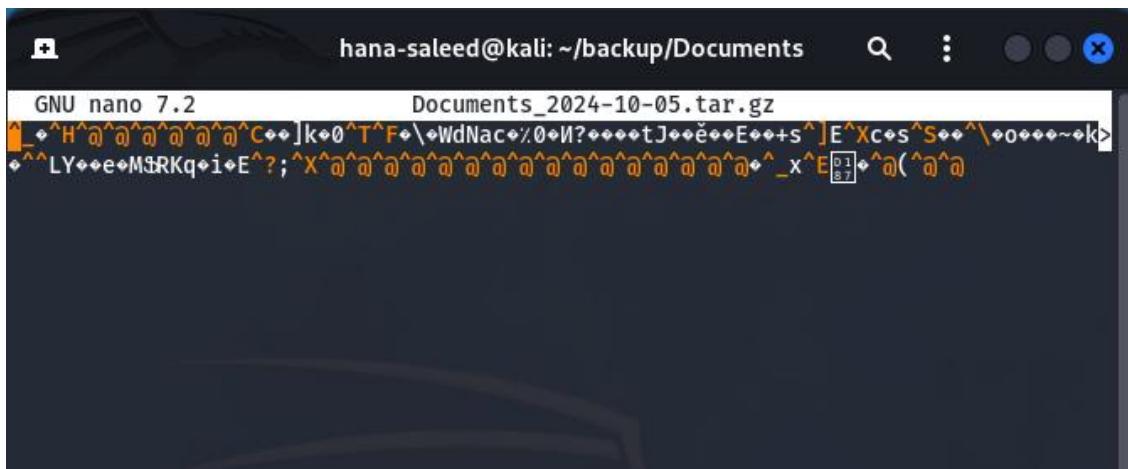
(hanna-saleed㉿kali)-[~/backup]
$ ls
Documents

(hanna-saleed㉿kali)-[~/backup]
$ cd Documents

(hanna-saleed㉿kali)-[~/backup/Documents]
$ ls
Documents_2024-10-05.tar.gz

(hanna-saleed㉿kali)-[~/backup/Documents]
$ nano Documents_2024-10-05.tar.gz
```

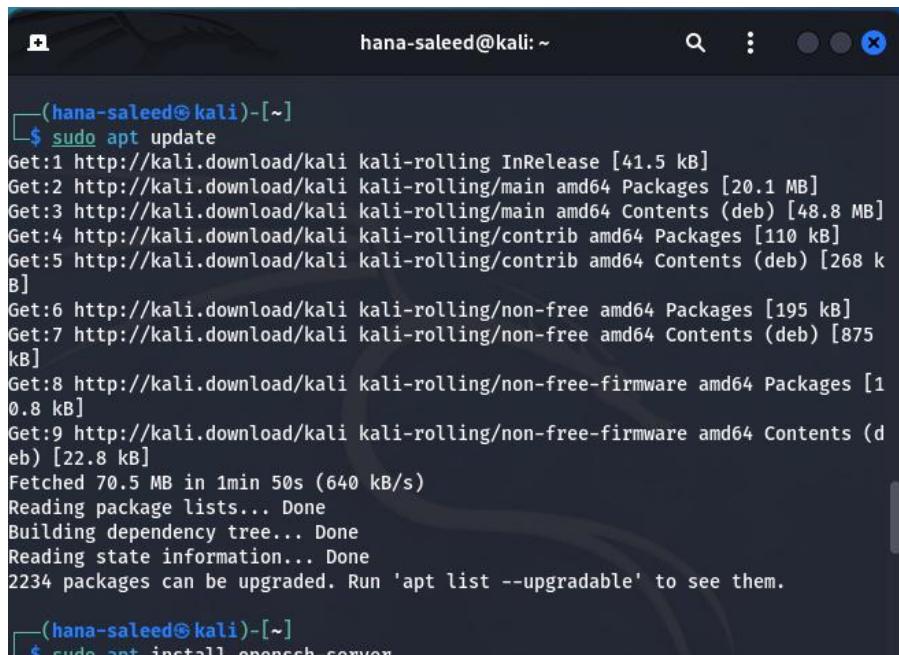
The file can be opened with the date can be opened with the command **nano** and the view of the backup file be as



A screenshot of a terminal window titled "hana-saleed@kali: ~/backup/Documents". The window shows the command "nano 7.2" followed by the file name "Documents_2024-10-05.tar.gz". The terminal displays a large amount of compressed binary data, appearing as a series of orange and yellow characters. In the bottom right corner of the terminal window, there is a status bar with the text "01 87".

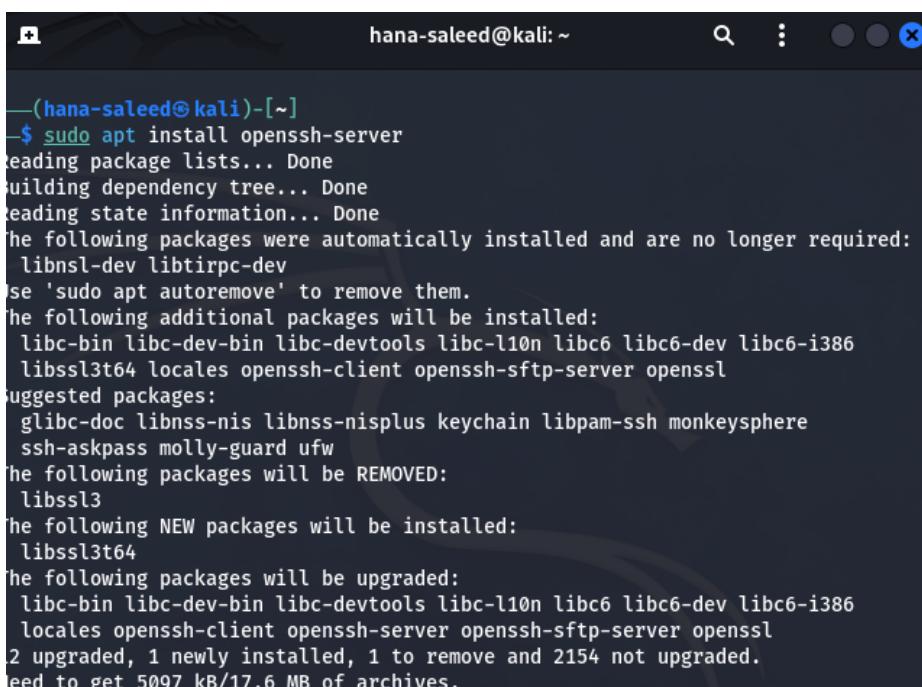
03.2 SSH (Secure Shell)

SSH is a cryptographic network protocol that allows you to securely communicate with a remote system. It provides a secure way to access and manage remote devices over an unsecured network. Initially, kali has to be updated by **sudo apt update** and OpenSSH server package has to be installed by **sudo apt install openssh-server**



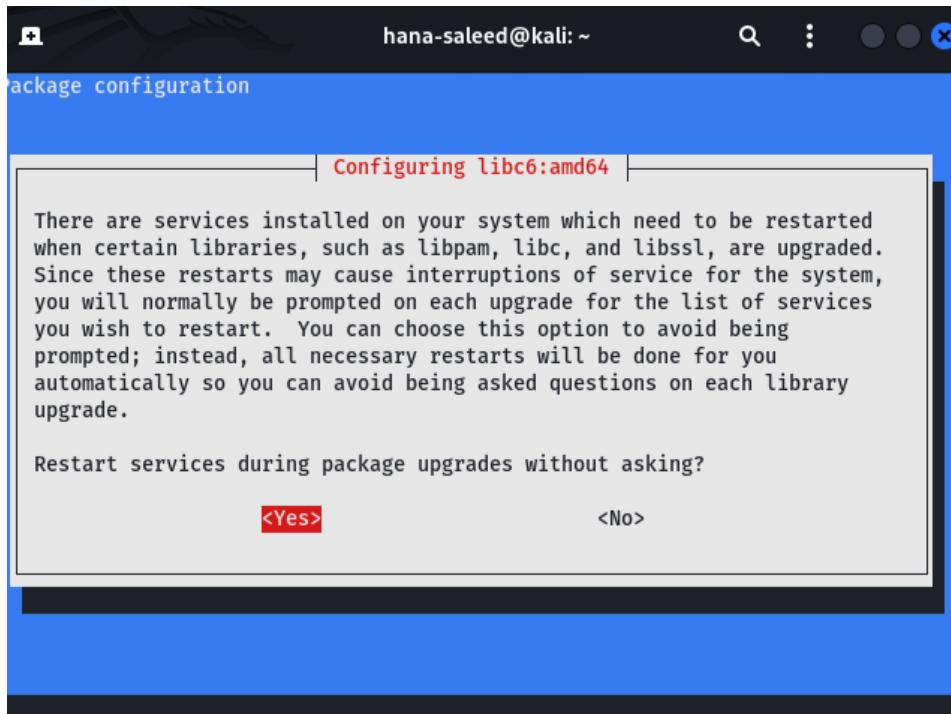
```
(hana-saleed㉿kali)-[~]
$ sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.1 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [48.8 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [110 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [268 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [195 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [875 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [1 0.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [22.8 kB]
Fetched 70.5 MB in 1min 50s (640 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2234 packages can be upgraded. Run 'apt list --upgradable' to see them.

