



Tech It Easy

Confidential – Capstone Project Cover

Capstone Project

Enterprise IT – System & Network Design



CAPSTONE LUXURY EDITION

Designed by
Tech It Easy Graphic Team

Project Tech it Easy – Gold Edition

Group members

names	Surname
Lindiwe	Mkuzangwe
Konke	Maphisa
Zikhona	Bam
Siyabonga	Mdaweni
Adivhaho	Ndou

Table of Contents

System Setup	3
Operating system Configuration.....	3
Kali Linux configuration	3
Os image installation.....	4
password	5
Virtual machine	5
Desktop	6
Creating Sudo user	7
Grant Administrative Privileges.....	7
Verify Group Membership.....	8
Validate Sudo Access.....	8
Testing user permissions	9
.....	9
Creating user without root permission.....	9
Verify Account Creation and Privileges.....	10
Test the Account	10
Conclusion and Key Findings	11
Networking	12
Virtual network	12
Netcat to send messages.....	12
File transfer between two virtual machines	14
Network diagram	17
Basic local network setup	18
Assigning IP by editing interface file	18
Assign IPs manually or using DHCP	20
Mac address spoofing	20
Basic configuration	22
Ping test	22
ifconfig.....	22
Traceroute	23
Server & Services.....	24

Server and layout	24
Installing the Apache server	24
Starting and enabling Apache.....	24
Testing Apache	25
Custom intranet page.....	25
Security.....	27
Enabling firewall	27
Installing ufw.....	27
Enable UFW and set default rules	28
Allow specific services	29
Checking status	29
Set Up Basic Antivirus.....	30
Change Default Passwords	31
Change user password	31
Audit user accounts	31
Document Security Settings.....	32
Troubleshooting & Documentation.....	34
Problem 1 Wireshark won't launch.....	34
Installing Wireshark.....	34
problem 2 slow internet access	35
Kali Linux 2025.3 XFCE — Troubleshooting Log	38
What we have learned.....	39
What We'd Do Differently Next Time	39

System Setup

Operating system Configuration

Desktop: Asus AIO

Operating system: kali Linux 2025.3

Processor: 12th gen intel(R)

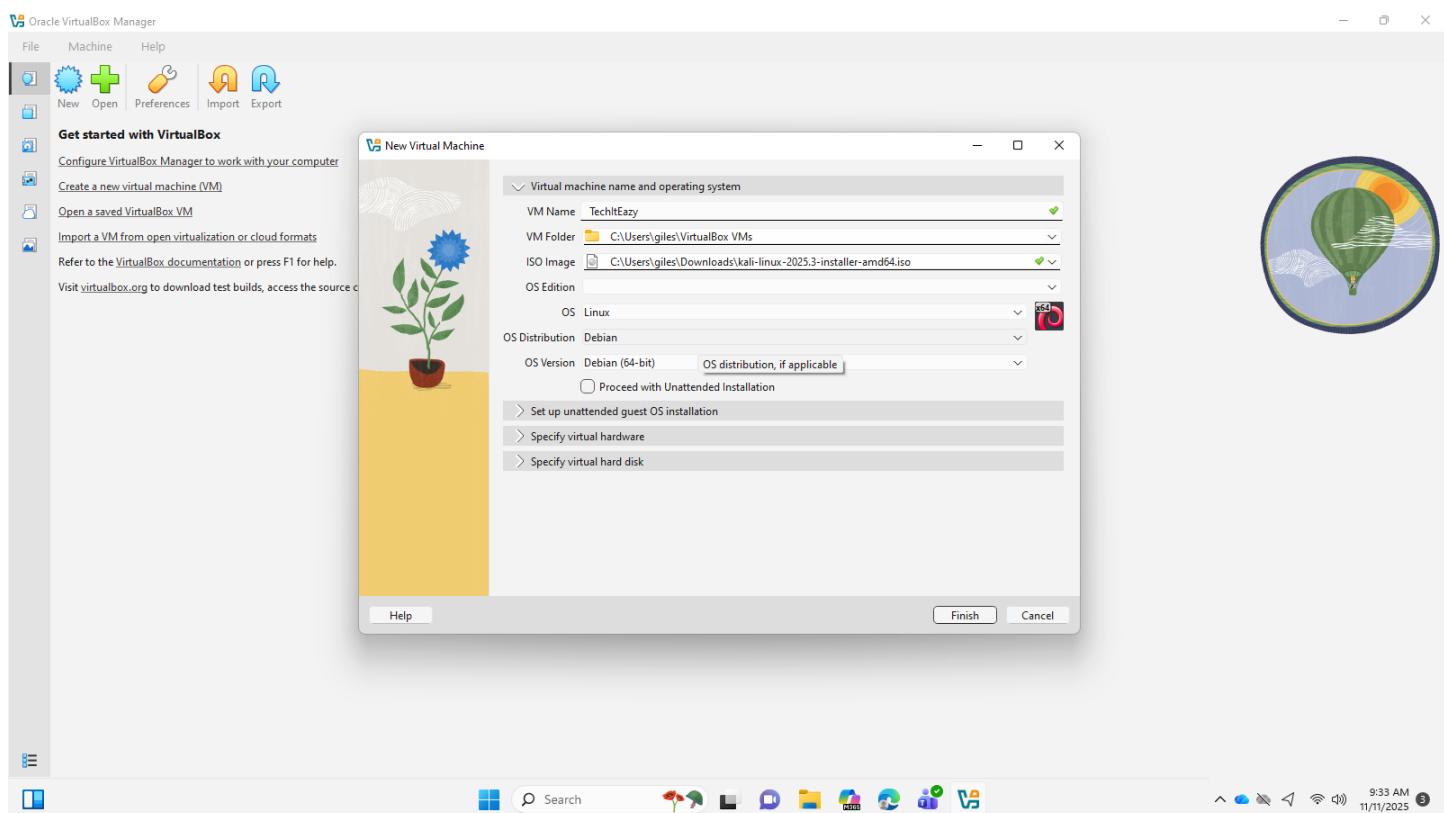
Installed ram: 8.00 GB

System type: 64-bit x64 base CPU

We went with **Kali Linux** because it's practical, flexible, and security-minded — exactly what an IT Support Specialist needs when setting up systems, troubleshooting, and protecting a company's network. It gives us the freedom to configure users and servers, test networking tools, and apply strong security measures all in one environment. In short, Kali makes it easier to learn the real-world skills of keeping systems running smoothly while staying safe, which is why it's the right fit for this project.

