Sri Lanka Institute of Information Technology

2024

System and Networking programming

Assignment

Year 2, Semester 1

IT23400368

MMM ARKAM

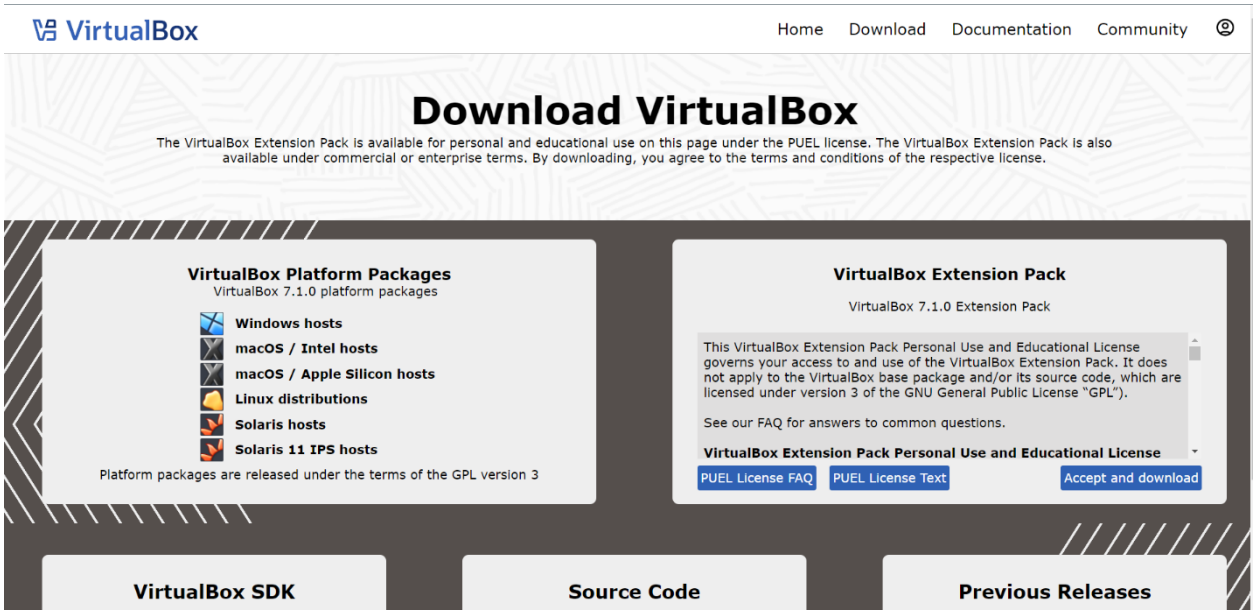
Y2.S1.WD.CS.01.01

MALABE CAMPUS

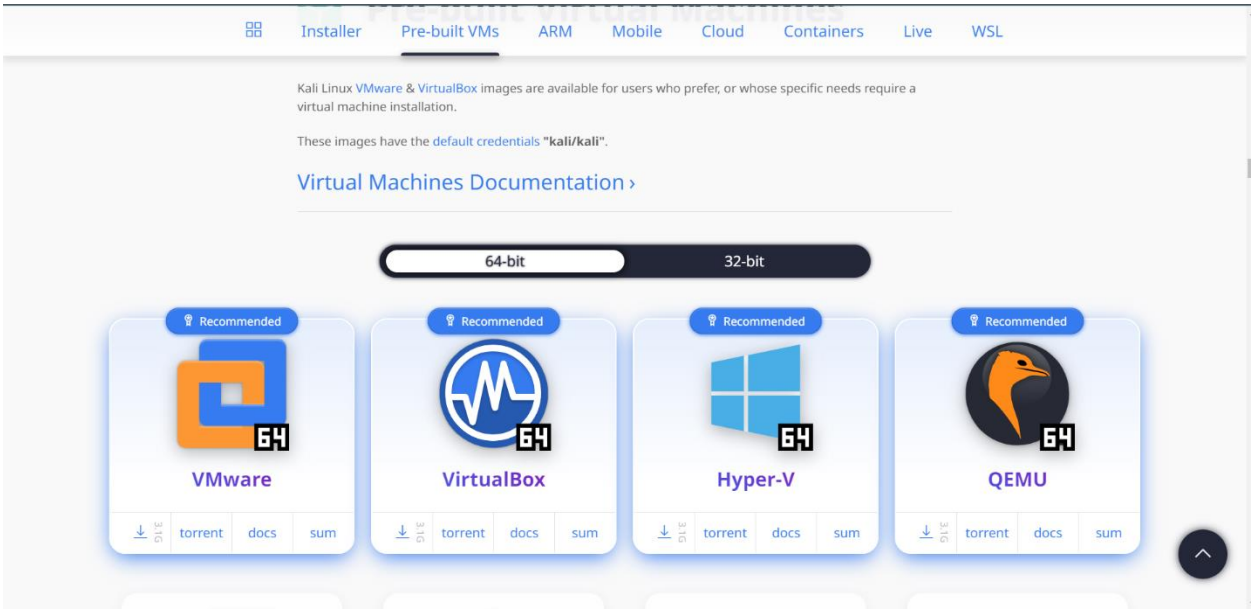
1. Basics of Linux environments.