(hana-saleed㉿kali)-[~]
```



```
(hana-saleed㉿kali)-[~]
$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dev libc6-i386
  libssl3t64 locales openssh-client openssh-sftp-server openssl
Suggested packages:
  glibc-doc libnss-nis libnss-nisplus keychain libpam-ssh monkeysphere
  ssh-askpass molly-guard ufw
The following packages will be REMOVED:
  libssl3
The following NEW packages will be installed:
  libssl3t64
The following packages will be upgraded:
  libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dev libc6-i386
  locales openssh-client openssh-server openssh-sftp-server openssl
2 upgraded, 1 newly installed, 1 to remove and 2154 not upgraded.
Need to get 5097 kB/17.6 MB of archives.
```

After the installation it requests to restart and confirm by giving yes.



Navigate to `/etc/ssh/sshd_config` and add Port 2220

A screenshot of a terminal window showing the contents of the "/etc/ssh/sshd_config" file in a nano editor. The file includes comments about the configuration and the compiled PATH. It shows the "Include" directive, the original "Port 22" line, and the new "Port 2220" line that has been added. The bottom of the screen shows nano editor command keys.

```
GNU nano 7.2          /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

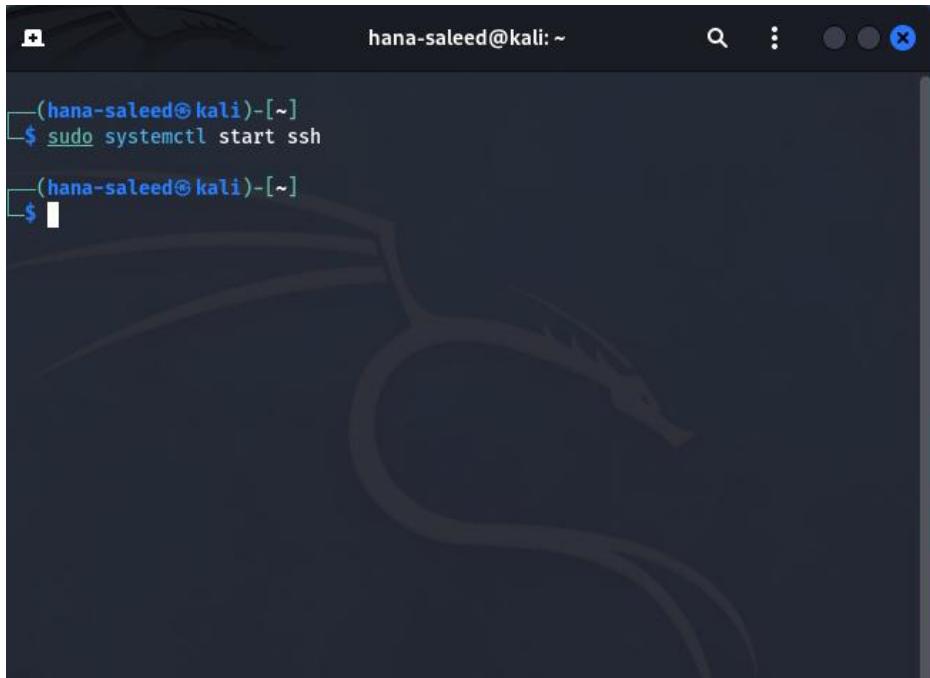
Port 2220
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

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

[Read 123 lines]

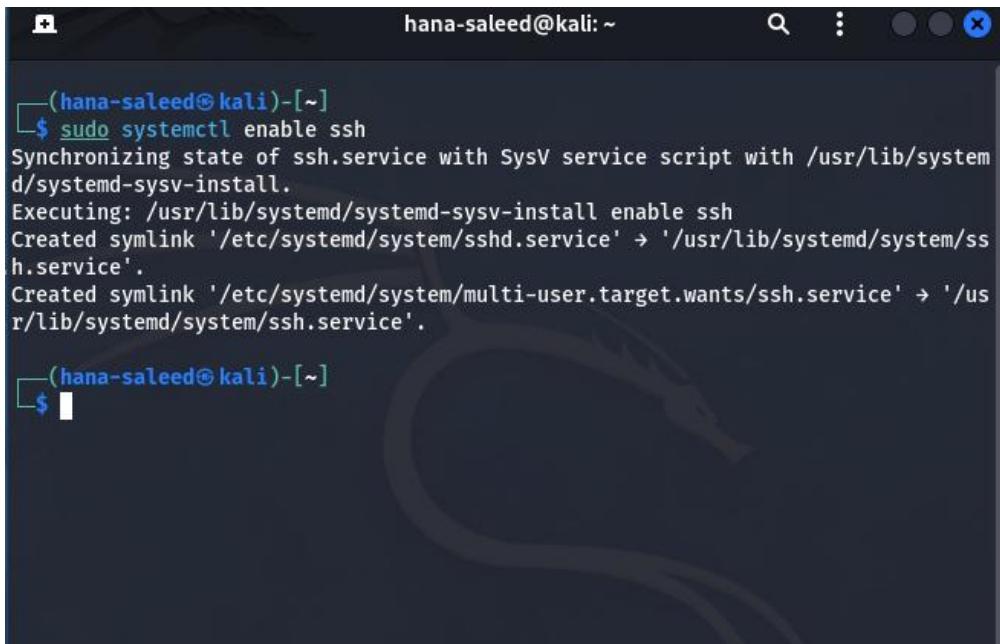
^G Help **^O Write Out** **^W Where Is** **^K Cut** **^T Execute** **^C Location**
^X Exit **^R Read File** **^\\ Replace** **^U Paste** **^J Justify** **^/ Go To Line**

Starting SSH launches the service, allowing connections to the service. If SSH is not started, no remote user can access the machine securely.