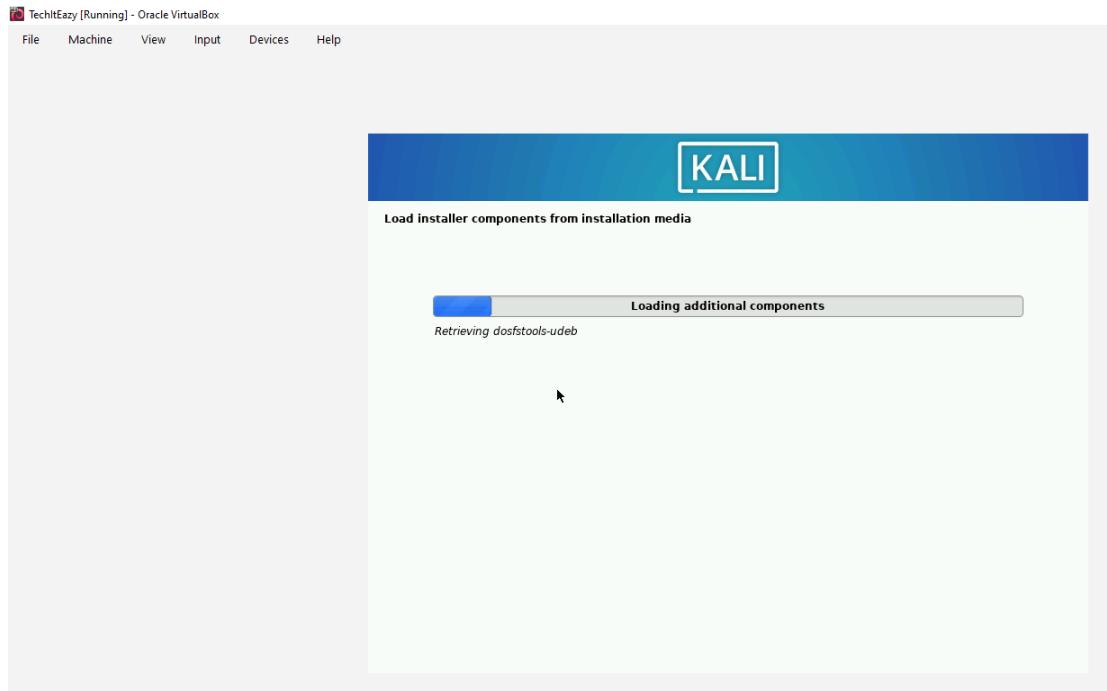
Kali Linux configuration

We used VirtualBox to install the kali Linux 2025.3 operating system

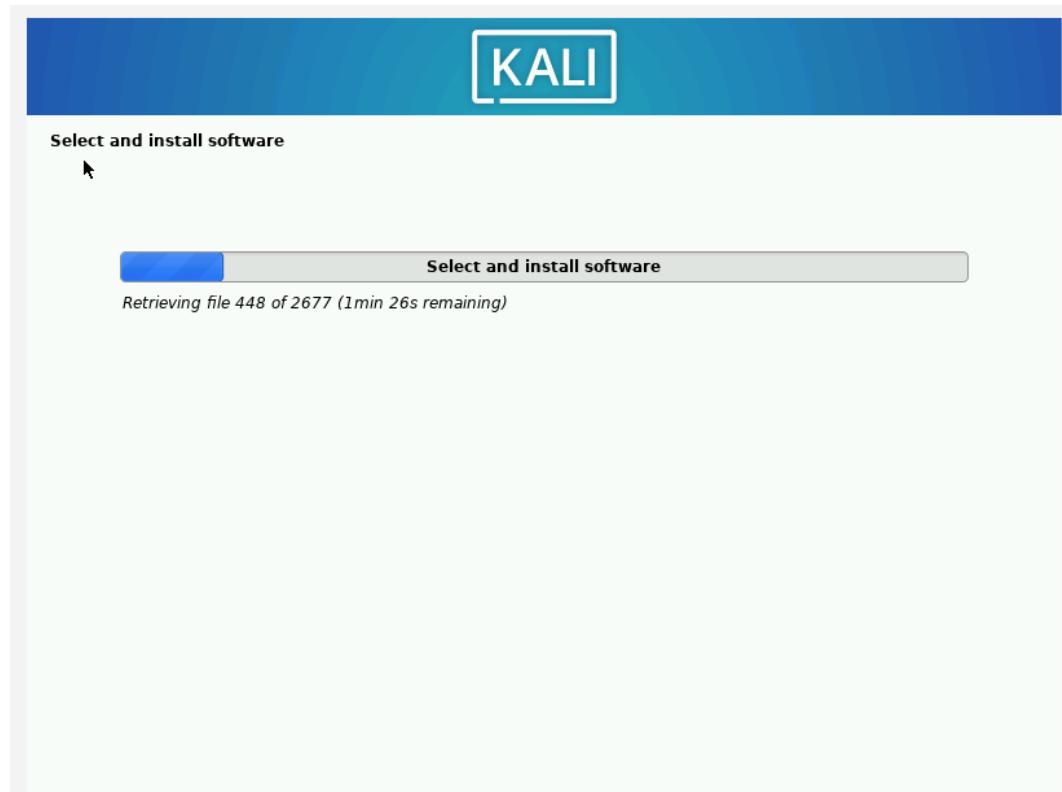


Os image installation

installing the OS using the graphical user interface

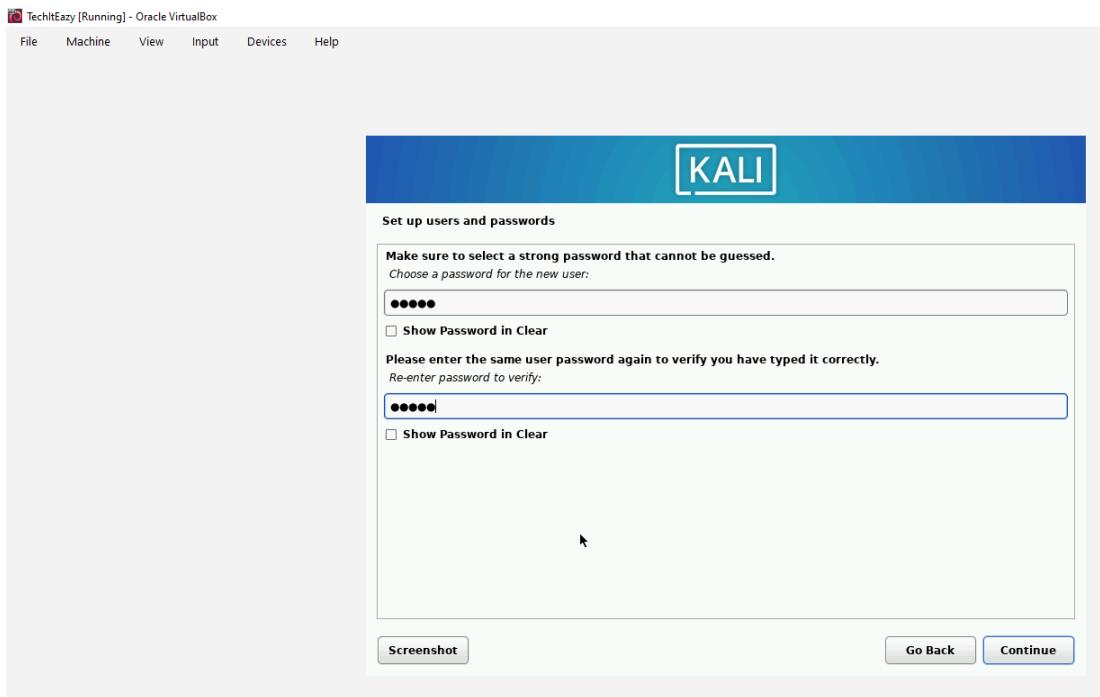


Installing additional system components and software packages



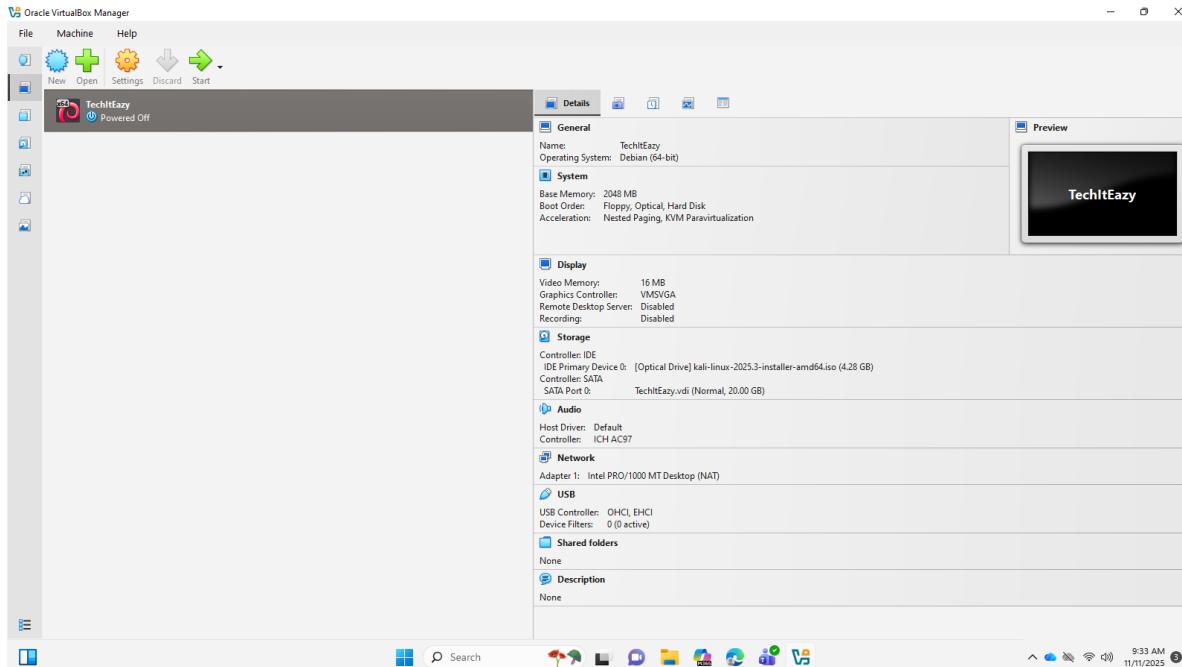
password

Setting up password for the root user which is going to have sudo privileges.