1.1 Virtual machine setup.

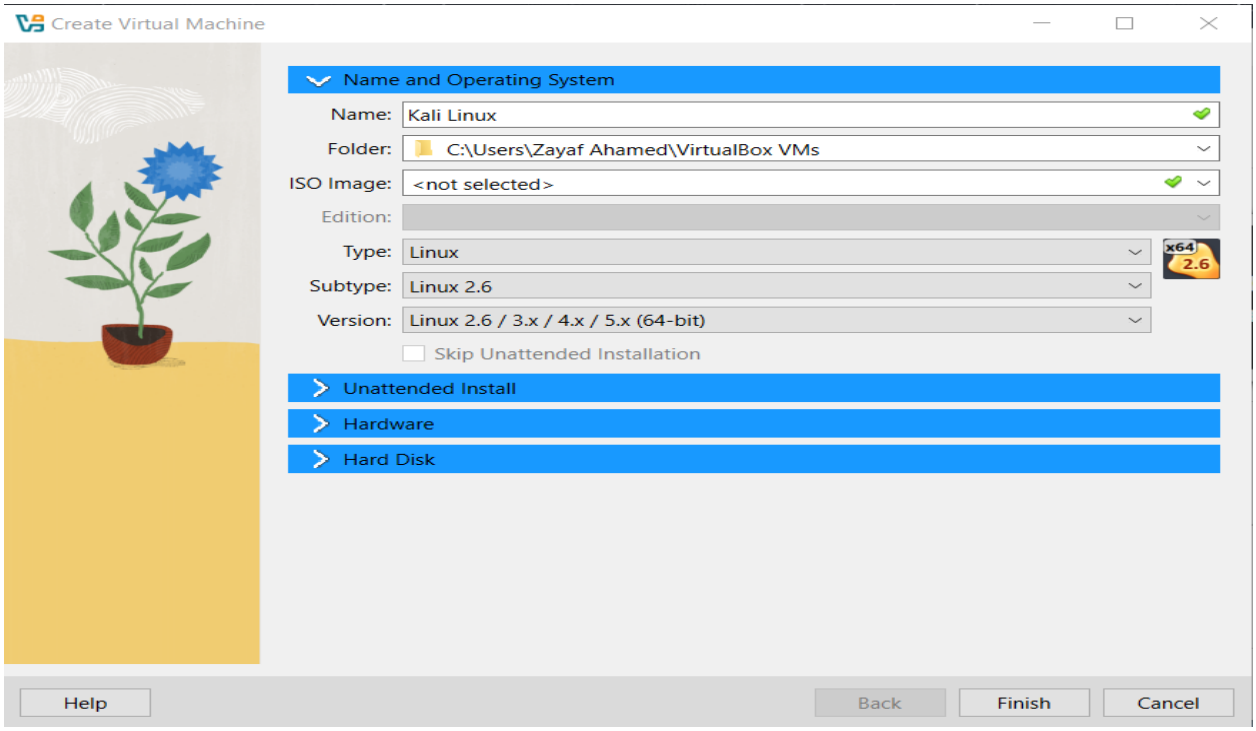
- Installation process of Oracle virtual box.
- Download the windows hosts.



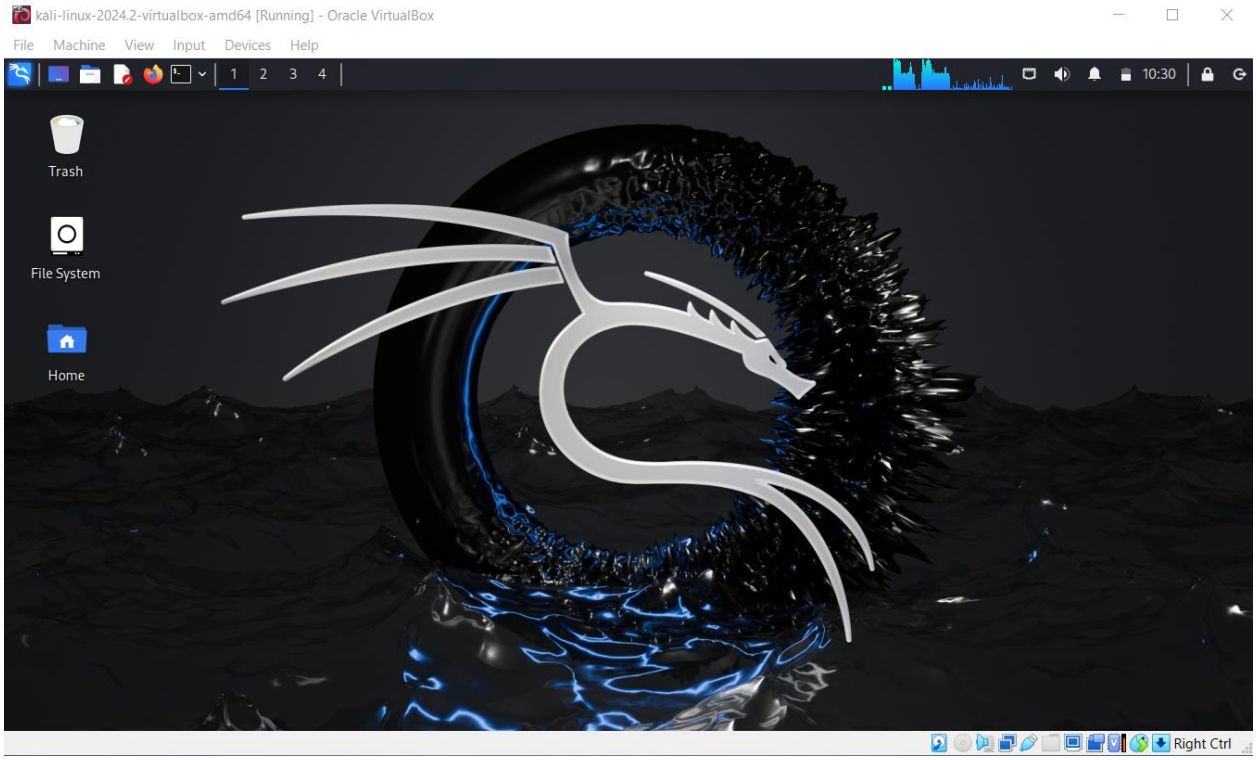
- Downloading an appropriate Linux distribution. I chose Kali.



- Configuring the VM



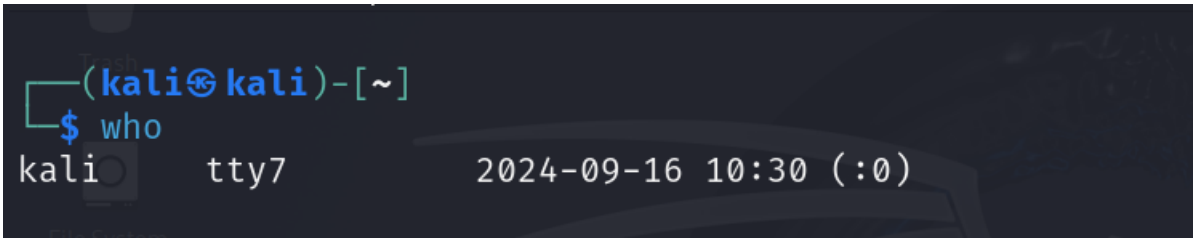
- Your virtual machine should be up and running.



1.2 Command line introduction.

Basic commands in Linux environments.