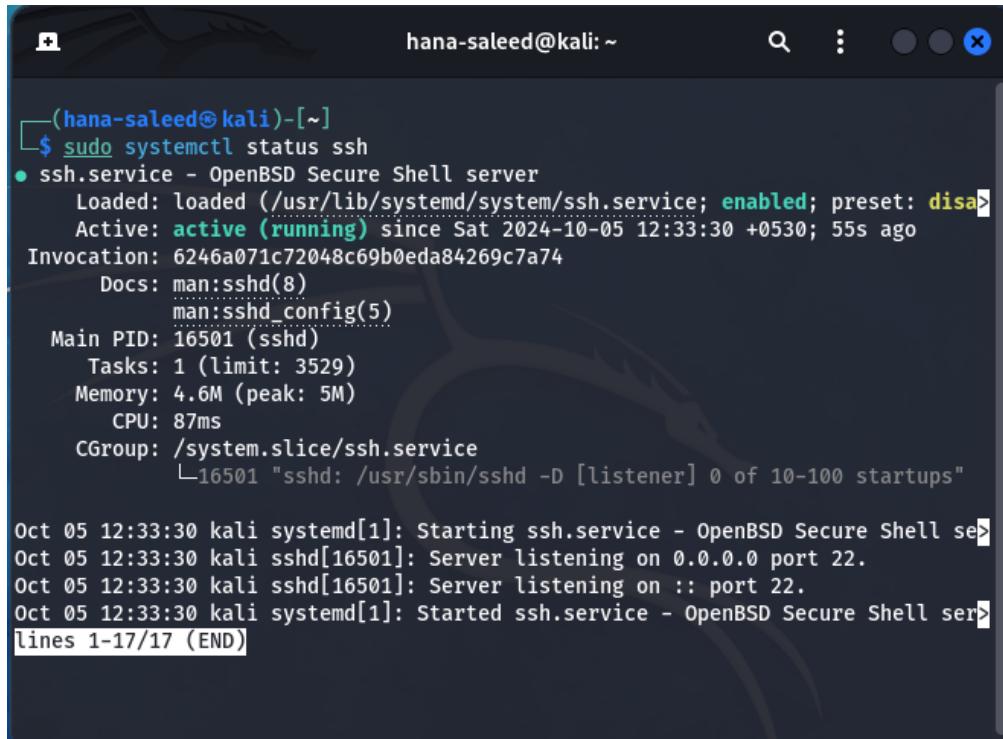
```
(hana-saleed㉿kali)-[~]
$ sudo systemctl start ssh
```

To ensure that the SSH service starts automatically every time the system boots, **sudo systemctl start ssh** has to be run.



```
(hana-saleed㉿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/ssh.service'.
Created symlink '/etc/systemd/system/multi-user.target.wants/ssh.service' → '/usr/lib/systemd/system/ssh.service'.
```

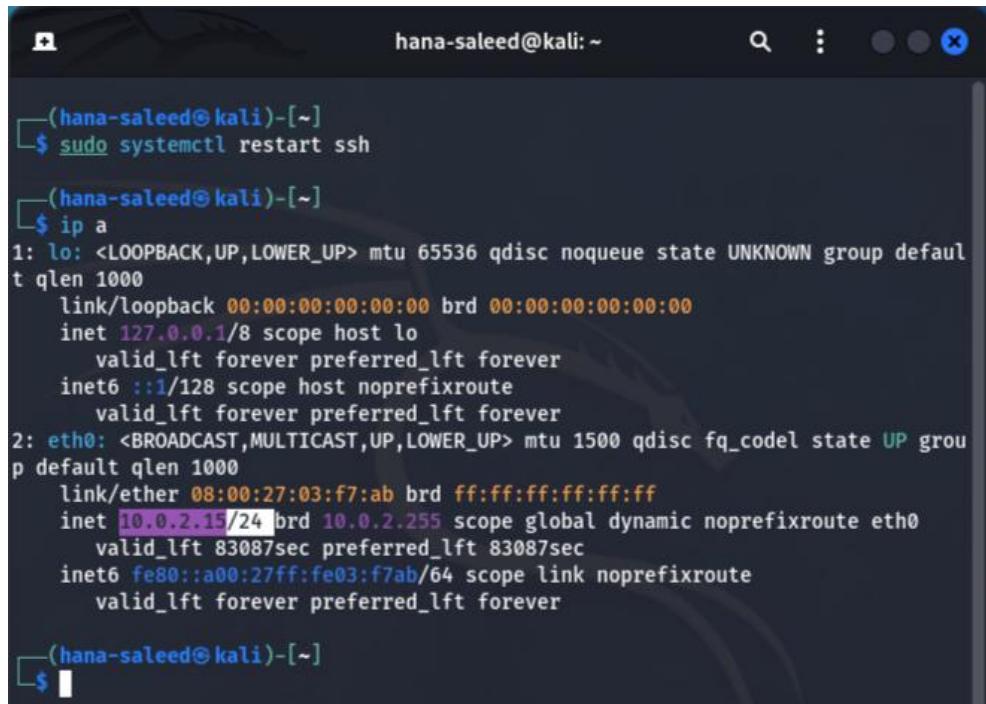
After the activation of the server the status of SSH has to be checked



```
(hana-saleed㉿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 Sat 2024-10-05 12:33:30 +0530; 55s ago
    Invocation: 6246a071c72048c69b0eda84269c7a74
      Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 16501 (sshd)
     Tasks: 1 (limit: 3529)
    Memory: 4.6M (peak: 5M)
      CPU: 87ms
     CGroup: /system.slice/ssh.service
             └─16501 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Oct 05 12:33:30 kali systemd[1]: Starting ssh.service - OpenBSD Secure Shell se...
Oct 05 12:33:30 kali sshd[16501]: Server listening on 0.0.0.0 port 22.
Oct 05 12:33:30 kali sshd[16501]: Server listening on :: port 22.
Oct 05 12:33:30 kali systemd[1]: Started ssh.service - OpenBSD Secure Shell ser...
lines 1-17/17 (END)
```

The ip address of kali is needed. Therefore, to find this information the command **ip a** has to run



```
(hana-saleed㉿kali)-[~]
$ sudo systemctl restart ssh

(hana-saleed㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
        inet6 ::1/128 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 08:00:27:03:f7:ab brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
        valid_lft 83087sec preferred_lft 83087sec
        inet6 fe80::a00:27ff:fe03:f7ab/64 scope link noprefixroute
            valid_lft forever preferred_lft forever

(hana-saleed㉿kali)-[~]
```

To connect from a client machine the command (windows)

```
PS C:\Users\DELL> ssh hana-saleed@10.0.2.15 -p 2220
ssh: connect to host 10.0.2.15 port 2220: Connection timed out
```