Virtual machine

Our operating system has been installed and ready for use.



Desktop

Using kali Linux 2025.3

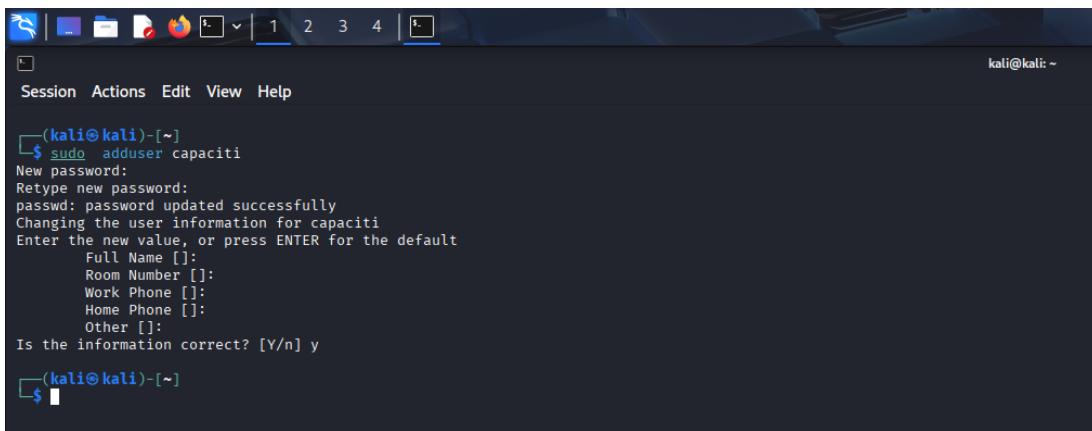


Creating Sudo user

The following process is executed from an administrative account with that has pre-existing Sudo privileges the name of the account is kali.

To create the user account, we used the following command **sudo adduser capaciti** this command creates a new user account named capaciti in the system. The adduser command is a high-level utility that does the following:

- Creates the user in the /etc/passwd and /etc/shadow databases.
- Prompts you to set a secure password for the account.
- Creates a new home directory (/home/newusername).
- Copies default configuration files into the new home directory.



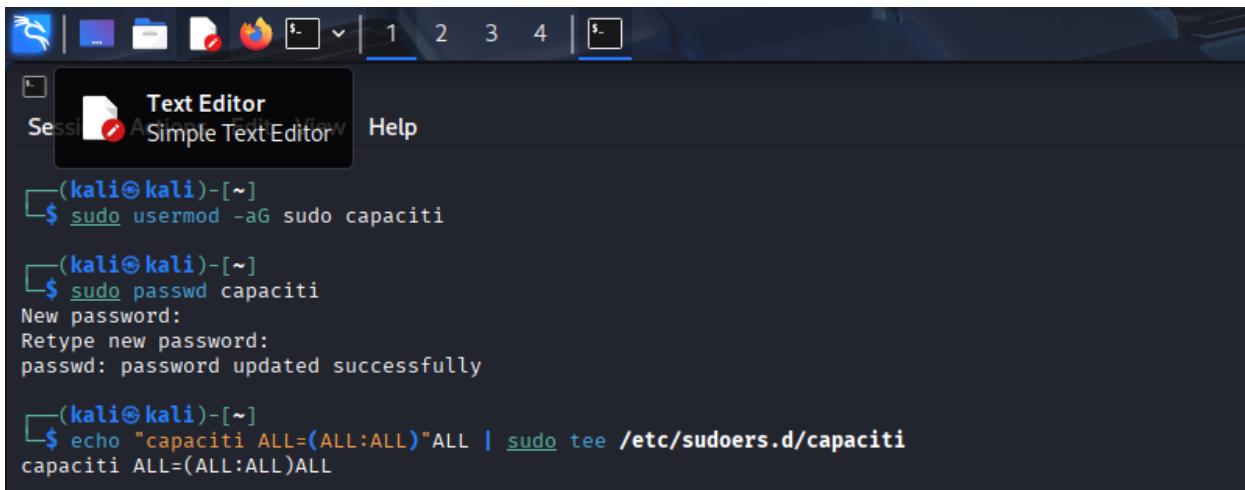
A screenshot of a terminal window on a Kali Linux desktop. The window title bar shows the Kali logo and the text 'Session Actions Edit View Help'. The terminal prompt is '(kali㉿kali)-[~]'. The user runs the command '\$ sudo adduser capaciti'. The terminal then prompts for a new password, which is retyped. It shows that the password was updated successfully and asks for additional user information like full name, room number, work phone, home phone, and other details. The user enters their information and then confirms if it is correct by typing 'y' at the prompt 'Is the information correct? [Y/n]'. The terminal ends with the prompt '\$'.

```
(kali㉿kali)-[~]
$ sudo adduser capaciti
New password:
Retype new password:
password updated successfully
Changing the user information for capaciti
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
(kali㉿kali)-[~]
$
```

Grant Administrative Privileges

We then went on to grant administrative permissions (sudo permissions) to the capaciti user using the following command **sudo usermod -aG capaciti** this command modifies the user's group memberships to grant them administrative rights. The **sudo usermod -aG command** does the following:

- **usermod**: The command for modifying an existing user account.
- **-aG**: The flags -a (append) and -G (groups) are used together to add the user to a supplementary group without removing them from their existing groups.
- **sudo**: This is the target group. Members of the sudo group are permitted to execute commands with elevated privileges.



The screenshot shows a terminal window titled "Text Editor" with the session name "Simple Text Editor". The terminal is running on a Kali Linux system. The user has run the command `sudo usermod -aG sudo capaciti`. They then run `sudo passwd capaciti` and provide a new password. The command `echo "capaciti ALL=(ALL:ALL)ALL" | sudo tee /etc/sudoers.d/capaciti` is used to add the user to the sudo group in the sudoers file.

```
(kali㉿kali)-[~]
$ sudo usermod -aG sudo capaciti

(kali㉿kali)-[~]
$ sudo passwd capaciti
New password:
Retype new password:
passwd: password updated successfully

(kali㉿kali)-[~]
$ echo "capaciti ALL=(ALL:ALL)ALL" | sudo tee /etc/sudoers.d/capaciti
capaciti ALL=(ALL:ALL)ALL
```

Verify Group Membership

Using the **groups capaciti** this command provides a verification step. It displays all groups to which the specified user belongs. A successful output will confirm the user is a member of both their primary group and the sudo group.