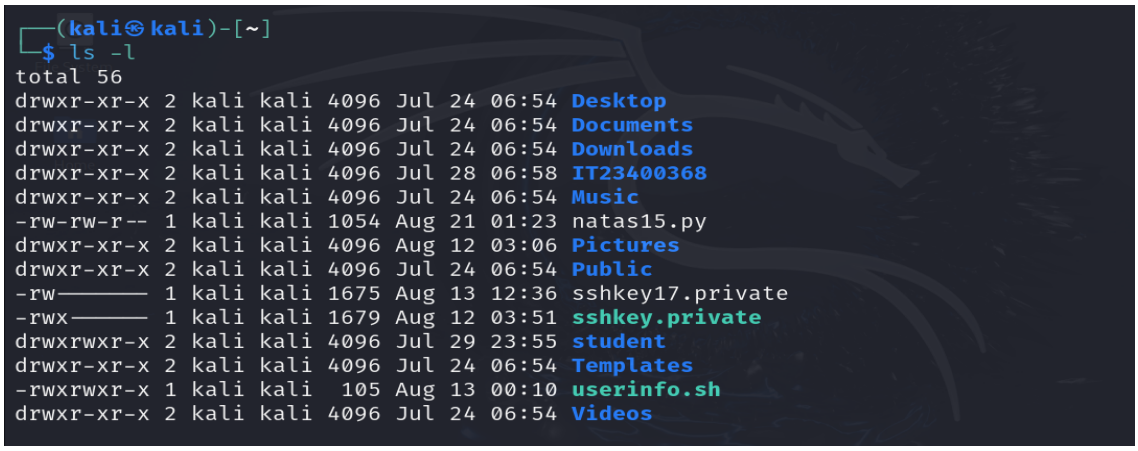
- **who:** show who is logged in.



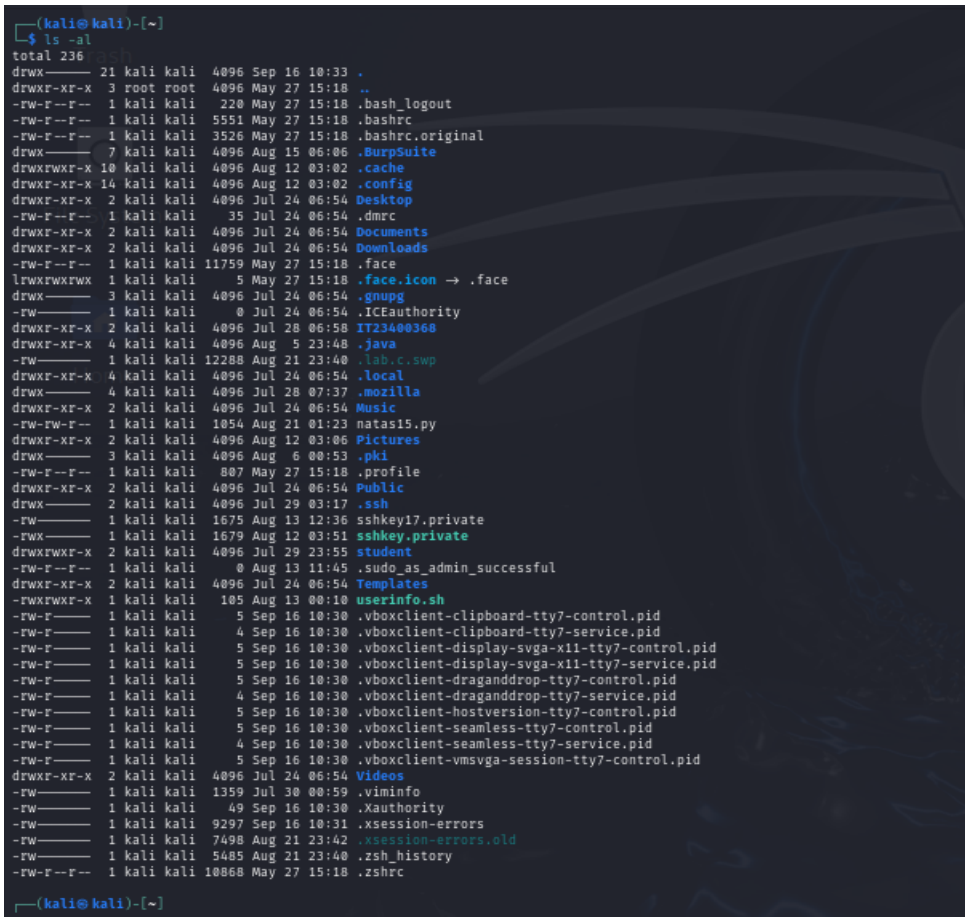
- **whoami**: Displays who is currently logged in.

- **ls**: List the directories.

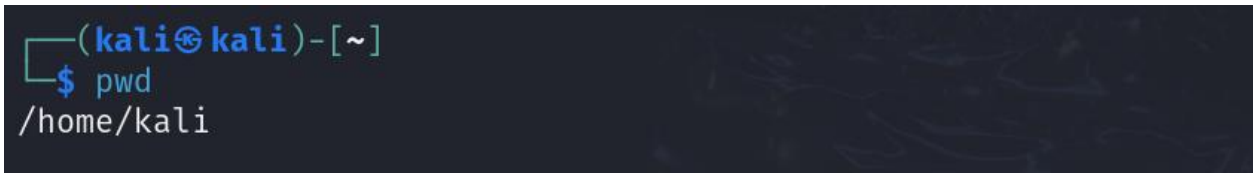
- **ls -l**: Lists files with detailed information (permissions, size, date).



- **ls -al**: Lists all files, including hidden ones, in detail.



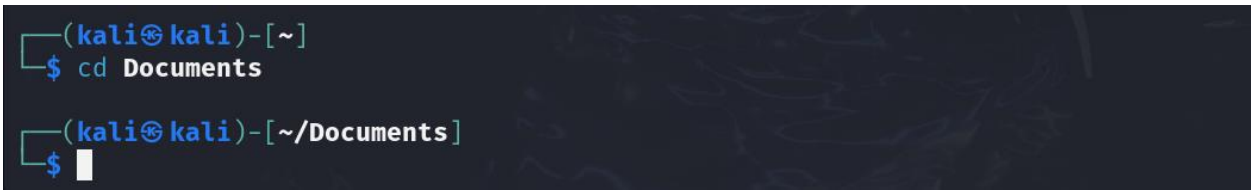
- **pwd**: Displays the current working directory path.



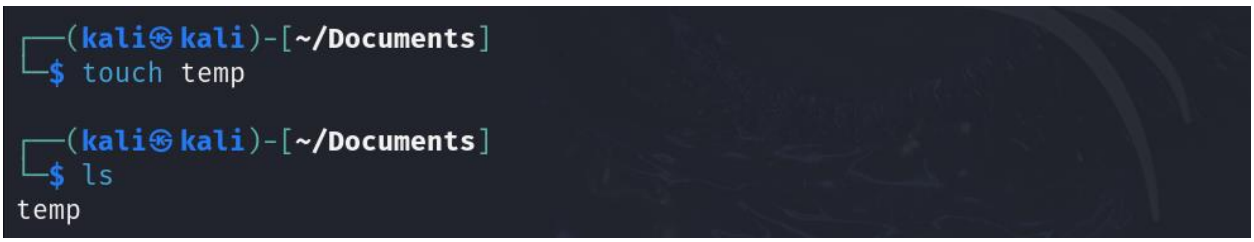
- **mkdir**: Creates a new directory.