03.3 iptables and ACLs

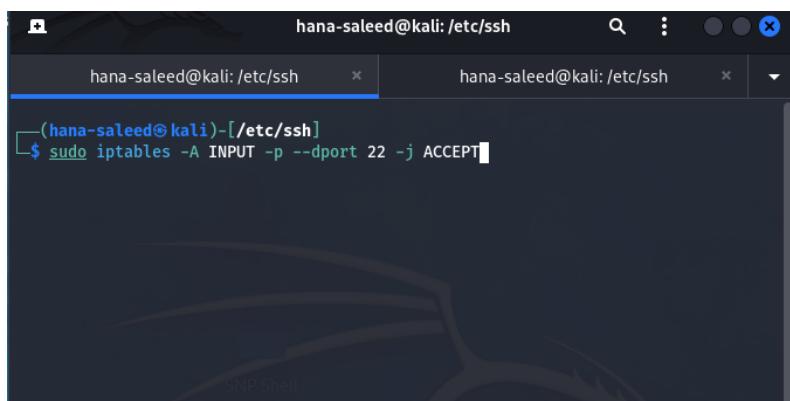
Iptables is a command-line firewall utility that uses policy chains to allow or block traffic that will be enforced by linux kernels netfilter framework. Iptables packet filtering mechanism is organized into three different kinds of structures:

❖ DefaultTables:

- a. Filter : The default table for most filtering operations
 - Input – This chain handles all packets that are destined to the server and also controls the behaviour for incoming connections.
 - Forward – This chain is used for packets routed through the system.
 - Output – This chain contains rules for packet generated locally.
- b. NAT (network Address Translation)
 - Prerouting – This chain rule alters a packet as soon as it's received.
 - Postrouting – This chain rule alters a packet as it is about to go out
 - Output – This chain rule alerts locally generated traffic.
- c. Raw : This table is used to exempt packets from connection tracking
 - Output – To alter locally generated packets.
 - Prerouting – to alter incoming connections.
- d. Mangles : This table adjusts the IP header properties of packets
 - Input – for incoming packets
 - Output – To alter locally generated packets.
 - Forward -- for packets routed through the linux box
 - Prerouting – To alter incoming connections
 - Postrouting – To alter outgoing connections.

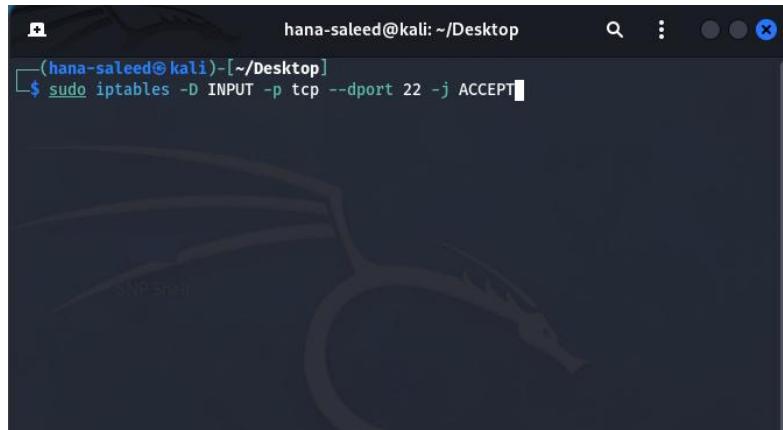
❖ Basic iptables commands syntax:

- a. **sudo iptables -L** : List current rules.
- b. **sudo iptables -A <CHAIN> <CONDITIONS> -j <TARGET>** : Add a rule



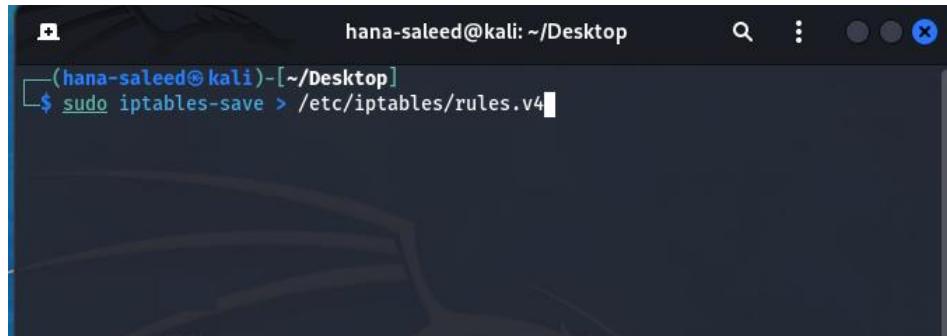
The screenshot shows a terminal window with two tabs open, both titled "hana-saleed@kali: /etc/ssh". The user is running a Kali Linux distribution. In the terminal, the command `$ sudo iptables -A INPUT -p --dport 22 -j ACCEPT` is being typed. The command has been partially entered, with the cursor at the end of "-j ACCEPT". The background of the terminal window features a dark, abstract graphic.

c. **`sudo iptables -D <CHAIN> <CONDITIONS>`** : Delete a rule



```
(hana-saleed㉿kali)-[~/Desktop]
$ sudo iptables -D INPUT -p tcp --dport 22 -j ACCEPT
```

d. **`sudo iptables-save > /etc/iptables/rules.v4`** : Save the rules



```
(hana-saleed㉿kali)-[~/Desktop]
$ sudo iptables-save > /etc/iptables/rules.v4
```

❖ Basic Firewall Rules

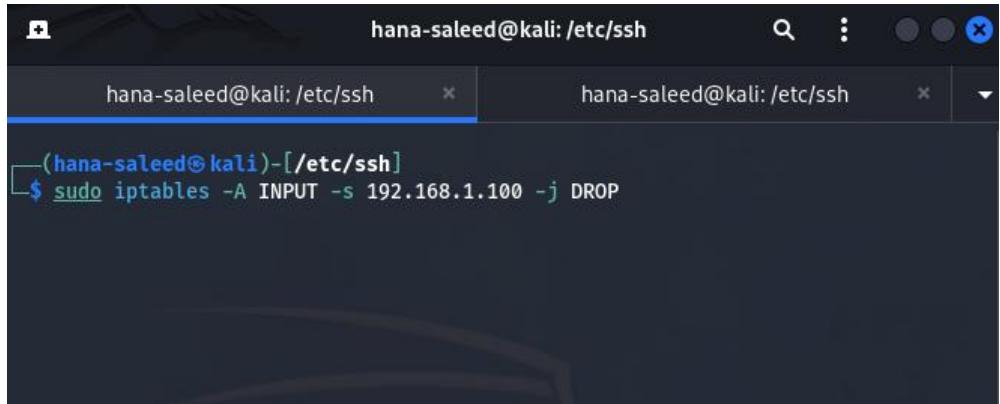
a. Allow specific traffic on a port



```
(hana-saleed㉿kali)-[~/Desktop]
$ sudo iptables -D INPUT -p tcp --dport 22 -j ACCEPT
```

This command allows incoming TCP traffic on port 22 (SSH) through firewall.