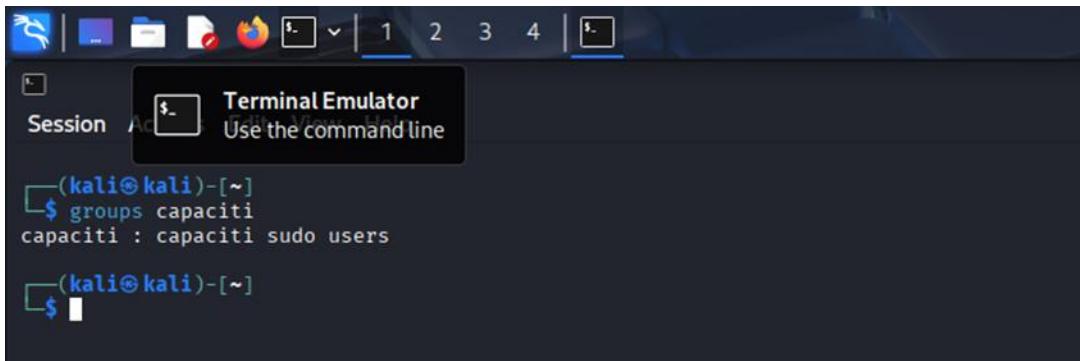
The screenshot shows a terminal window titled "Firefox ESR" with the session name "Browse the World Wide Web". The user runs `su - capaciti` to switch to the capaciti user. They then run `sudo whoami`, which returns "root", indicating successful sudo access.

```
(kali㉿kali)-[~]
$ su - capaciti
Password:
(capaciti㉿kali)-[~]
$ sudo whoami
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)
^

root
```

Validate Sudo Access

- Using the **su – capaciti** and the **sudo whoami** this two commands have the following purpose : **su - newusername**: This command switches the active session to the new user, providing a login shell with that user's environment.
- **sudo whoami**: When executed by the new user, this command tests the sudo configuration. If successful, it will return root, proving the user can execute commands with elevated privileges.



Session Terminal Emulator Use the command line

```
(kali㉿kali)-[~]
$ groups capaciti
capaciti : capaciti sudo users

(kali㉿kali)-[~]
$
```

Testing user permissions

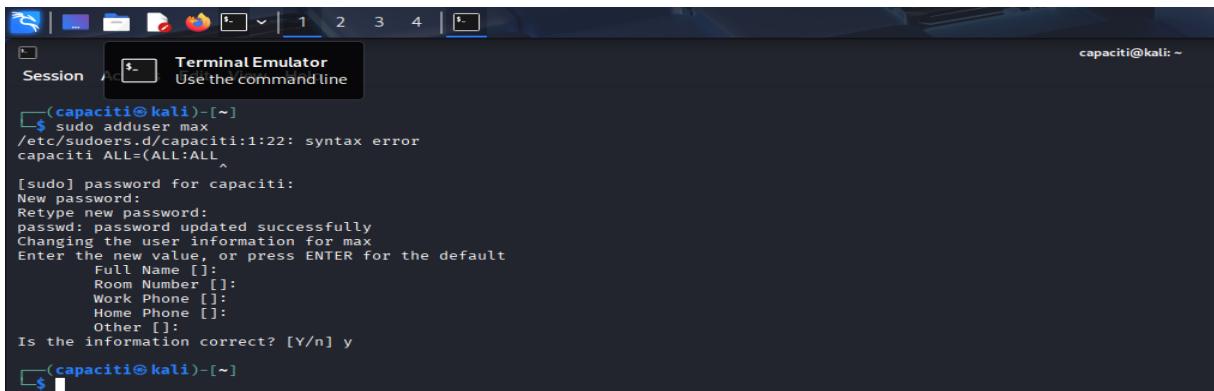
Testing if the sudo privileges are working using the **sudo apt update** command

```
(capaciti㉿kali)-[~]
$ sudo apt update
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Ign:1 http://http.kali.org/kali kali-rolling InRelease
Err:1 http://http.kali.org/kali kali-rolling InRelease
  Temporary failure resolving 'http.kali.org'
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
Warning: Failed to fetch http://http.kali.org/kali/dists/kali-rolling/InRelease  Temporary failure resolving 'http.
kali.org'
Warning: Some index files failed to download. They have been ignored, or old ones used instead.

(capaciti㉿kali)-[~]
$
```

Creating user without root permission

We then created a user that doesn't have sudo privileges this user cant use the sudo command its not given permissions or right to use such high level command



Session Terminal Emulator Use the command line

```
capaciti@kali: ~

(capaciti㉿kali)-[~]
$ sudo adduser max
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)
[sudo] password for capaciti:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for max
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []
Is the information correct? [Y/n] y

(capaciti㉿kali)-[~]
$
```

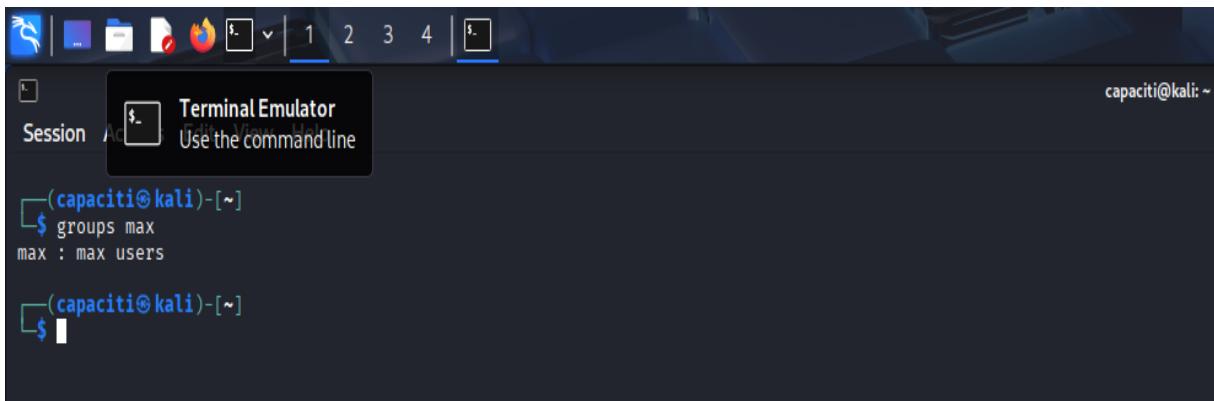
The purpose of the sudo adduser max command creates a new, standard user account, this standard user account doesn't have any sudo permissions.

The sudo adduser does the following:

- Add the user to the system authentication databases (/etc/passwd, /etc/shadow).
- Prompt for the creation of a secure password for the account.
- Generate a dedicated home directory (/home/max).
- Set up a default user environment with essential configuration files.

Verify Account Creation and Privileges

We then used the **groups max** command which confirms the user's group memberships. For a standard user, the output should show the user is only a member of their primary group (max : max). The absence of the sudo group in the output confirms the account has no inherent administrative rights.

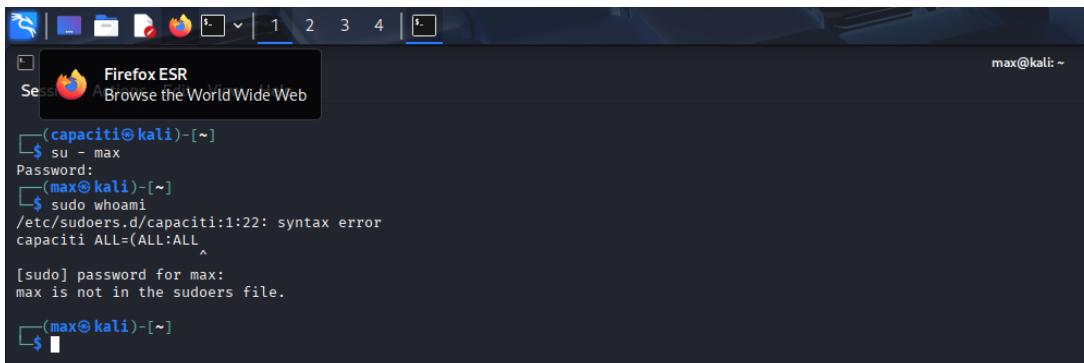


```
(capaciti㉿kali)-[~]
$ groups max
max : max users

(capaciti㉿kali)-[~]
$
```

Test the Account

We then switched to the max user account and attempted to run a sudo command which then gave us an error.



```
(capaciti㉿kali)-[~]
$ su - max
Password:
(max㉿kali)-[~]
$ sudo whoami
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)
[sudo] password for max:
max is not in the sudoers file.

(max㉿kali)-[~]
$
```

This is the expected behaviour, confirming the user cannot execute privileged commands.