- **rmdir**: removes a specified directory.

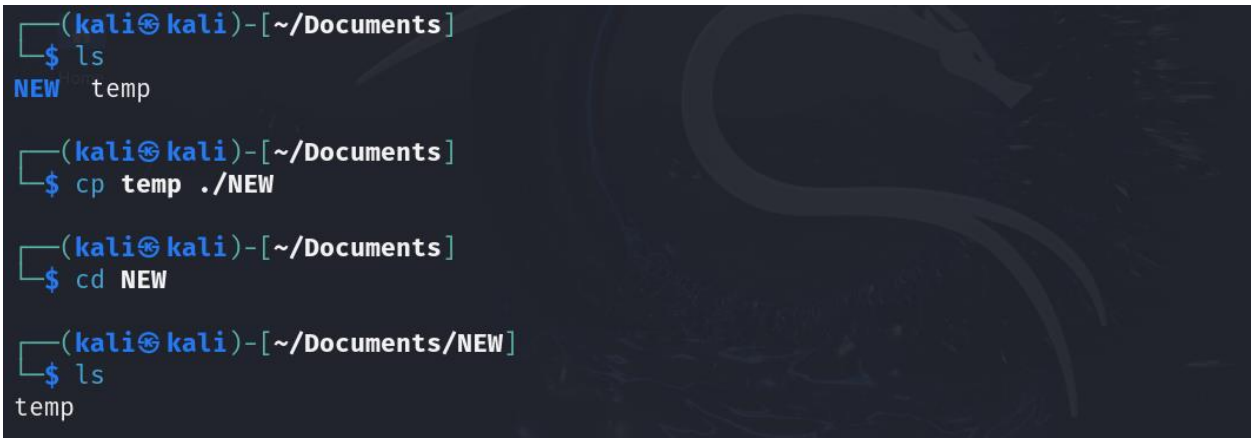
- **cd**: Changes the current directory.



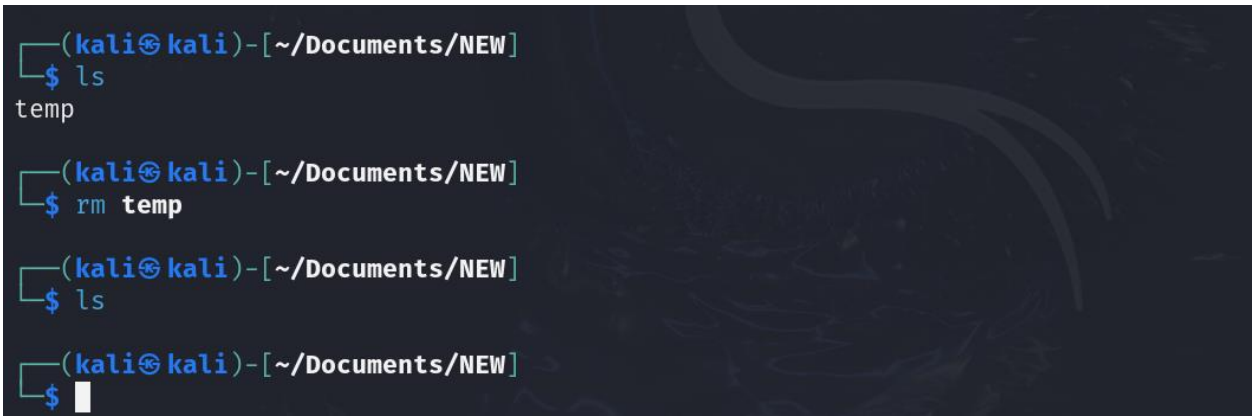
- **touch**: Creates an empty file.



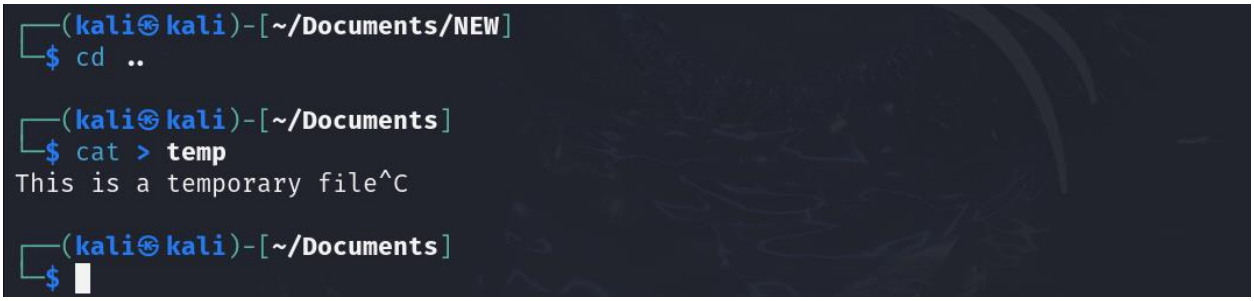
- **cp**: Copies files or directories.



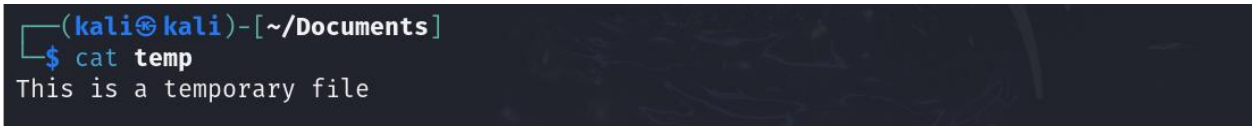
- **rm**: Removes files.



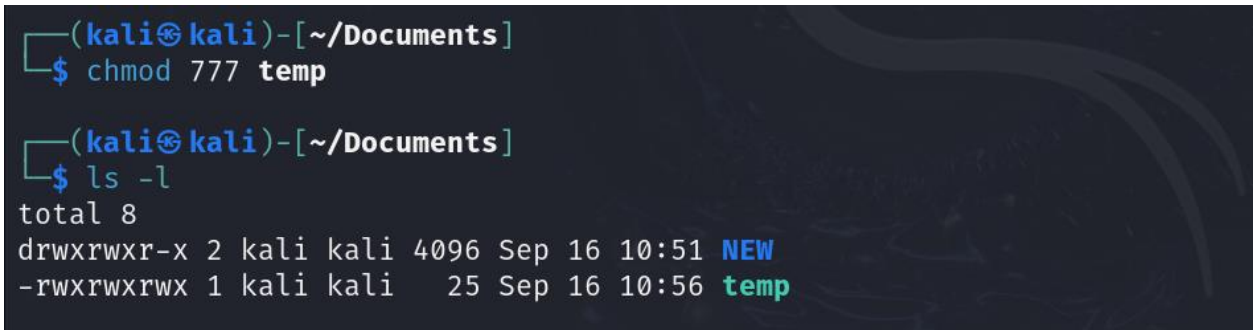
- **cat >**: Creates a new file and writes to it.