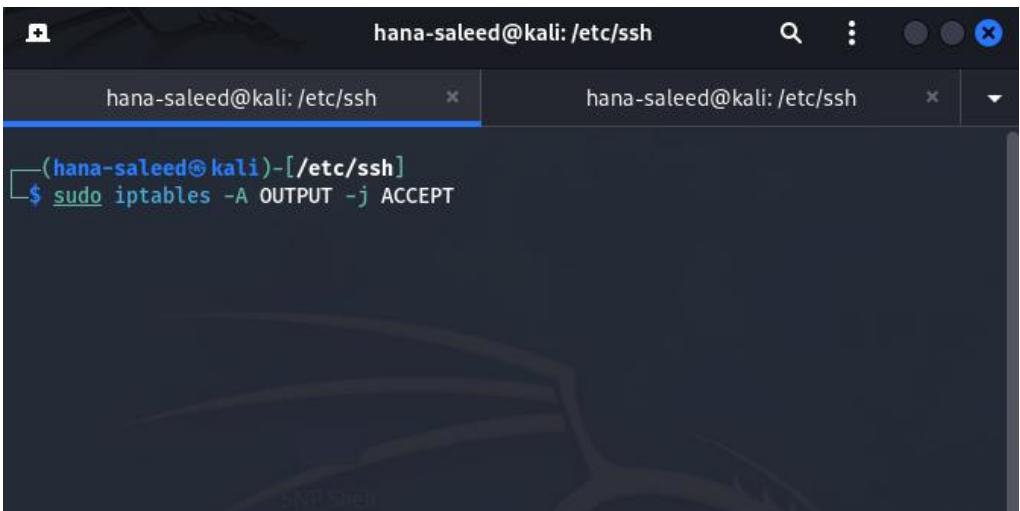
- b. Block traffic from a specific IP address.



```
(hana-saleed㉿kali)-[~/etc/ssh]
$ sudo iptables -A INPUT -s 192.168.1.100 -j DROP
```

This example will block the incoming IP address 192.168.1.100

- c. Allow all outgoing traffic.



```
(hana-saleed㉿kali)-[~/etc/ssh]
$ sudo iptables -A OUTPUT -j ACCEPT
```

An access control list (ACL) contains rules that grant or deny access to certain digital environments. It is a list of permissions that dictate what a user has access to and what types of operations they are allowed to do with that access. An **Allow** or **Deny** rules either permit or deny access.

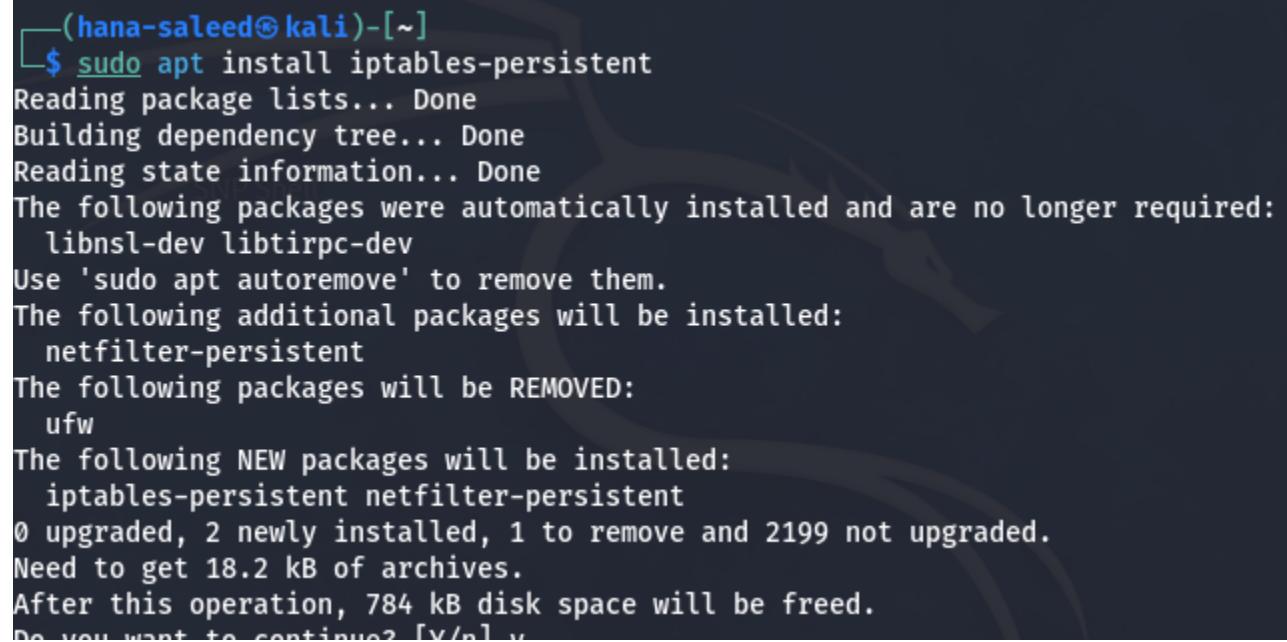


A screenshot of a terminal window titled "hana-saleed@kali: ~/Desktop". The command entered is "\$ permit ip 192.168.1.100 any". The terminal has a dark background with light-colored text and standard window controls at the top.

```
(hana-saleed@kali)-[~/Desktop]
$ permit ip 192.168.1.100 any
```

i. Web Server Security

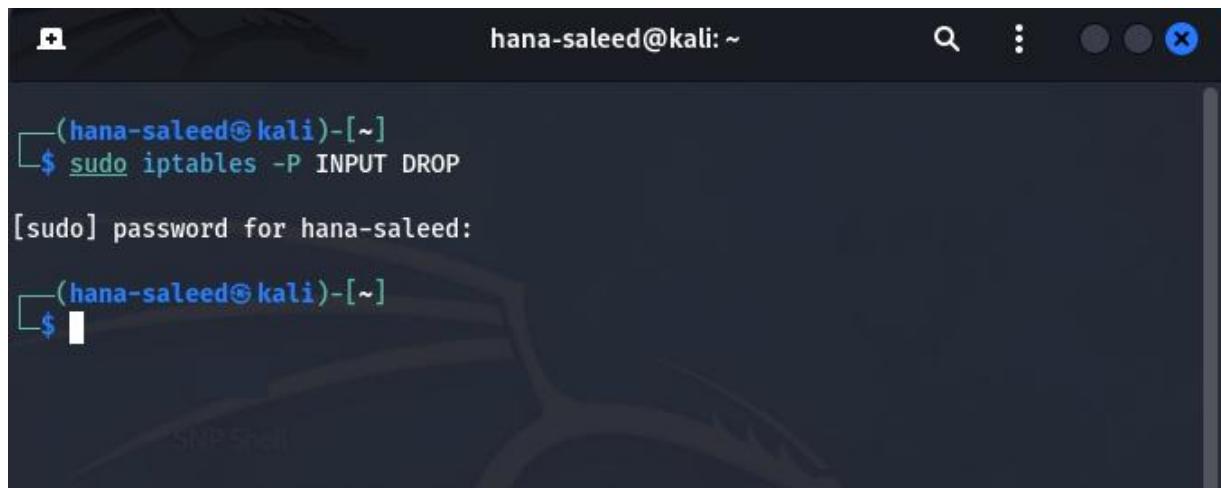
Install the package using the command



A screenshot of a terminal window showing the output of the command "\$ sudo apt install iptables-persistent". The terminal shows the package being installed, dependencies being built, and packages being removed. It also lists new packages to be installed and asks if the user wants to continue.