Conclusion and Key Findings

Throughout this process, we have systematically addressed user account management and privilege delegation in a Kali Linux environment. The procedure began by successfully creating a privileged user account named 'capaciti,' which involved using the adduser command for account creation, followed by the usermod utility to grant administrative rights through membership in the 'sudo' group. This was validated using the groups command to confirm group membership and tested by verifying sudo command execution. We further contrasted this by outlining the simpler workflow for creating a standard user account without sudo privileges, emphasizing the security principle of least privilege. The key learning reinforces that user creation and privilege assignment must be performed from an existing administrative account, and that proper verification steps are critical to ensure the intended access levels are correctly configured. This methodology provides a clear framework for managing user permissions while maintaining system security integrity.

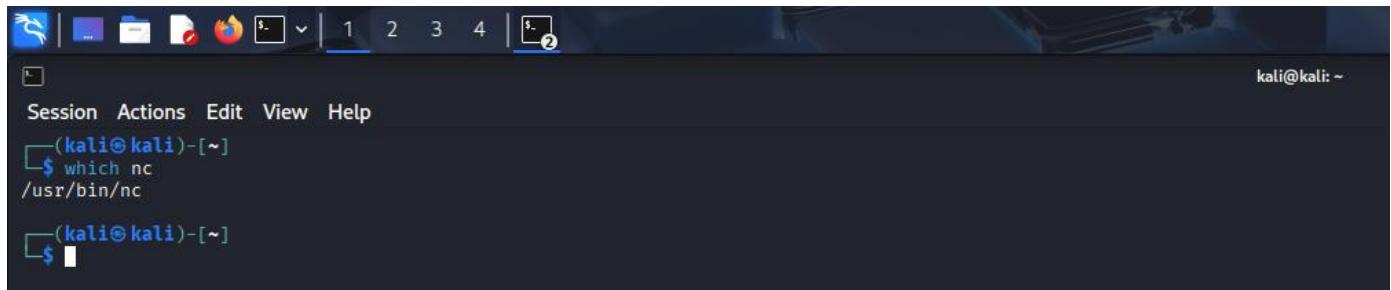
Networking

Virtual network

Netcat to send messages

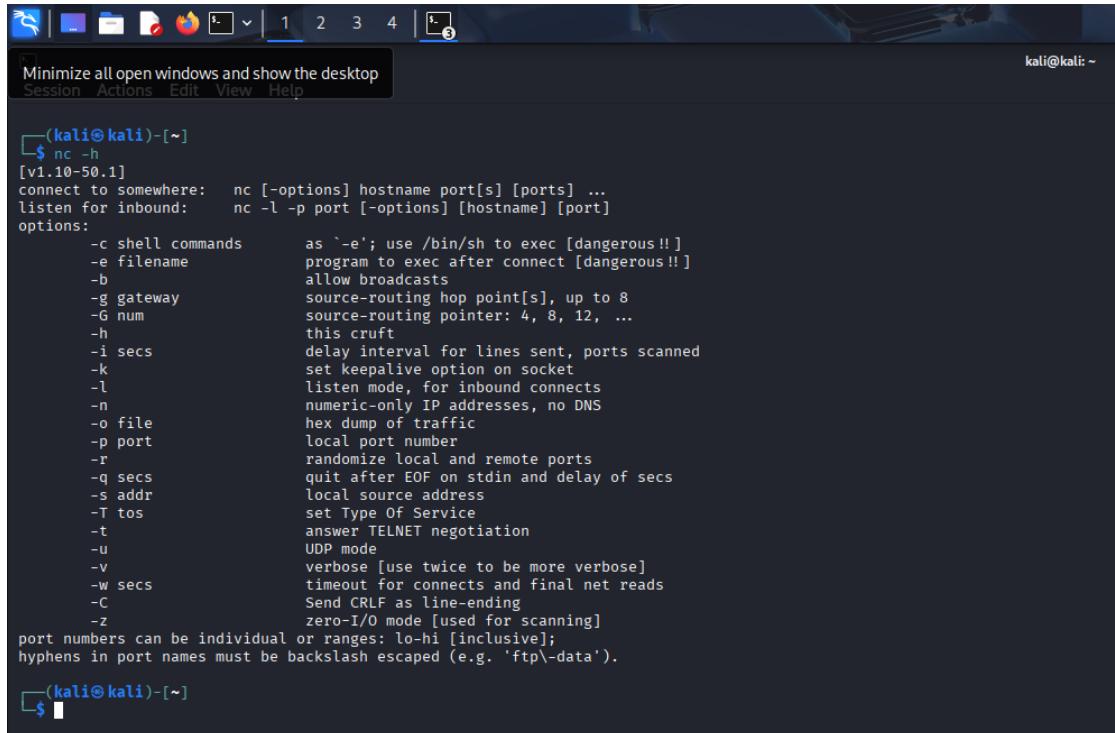
We want to make the two virtual machines to communicate so we installed Netcat which we were using to setup communication between the two computers.

We used the following command to verify if Netcat is installed **why nc**, the results were positive which confirmed that Netcat is installed.



```
kali@kali: ~
Session Actions Edit View Help
└─(kali㉿kali)-[~]
  $ which nc
/usr/bin/nc
└─(kali㉿kali)-[~]
  $
```

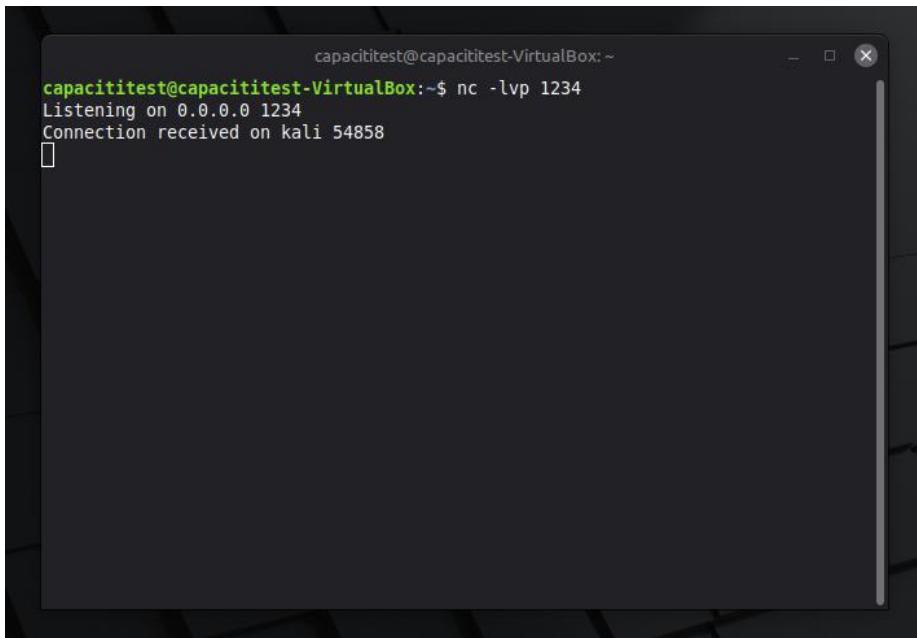
We then used the following command **nc -h** to confirm if Netcat is installed and working



```
Minimize all open windows and show the desktop
Session Actions Edit View Help
└─(kali㉿kali)-[~]
  $ nc -h
[v1.10-50.1]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound: nc -l -p port [-options] [hostname] [port]
options:
  -c shell commands      as '-e'; use /bin/sh to exec [dangerous!!]
  -e filename            program to exec after connect [dangerous!!]
  -b                    allow broadcasts
  -g gateway             source-routing hop point[s], up to 8
  -G num                source-routing pointer: 4, 8, 12, ...
  -h                    this crust
  -i secs               delay interval for lines sent, ports scanned
  -k                    set keepalive option on socket
  -l                    listen mode, for inbound connects
  -n                    numeric-only IP addresses, no DNS
  -o file               hex dump of traffic
  -p port               local port number
  -r                    randomize local and remote ports
  -q secs               quit after EOF on stdin and delay of secs
  -s addr               local source address
  -T tos                set Type Of Service
  -t                    answer TELNET negotiation
  -u                    UDP mode
  -v                    verbose [use twice to be more verbose]
  -w secs               timeout for connects and final net reads
  -C                  Send CRLF as line-ending
  -z                  zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\\-data').
└─(kali㉿kali)-[~]
  $
```