- **cat**: Displays file content or concatenates files.



- **chmod**: Changes file or directory permissions.



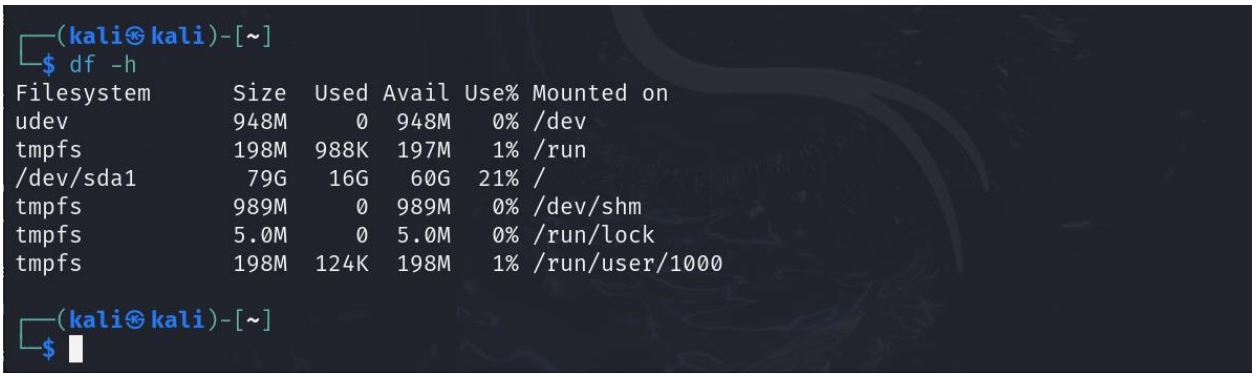
- Nano and vi editors: built is text editors used to write code.

1.3 System information and user management.

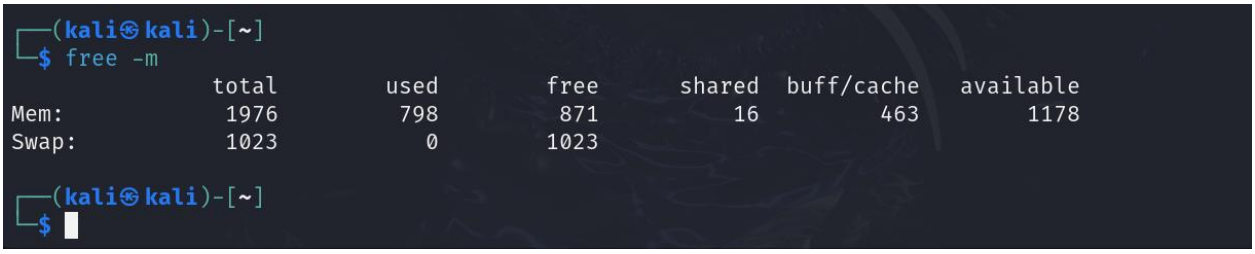
- **uname -a**: Displays detailed system information (kernel, OS, processor, etc.).

- **cat /proc/version:** Shows Linux kernel version and build information.

- **df -h**: Displays disk space usage in a human-readable format.



- **free -m**: Displays memory usage (free, used, total) in megabytes.



- **id**: Displays user and group IDs for the current or specified

passwd: Changes the user password.

- **useradd**: Creates a new user account in the system.

2. DHCP, DNS and NTP Services.

2.1 DHCP (Dynamic Host Configuration Protocol)

Step 1:

- Installing DHCP server.
- **sudo apt install isc-dhcp-server**.

Step 2:

Configure the DHCP server.

sudo nano /etc/dhcp/dhcpd.conf this command will open the configuration file.

Edit the configuration according to our network infrastructure.

```
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, 8.8.4.4;
    option domain-name "local";
}
```

After the modification save and exit from the dhcpd.conf file.

Step 3:

Configure the network interface.
sudo nano /etc/default/isc-dhcp-server. Set INTERFACESv4 to eth0

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

Step 4:

After making these changes, restart the DHCP server to apply the new changes.
This code helps to do that: **sudo systemctl restart isc-dhcp-server.**

With these settings, my DHCP server will provide IP addresses in the range 10.0.2.20 to 10.0.2.50, with the default gateway set to 10.0.2.1 and the DNS server set to 8.8.8.8.

2.2 DNS (Domain Name System)

Step 1:

Install BIND.

we can use:- **sudo apt update.**

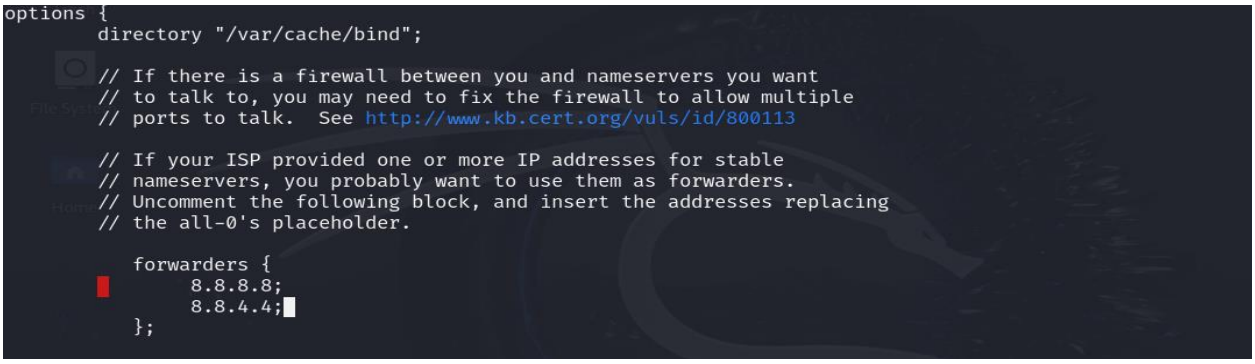
sudo apt install bind9.

Step 2:

After the installation. We must configure BIND to use our local DNS server or a public DNS server like Google DNS (8.8.8.8).

Add google DNS server as a forwarder using this command.