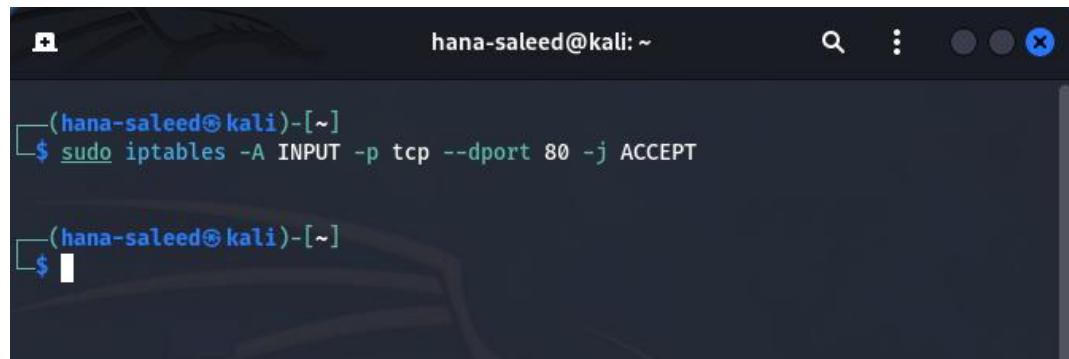
```
(hana-saleed@kali)-[~]
$ sudo apt install iptables-persistent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  netfilter-persistent
The following packages will be REMOVED:
  ufw
The following NEW packages will be installed:
  iptables-persistent netfilter-persistent
0 upgraded, 2 newly installed, 1 to remove and 2199 not upgraded.
Need to get 18.2 kB of archives.
After this operation, 784 kB disk space will be freed.
Do you want to continue? [y/n] y
```

To allow only incoming traffic port 80 (HTTP) and port 443(HTTPS), First we have to block all incoming traffic unless explicitly allowed



```
hanna-saleed@kali: ~
└─(hanna-saleed㉿kali)-[~]
$ sudo iptables -P INPUT DROP
[sudo] password for hanna-saleed:
└─(hanna-saleed㉿kali)-[~]
$
```

Then we to allow web traffic in port 80 using the following command



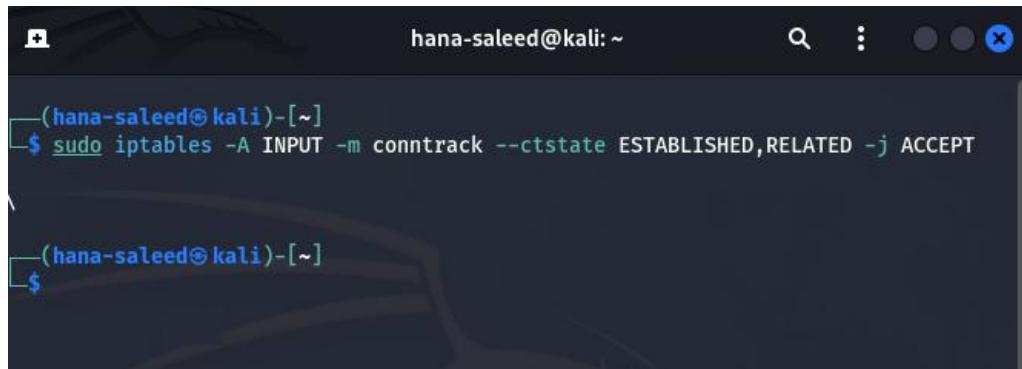
```
hanna-saleed@kali: ~
└─(hanna-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
└─(hanna-saleed㉿kali)-[~]
$
```

Then we to allow web traffic in port 443 using the following command



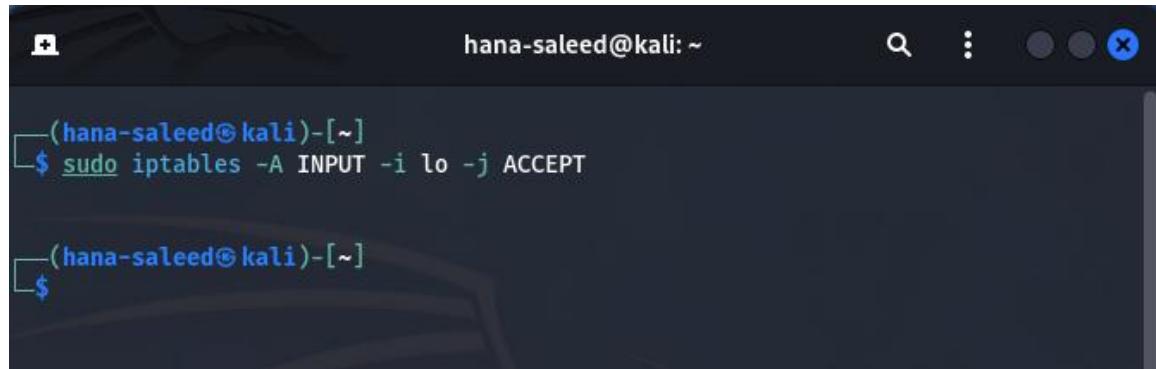
```
hanna-saleed@kali: ~
└─(hanna-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
└─(hanna-saleed㉿kali)-[~]
$
```

Established connections responses like HTTP/HTTPS are allowed back in with the command



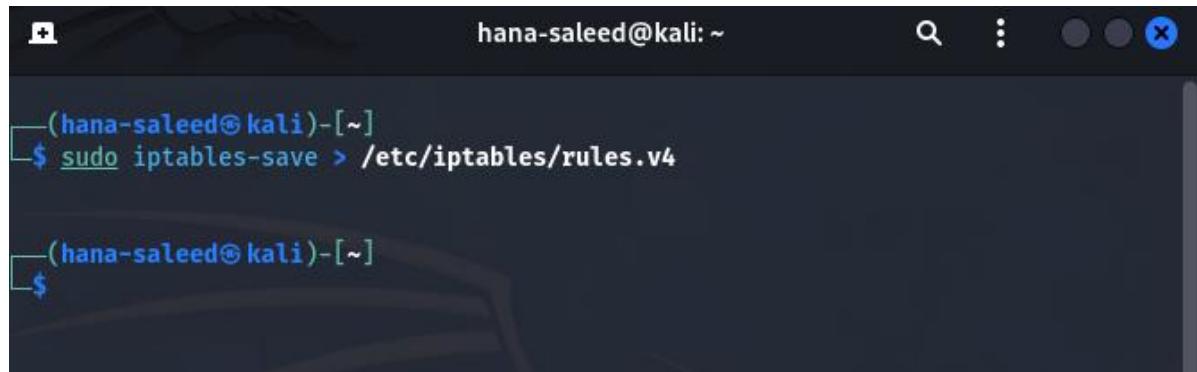
```
(hana-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Thereafter, we have to ensure the server can communicate with itself.



```
(hana-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -i lo -j ACCEPT
```

To make sure these rules persist after a reboot we have to save after editing respective permissions. **sudo chmod 644 /etc/iptables/rules.v4**



```
(hana-saleed㉿kali)-[~]
$ sudo iptables-save > /etc/iptables/rules.v4
```

Finally, we can confirm that the web server security configuration is correctly allowing incoming traffic on ports 80 and 443 as **tcp dpt:80 tcp dpt :443**

```
hana-saleed@kali: ~
└─(hana-saleed㉿kali)-[~]
$ sudo iptables -L -v -n

Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target  prot opt in     out      source          destination
      0     0 ACCEPT   6    -- *       *        0.0.0.0/0        0.0.0.0/0
      0     0 ACCEPT   6    -- *       *        0.0.0.0/0        0.0.0.0/0
      25 23104 ACCEPT   0    -- *       *        0.0.0.0/0        0.0.0.0/0
      0     0 ACCEPT   0    -- lo     *        0.0.0.0/0        0.0.0.0/0
```

ii. Remote Administration Access

Allows only a specific ip address to SSH