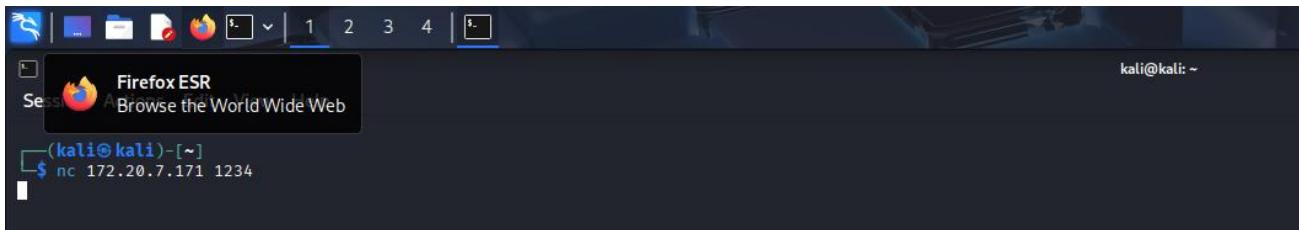
The results confirmed that Netcat is working.

We then used the **nc -lvp 1234** on Linux mint using Netcat to start a server that listens for incoming connections on port 1234.



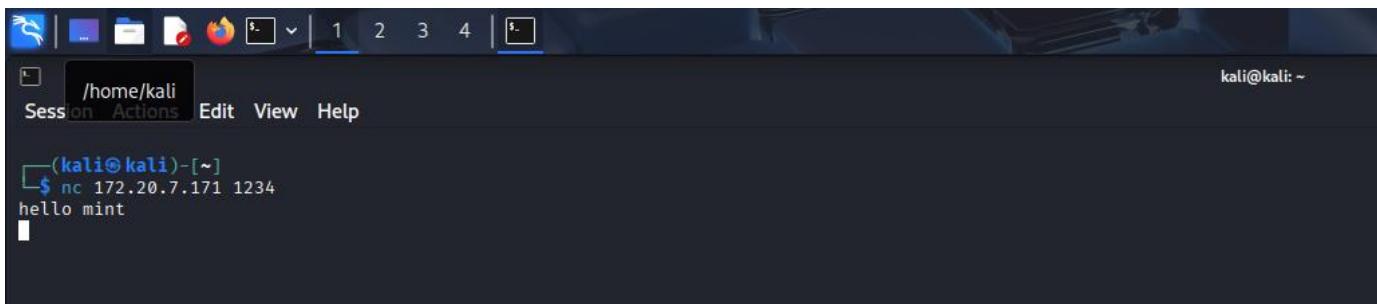
```
capacititest@capacititest-VirtualBox:~$ nc -lvp 1234
Listening on 0.0.0.0 1234
Connection received on kali 54858
```

Communicating with Linux mint VM using the kali Linux virtual machine which was successful



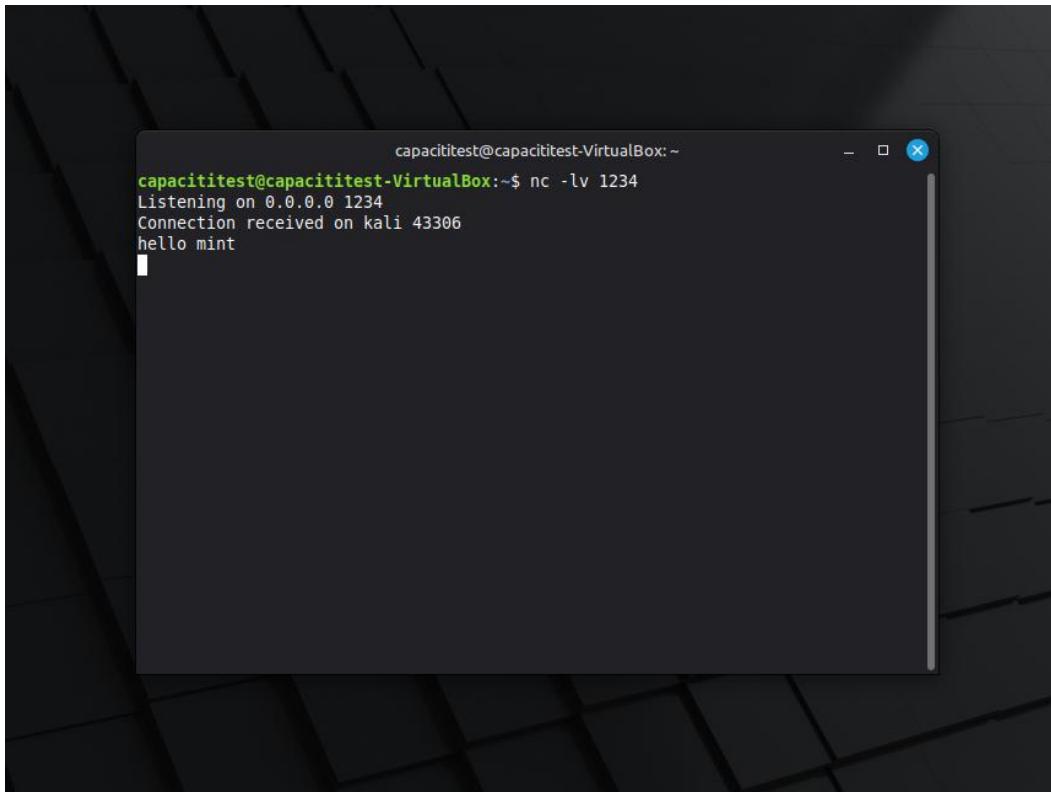
```
kali@kali:~$ nc 172.20.7.171 1234
```

We then sent a message to the Linux Mint virtual machine using the kali virtual mint



```
/home/kali Session Actions Edit View Help
kali@kali:~$ nc 172.20.7.171 1234
hello mint
```

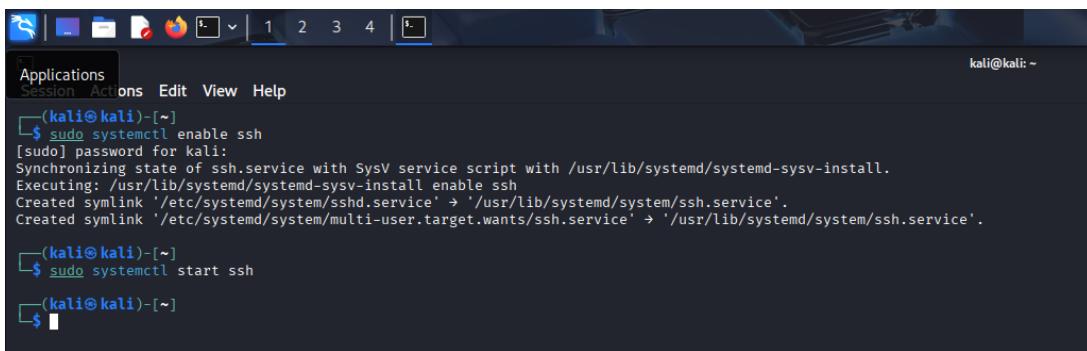
The message was successfully received by the Linux mint virtual machine.



```
capacititest@capacititest-VirtualBox:~$ nc -l 1234
Listening on 0.0.0.0 1234
Connection received on kali 43306
hello mint
```