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



Step 3:

After saving changes, Restart BIND to apply them.

we can use **sudo systemctl restart named.**

Step 4:

Ensure BIND is running without issues.

we can use **sudo systemctl status named.**

2.3 NTP(Network Time Protocol)

Step 1:

Install the NTP server.
we can use **sudo apt install ntp.**

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install ntp
[sudo] password for kali:
Installing:
  ntp

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 990
  Download size: 23.4 kB
  Space needed: 69.6 kB / 63.5 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 7s (3,570 B/s)
Selecting previously unselected package ntp.
(Reading database ... 391507 files and directories currently installed.)
Preparing to unpack .../ntp_1%3a4.2.8p15+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) ...
```

Step 2:

Configure the NTP client.
Ensure the following lines are present to use public NTP servers.

We can use: **sudo nano /etc/ntpsec/ntp.conf.**

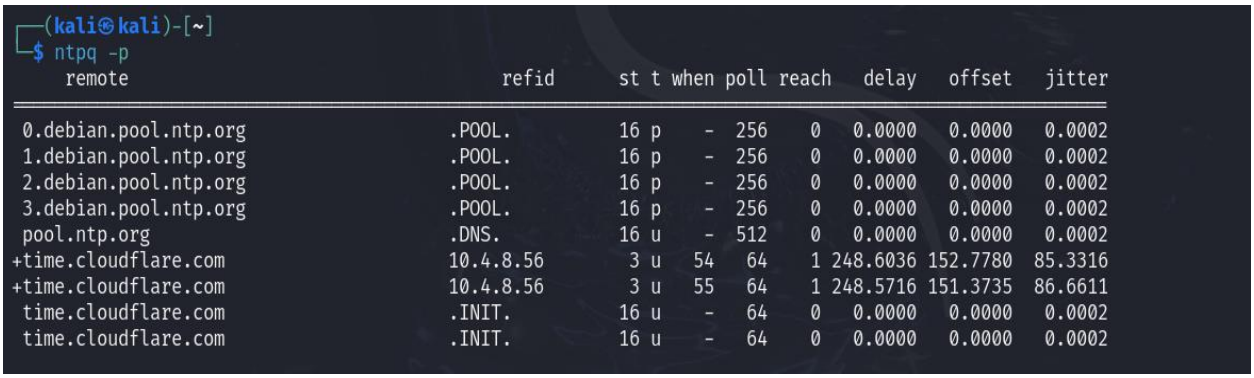
```
# pool.ntp.org maps to about 1000 low-stratum NTP servers.  Your server will
# pick a different set every time it starts up.  Please consider joining the
# pool: <https://www.pool.ntp.org/join.html>
pool 0.debian.pool.ntp.org iburst
pool 1.debian.pool.ntp.org iburst
pool 2.debian.pool.ntp.org iburst
pool 3.debian.pool.ntp.org iburst
server pool.ntp.org
# Access control configuration; see /usr/share/doc/ntpsec-doc/html/accopt.html
# for details.
```

Step 3:

After saving changes, Restart NTP to apply them.
we can use: **sudo systemctl restart ntpsec**.

Step 4:

Finally, verify if the time synchronizes.
we can use: **ntpq -p.**

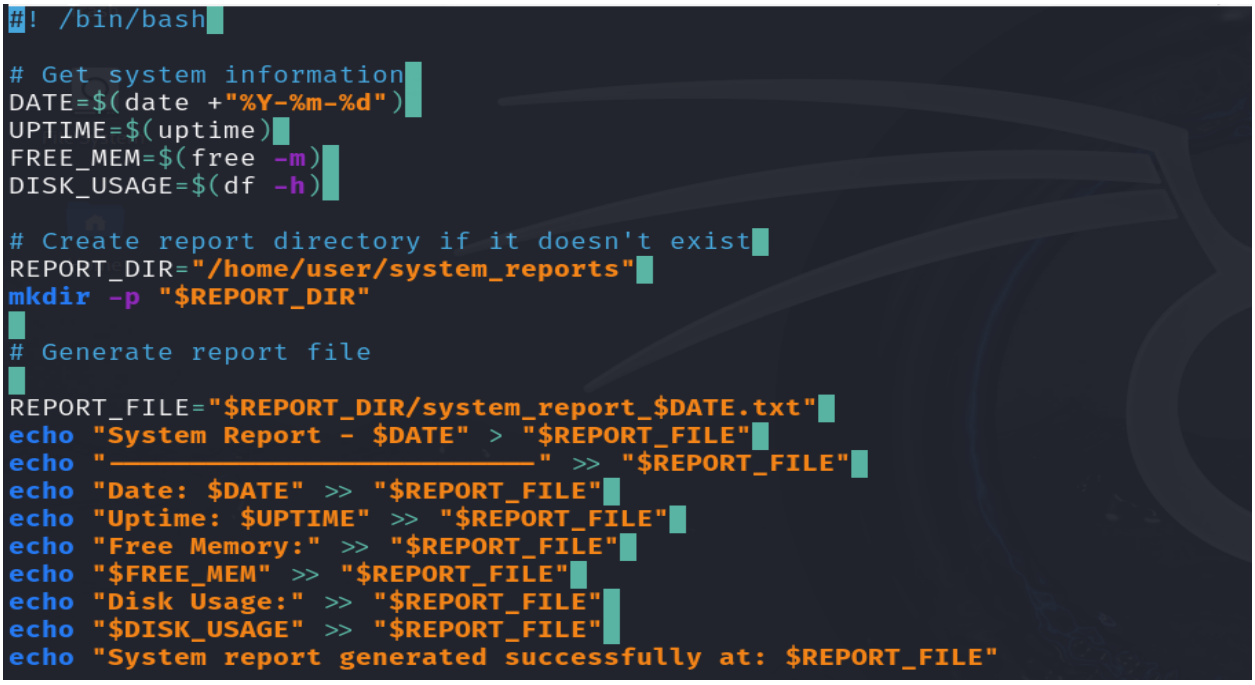


3. Shell Scripting and Security.

3.1 Shell Scripting

3.1.1 Script 1- automating a report that captures key system details every day.