```
└─(hana-saleed㉿kali)-[~/Desktop/Assignment]
$ sudo iptables -A INPUT -p tcp -s 10.0.2.15 --dport 22 -j ACCEPT
```

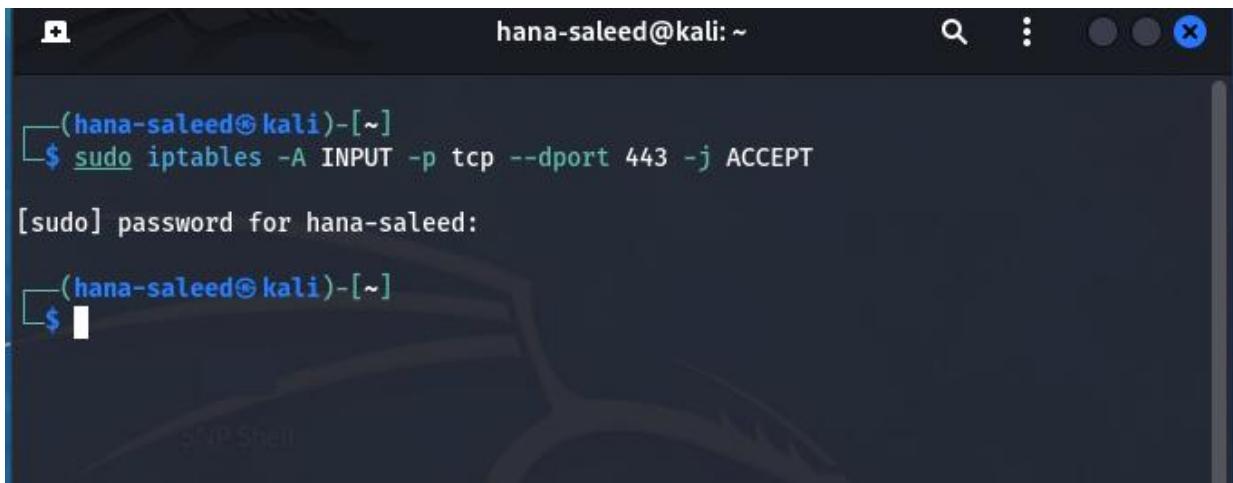
iii. Allow specific applications

Allows specific applications, permitting traffic on their associated port numbers. First, we have to drop all incoming traffic

```
hana-saleed@kali: ~
└─(hana-saleed㉿kali)-[~]
$ sudo iptables -P INPUT DROP

└─(hana-saleed㉿kali)-[~]
$
```

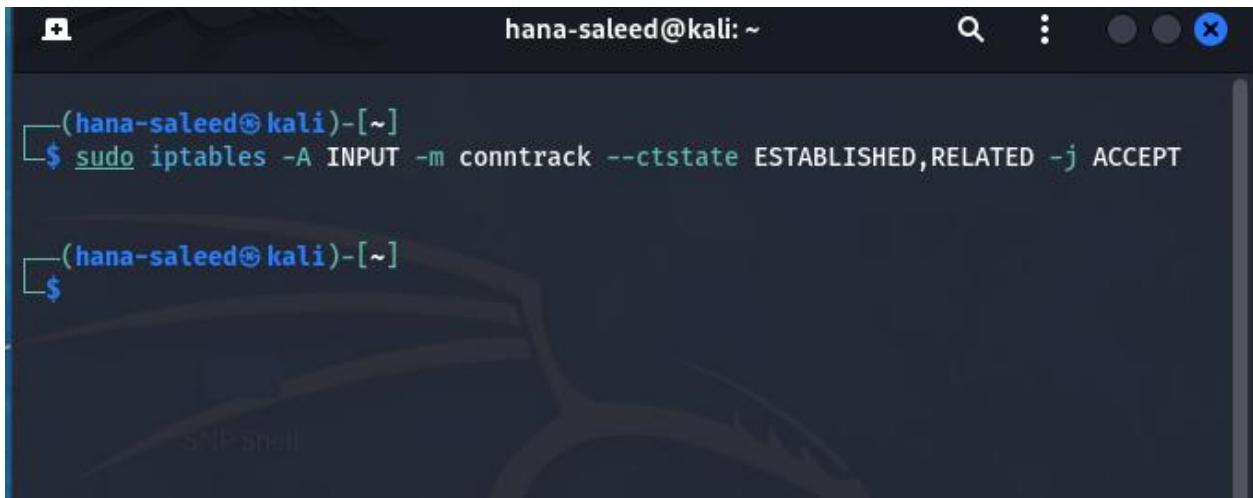
The port 443 should be allowed



A terminal window titled "hanna-saleed@kali: ~". The user runs the command `sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT`. A password prompt "[sudo] password for hanna-saleed:" appears, followed by a blank line for the password entry.

```
(hanna-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
[sudo] password for hanna-saleed:
$ 
```

Then we have to allow established connections



A terminal window titled "hanna-saleed@kali: ~". The user runs the command `sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`. A password prompt "[sudo] password for hanna-saleed:" appears, followed by a blank line for the password entry.

```
(hanna-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
$ 
```

iv. Allow Pings (ICMP Echo Request)

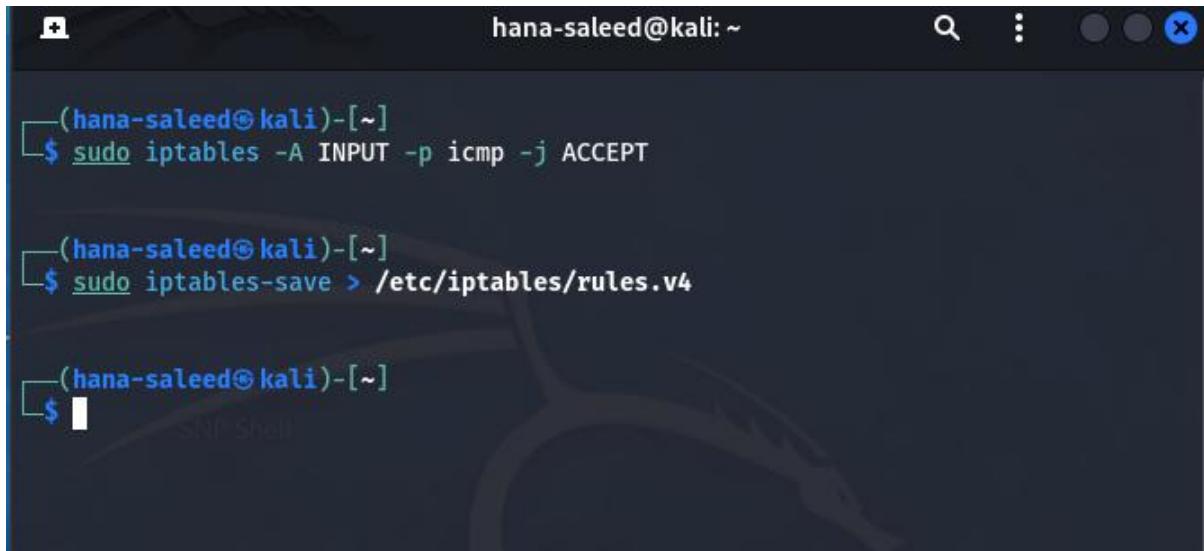
To allow ping requests we can start off with allowing incoming ICMP traffic



A terminal window titled "hanna-saleed@kali: ~". The user runs the command `sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT`. A password prompt "[sudo] password for hanna-saleed:" appears, followed by a blank line for the password entry.