File transfer between two virtual machines

We sent a file from the kali virtual machine to the Linux mint virtual machine using ssh the name of the file is test. We used the following command to enable ssh on the virtual machine **sudo systemctl enable ssh** which enabled ssh and followed by the **sudo systemctl start ssh** which starts the ssh. On the Linux mint virtual machine, we also enabled the ssh using the following command sudo systemctl enable ssh



```
kali@kali:~$ sudo systemctl enable ssh
[sudo] password for kali:
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'.

kali@kali:~$ sudo systemctl start ssh
```

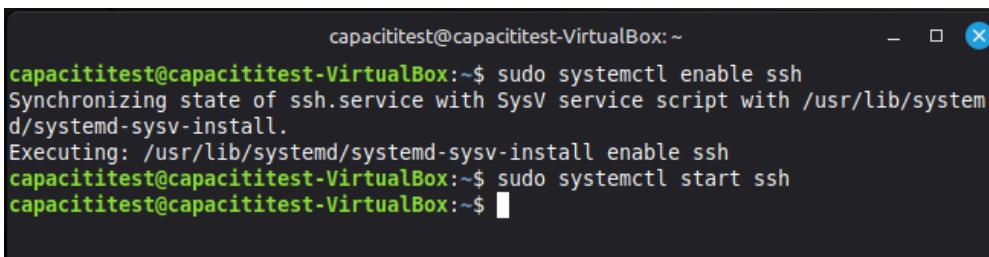
To transfer the file from the kali VM to the Linux mint VM we opened the terminal and navigated it to the downloads file using the following command **cd /home/kali/Downloads** and then we went on to start the python http server using the following command **python3 -m http.server 8000** which will be used to serve the files between kali and mint using http port number 8000.



A screenshot of a Kali Linux terminal window. The title bar says "Terminal Emulator". The window shows a session titled "Session 1" with the command line "Use the command line". The terminal output is as follows:

```
(kali㉿kali)-[~]
$ cd /home/kali/Downloads
(kali㉿kali)-[~/Downloads]
$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.56.102 - - [12/Nov/2025 09:45:06] "GET /test HTTP/1.1" 200 -
```

On the Linux mint terminal, we typed the following command **wget <http://192.168.56.101:8000/test>** this command tells your Mint VM to download a file named test from the Kali VM using HTTP. Kali is running a Python-based web server on port 8000, serving files from its Downloads folder. When Mint accesses that URL, it retrieves the file and saves it locally, then followed by the **ls -l test** command which we used to verify that the test file has been successfully sent.



A screenshot of a Linux Mint terminal window. The title bar says "capacititest@capacititest-VirtualBox: ~". The terminal output is as follows:

```
capacititest@capacititest-VirtualBox:~$ 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
capacititest@capacititest-VirtualBox:~$ sudo systemctl start ssh
capacititest@capacititest-VirtualBox:~$
```

```
capacititest@capacititest-VirtualBox:~$ wget http://192.168.56.101:8000/test
--2025-11-12 09:45:05-- http://192.168.56.101:8000/test
Connecting to 192.168.56.101:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [application/octet-stream]
Saving to: 'test'

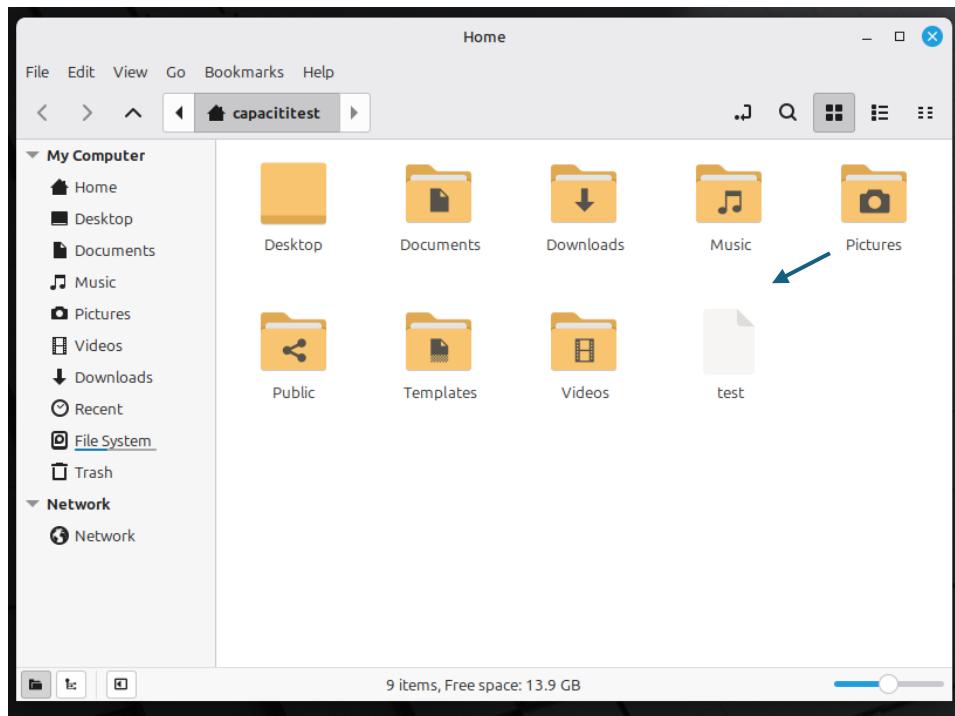
test [ <=> ] 0 --.-KB/s in 0s

2025-11-12 09:45:05 (0.00 B/s) - 'test' saved [0/0]

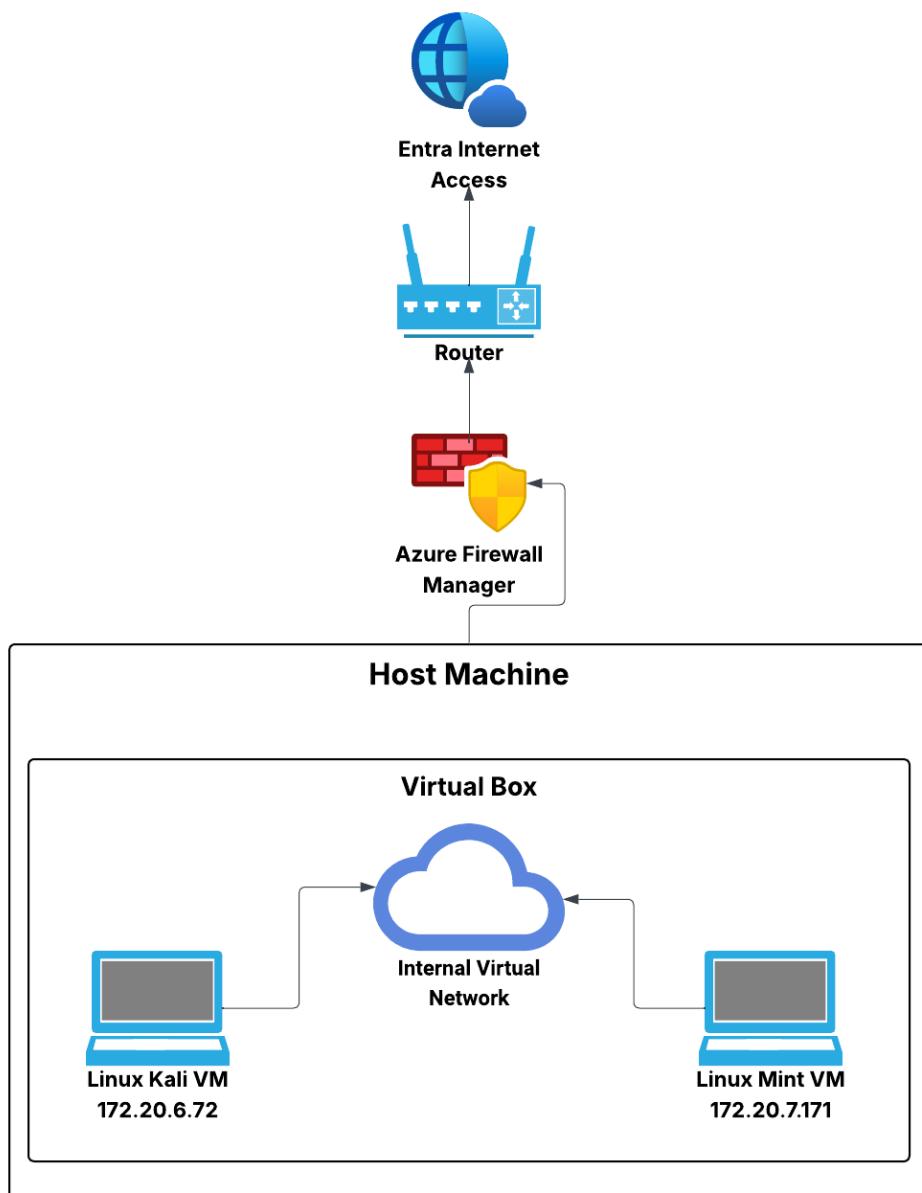
capacititest@capacititest-VirtualBox:~$ ls -l test
-rw-rw-r-- 1 capacititest capacititest 0 Nov 12 09:31 test
capacititest@capacititest-VirtualBox:~$
```

machine.