```
#! /bin/bash
# Get system information
DATE=$(date +"%Y-%m-%d")
UPTIME=$(uptime)
FREE_MEM=$(free -m)
DISK_USAGE=$(df -h)
# Create report directory if it doesn't exist
REPORT_DIR="/home/user/system_reports"
mkdir -p "$REPORT_DIR"
# Generate report file
REPORT_FILE="$REPORT_DIR/system_report_$DATE.txt"
echo "System Report - $DATE" > "$REPORT_FILE"
echo "--------------------------" >> "$REPORT_FILE"
echo "Date: $DATE" >> "$REPORT_FILE"
echo "Uptime: $UPTIME" >> "$REPORT_FILE"
echo "Free Memory:" >> "$REPORT_FILE"
echo "$FREE_MEM" >> "$REPORT_FILE"
echo "Disk Usage:" >> "$REPORT_FILE"
echo "$DISK_USAGE" >> "$REPORT_FILE"
echo "System report generated successfully at: $REPORT_FILE"
```



3.1.2 Script 2 -automating the backup of a critical directory (/home/user/documents) containing important files.

#!/bin/bash
 # set source and sestination directories

SOURCE_DIR="/home/user/documents"
BACKUP_DIR="/home/user/backup/documents"

 # Create the backup directory if it doesn't exist

mkdir -p "$BACKUP_DIR"

 # Backup files with date timestamp

DATE=$(date +"%Y-%m-%d")
BACKUP_FILE="$BACKUP_DIR/documents_backup_$DATE.tar.gz"

 # Archive and compress files

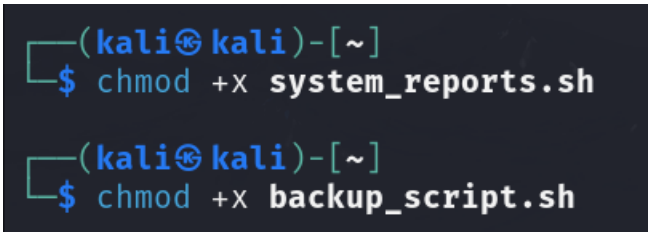
tar -czf "$BACKUP_FILE" "$SOURCE_DIR"

 # Notify user of completion

echo "Backup completed. File saved as $BACKUP_FILE"

```
#!/bin/bash

# set source and sestination directories

SOURCE_DIR="/home/user/documents"
BACKUP_DIR="/home/user/backup/documents"

# Create the backup directory if it doesn't exist

mkdir -p "$BACKUP_DIR"

# Backup files with date timestamp

DATE=$(date +"%Y-%m-%d")
BACKUP_FILE="$BACKUP_DIR/documents_backup_$DATE.tar.gz"

# Archive and compress files

tar -czf "$BACKUP_FILE" "$SOURCE_DIR"

# Notify user of completion
echo "Backup completed. File saved as $BACKUP_FILE"
```

Make the script executable using **chmod.**



These scripts will automate the tasks of capturing system details and generating reports, as well as backing up a critical directory, making the system administration tasks more efficient.

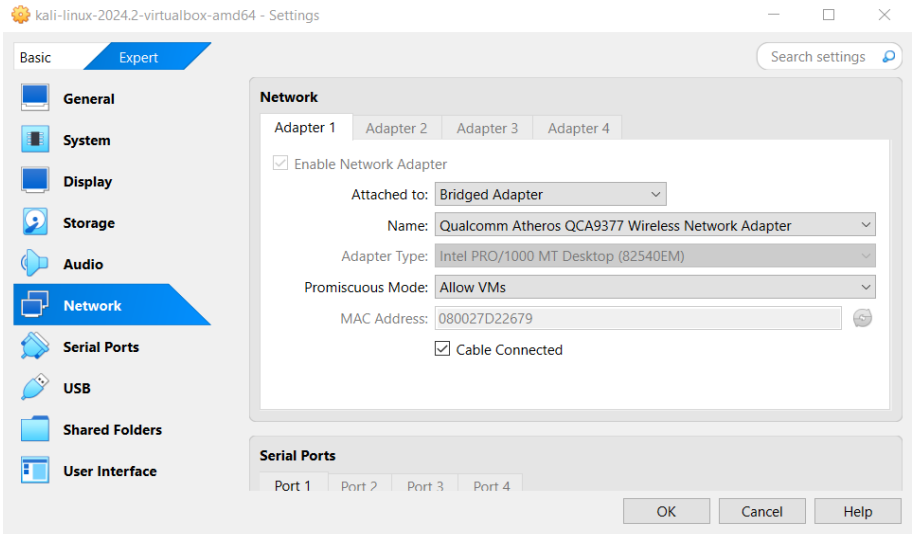
3.2 SSH (Secure Shell)

Installing SSH server.
using **sudo apt install openssh-server.**

After installing we should start the SSH service. By default, the SSH server will listen for connections on **port 22**.

Connecting to my virtual machine remotely.

- Make sure our network adapter is attached to bridge adapter.



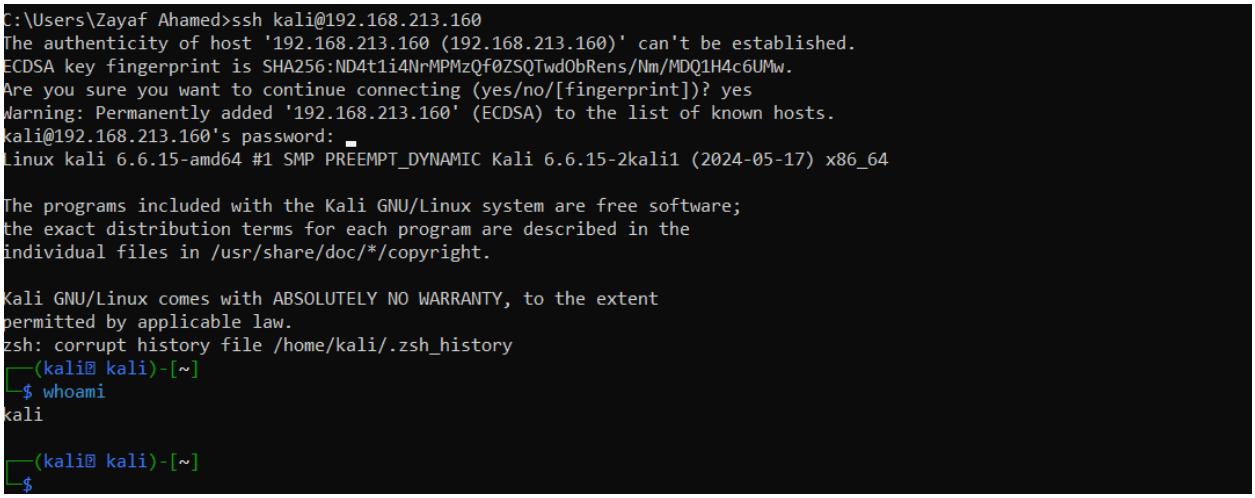
- Start the ssh server using **sudo systemctl start ssh.**
- To allow SSH traffic through the firewall (if we're using **ufw**).
  We can use **sudo ufw allow ssh**.

```
┌──(kali㉿kali)-[~]
└─$ sudo ufw allow ssh
sudo ufw enable
Rules updated
Rules updated (v6)
Firewall is active and enabled on system startup
```

- Find the virtual machines ip address using the **ip a** command in the kali linux terminal.