```
(hanna-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
[sudo] password for hanna-saleed:
$ 
```

Next we can allow related ICMP traffic and save the rule so that it persists after a reboot



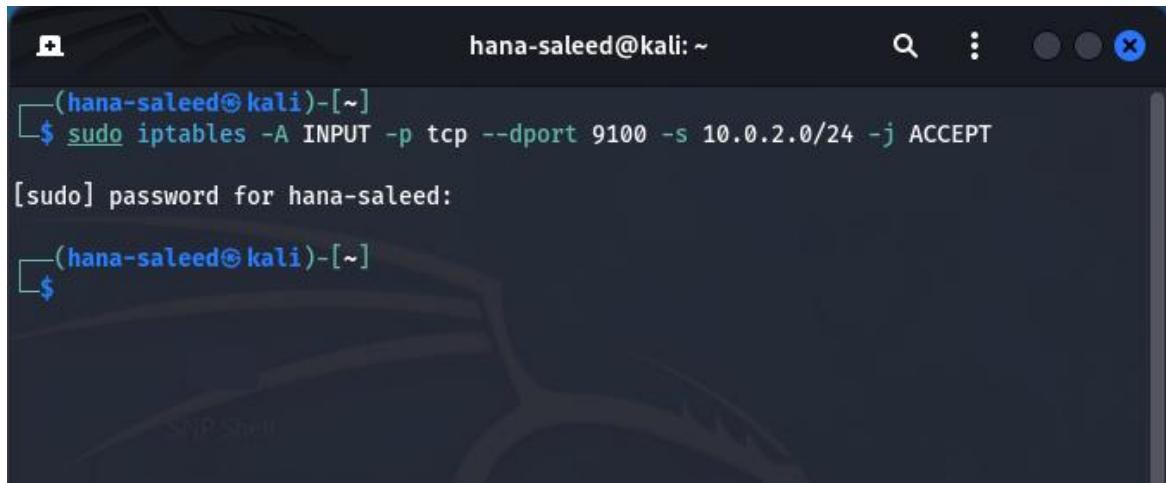
```
(hana-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -p icmp -j ACCEPT

(hana-saleed㉿kali)-[~]
$ sudo iptables-save > /etc/iptables/rules.v4

(hana-saleed㉿kali)-[~]
$
```

v. Printer Server Access

To secure the printer server, port 9100 traffic only has to be allowed from our IP address. First and foremost, we have to only allow traffic to the printer from our IP address range.

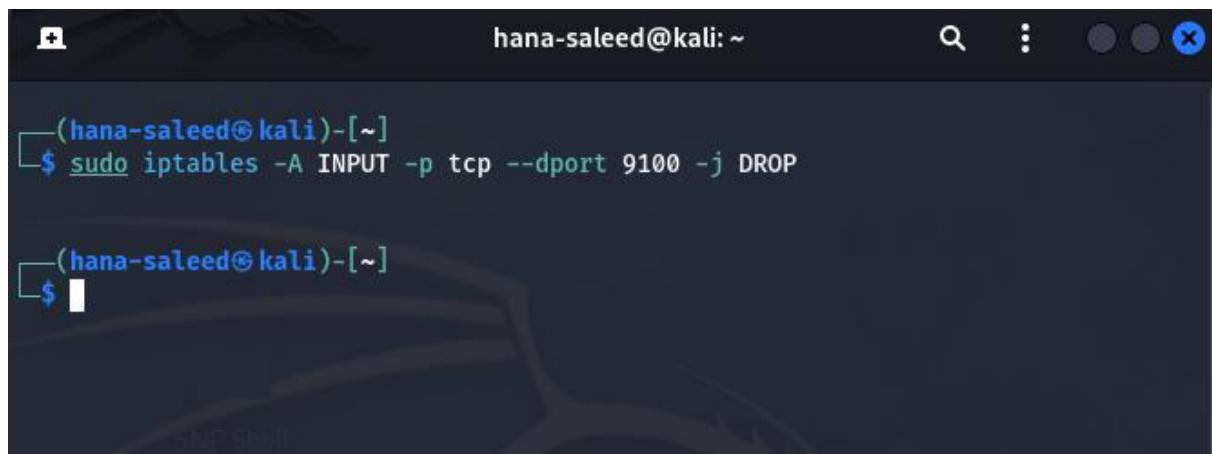


```
(hana-saleed㉿kali)-[~]
$ sudo iptables -A INPUT -p tcp --dport 9100 -s 10.0.2.0/24 -j ACCEPT

[sudo] password for hana-saleed:

(hana-saleed㉿kali)-[~]
$
```

Then all other traffic to port 9100 has to be blocked and the ongoing established connections have to be allowed



A screenshot of a terminal window titled "hana-saleed@kali: ~". The window shows a command being entered: "sudo iptables -A INPUT -p tcp --dport 9100 -j DROP". The command is partially typed, with the final closing bracket "]" still missing.

We can check if the rules have been applied by the command **sudo iptables -L -v -n**

Chain	Target	Priority	Protocol	Port	Source	Destination	
INPUT	ACCEPT	0	--	*	*	10.0.2.0/24	0.0.0.0/0
INPUT	DROP	0	--	*	*	0.0.0.0/0	0.0.0.0/0
INPUT	DROP	0	--	*	*	0.0.0.0/0	0.0.0.0/0
INPUT	ACCEPT	0	--	*	*	0.0.0.0/0	0.0.0.0/0
	ctstate RELATED,ESTABLISHED						

04. Best Practices

Understanding the security aspects of network interface configuration is crucial for protecting your system from unauthorized access, data breaches, and various network attacks

04.1 Close unused ports

We can find unused ports using the commands like netstat, nmap or ss

```
(hana-saleed㉿kali)-[~]
└─$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address      State
tcp        0      0 0.0.0.0:22              0.0.0.0:*            LISTEN
tcp6       0      0 :::22                  ::*:*               LISTEN
udp        0      0 10.0.2.15:123           0.0.0.0:*            *
udp        0      0 127.0.0.1:123          0.0.0.0:*            *
udp        0      0 0.0.0.0:123           0.0.0.0:*            *
udp6       0      0 fe80::a00:27ff:fe03:123  ::*:*
udp6       0      0 :::123                ::*:*
udp6       0      0 ::::123               ::*:*

(hana-saleed㉿kali)-[~]
└─$ nmap -sT localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-06 20:41 +0530
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00013s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

We can close the port with the syntax **sudo iptables -A INPUT -p tcp --dport port_number -j DROP**

04.2 Configure Firewalls

We can find the current rules by command **sudo iptables -L -v -n**

```
(hana-saleed㉿kali)-[~]
└─$ sudo iptables -L -v -n
[sudo] password for hana-saleed:
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

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

(hana-saleed㉿kali)-[~]
```

We can set some default policies like dropping all incoming traffic

```
(hana-saleed㉿kali)-[~]
$ sudo iptables -P INPUT DROP
```

04.3 Limit SSH Access

Editing the SSH configuration file to limit access by editing the file as

```
(hana-saleed㉿kali)-[~]
$ sudo nano /etc/ssh/sshd_config
```

```
(hana-saleed㉿kali)-[~]
$ █
```

PermitRootLogin no

AllowUsers user1 user2

04.4 Enable Automatic Security Updates

Ensure that your system automatically installs security updates.

```
(hana-saleed㉿kali)-[~]
$ sudo apt install unattended-upgrades
```

04.5 Change File Permissions

Set appropriate file permission to protect sensitive files

```
(hana-saleed㉿kali)-[~]
$ sudo chmod 600 /etc/secret.conf
```