```
┌──(kali㉿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:d2:26:79 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 80788sec preferred_lft 80788sec
    inet 192.168.213.160/24 brd 192.168.213.255 scope global dynamic eth0
       valid_lft 3593sec preferred_lft 3593sec
    inet6 fd00::2062:370c:c3a8:2770/64 scope global dynamic noprefixroute
       valid_lft 85138sec preferred_lft 13138sec
    inet6 fe80::fe81:2669:b8b5:db7a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- In the terminal of the remote computer, we can connect using **ssh user@Ip_Address.**
- Replace **user** with our **Linux username** and **IP_Address** with the **IP address of our Linux** machine.
- In this case it is **ssh kali@192.168.213.160.**



- Successfully connected.

3.3 iptables and ACLs

- we don't need to install iptables on Kali Linux because it comes pre-installed.
- Viewing the default iptables rules.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -L -v
[sudo] password for kali:
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

┌──(kali㉿kali)-[~]
└─$
```

3.3.1 Web server security

Allow incoming traffic on port 80 (HTTP) and port 443 (HTTPS) and block all other incoming traffic.

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

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

sudo iptables -P INPUT DROP

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -P INPUT DROP
```

check if the rules were defined

```
Chain INPUT (policy DROP 0 packets, 0 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:https

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
```

This command helps to save and automatically restore iptables rules across reboots.

sudo apt install iptables-persistent

sudo netfilter-persistent save

3.3.2 Remote administration access

Allow SSH access from specific IP addresses:
include the trusted IP address.
sudo iptables -A INPUT -p tcp -s 192.168.213.64 --dport 22 -j ACCEPT

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp -s 192.168.213.64 --dport 22 -j ACCEPT
[sudo] password for kali:
```

Block SSH from other sources.
sudo iptables -A INPUT -p tcp --dport 22 -j DROP.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp --dport 22 -j DROP
```

3.3.3 Allow specific applications

Allow traffic for specific applications based on known port numbers.
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT

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

3.3.4 Allow pings (ICMP echo requests)

create a rule using iptables to permit ICMP traffic.
sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT.

```
┌──(kali㉿kali)-[~]
└─$ sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
```

3.3.5 printer server access

Allow printing traffic (port 9100) only from specific IP addresses on your local network, blocking all external access.

sudo iptables -A INPUT -p tcp -s 10.0.2.0/24 --dport 9100 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 9100 -j DROP

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

4. Best Practices

4.1 Disable root login over SSH

Allowing root login over SSH is risky because it gives full control of the system to anyone who accesses it. Disabling root login adds an extra layer of protection to keep our system more secure.

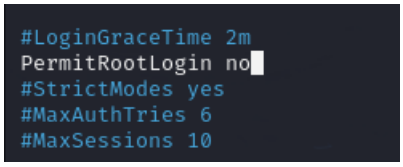
implementation:

step 1: Open the SSH configuration file.

sudo nano /etc/ssh/sshd_config

step2: disable root login

PermitRootLogin no



step3: Restart the SSH service to apply the changes

sudo systemctl restart sshd

4.2 Regularly update and patch network-related software.

Keeping our network software up-to-date helps fix security issues and lowers the chances of our system being attacked.

Implementation:

step1: Regularly check for updates and apply them.

sudo apt update.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt update
[sudo] password for kali:
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) [49.1 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 [193 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 [10.8 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [22.8 kB]
Fetched 70.8 MB in 2min 43s (434 kB/s)
1307 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

4.3 Use strong password.

- if we are planning Linux server security, one key practice is to set a powerful password.

Step 1: changing the password.

sudo passwd

Type the current password and after enter the new password.

```
┌──(kali㉿kali)-[~]
└─$ sudo passwd
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
```

- Create passwords with at least ten characters, including special characters and a combination of uppercase and lowercase letters.
- It makes brute force attack impractical.

4.4 Restrict Access to Configuration Files.

- This is because network configuration files store valuable information, such as IP addresses, routing rules, and DNS settings. Improper access or editing may lead to network downtime or security breaches.

Step 1: Limit file permission.

Set permissions such that only the root user has access and can modify these files. This will prevent normal users or even harmful programs from making changes in the configurations.

We are following one of the fundamental best practices in security-known as the **principle of least privilege-in** implementing restriction to configuration files.

4.5 Enable Logging and Monitoring.

Logging and monitoring of network interface configurations, when enabled, will help in the detection of unauthorized changes.

Step 1: Install rsyslog server.

```
┌──(kali㉿kali)-[~]
└─$ sudo apt install rsyslog
The following packages were automatically installed and are no longer required:
  fonts-liberation2         libboost-thread1.83.0  libgfrpc0          libjsoncpp25   libpostproc57  libu2f-udev              python3-mistune0
  ibverbs-providers         libcephfs2             libgfxdr0          libndctl6      librados2      libusbmuxd6              python3-pendulum
  libassuan0                libdaxctl1             libglusterfs0      libplacebo338  librdmacm1t64  openjdk-17-jre           python3-pytzdata
  libavfilter9              libgeos3.12.1t64       libibverbs1        libplist3      libre2-10      openjdk-17-jre-headless  rwho
  libboost-iostreams1.83.0  libgfapi0              libimobiledevice6  libpmem1       libroc0.3      python3-diskcache        rwhod
Use 'sudo apt autoremove' to remove them.

Installing:
  rsyslog

Installing dependencies:
  libestr0  libfastjson4  liblognorm5

Suggested packages:
  rsyslog-mysql  | rsyslog-pgsql  rsyslog-mongodb  rsyslog-doc  rsyslog-openssl  | rsyslog-gnutls  rsyslog-gssapi  rsyslog-relp

Summary:
  Upgrading: 0, Installing: 4, Removing: 0, Not Upgrading: 187
  Download size: 848 kB
  Space needed: 2,317 kB / 62.1 GB available

Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 libestr0 amd64 0.1.11-1+b1 [9,236 B]
Get:3 http://http.kali.org/kali kali-rolling/main amd64 liblognorm5 amd64 2.0.6-4+b1 [65.8 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 libfastjson4 amd64 1.2304.0-1+b1 [29.0 kB]
Get:4 http://xsrv.moratelindo.io/kali kali-rolling/main amd64 rsyslog amd64 8.2406.0-1 [744 kB]
Fetched 848 kB in 13s (65.1 kB/s)
```

Step 2:

- Configure the server by adding the information about the file we want to monitor.
- I chose **/etc/network/interfaces**.

```
module(load="imfile")
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")
input(type="imfile"
      File="/etc/network/interfaces"
      Tag="net-config-change"
      Severity="warning")
```

Step 3:

- After the configuration we should restart the service.
  Using **sudo systemctl restart rsyslog.**

Now the service should be running without errors.

Step 4: Verify logging

Make a change to **the /etc/network/interfaces** file and check the syslog to ensure it records changes: **sudo tail -f /var/log/syslog**.