File has been successfully transferred to the mint virtual machine.



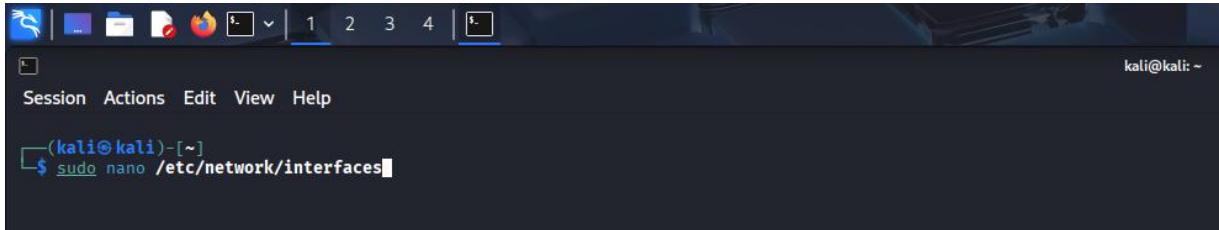
Network diagram



Basic local network setup

Assigning IP by editing interface file

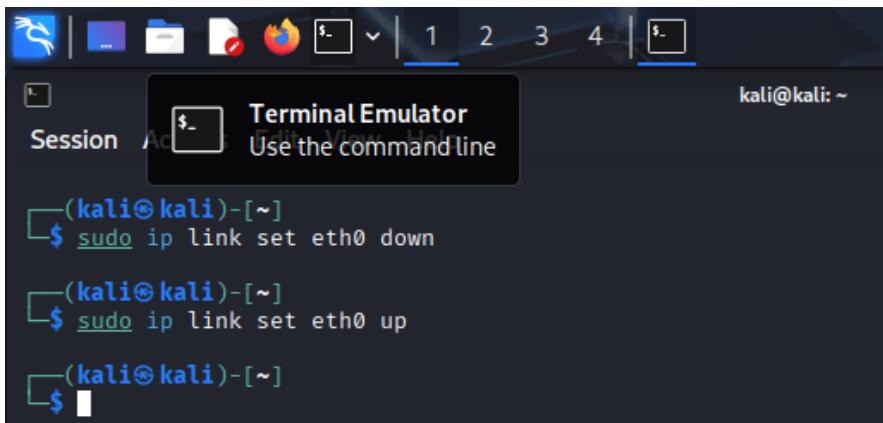
Using the **sudo nano /etc/network/interface** we opened the file and elevated the privileges



The command opens the Linux open-source GNU nano text editor where we are going to edit the interface document which will be used to manually change the IP address and net mask and then gateway.

A screenshot of a terminal window titled 'Terminal Emulator - Use the command line'. The window shows the contents of the '/etc/network/interfaces' file. The file contains configuration for the 'lo' and 'eth0' interfaces, including static IP settings. The terminal is running on a Kali Linux desktop environment.

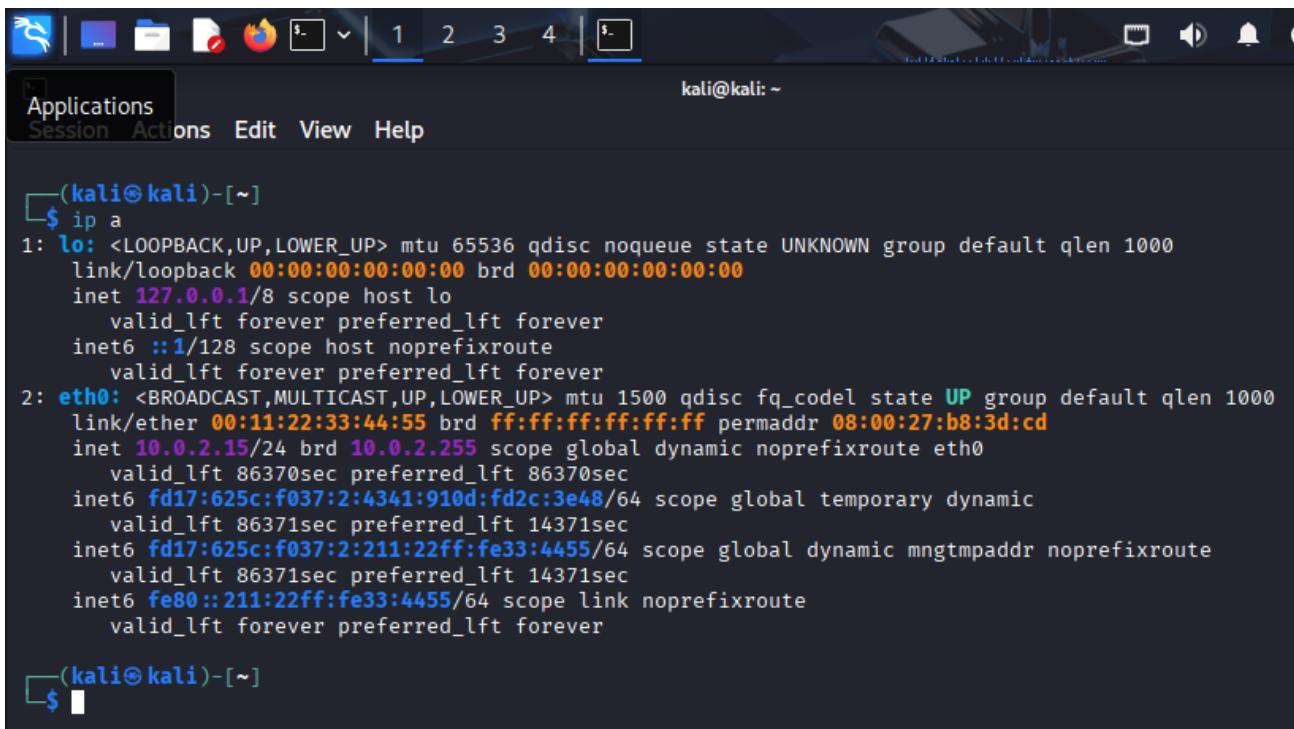
We used the following command Running **sudo ip link set eth0 down** followed by **sudo ip link set eth0 up** temporarily disables and then re-enables the eth0 network interface, effectively resetting it. This is a quick way to refresh the connection, apply new settings, or force a DHCP renewal without rebooting the system.



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "Terminal Emulator" and the subtitle is "Use the command line". The terminal content shows the user running commands to change the state of the eth0 interface:

```
(kali㉿kali)-[~]
$ sudo ip link set eth0 down
(kali㉿kali)-[~]
$ sudo ip link set eth0 up
(kali㉿kali)-[~]
```

We then went forward to use the **Ip a** command to check if the Ip address had changed and the ip address had successfully changed.



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is "Terminal Emulator" and the subtitle is "Use the command line". The terminal content shows the user running the **ip a** command to check the network interfaces:

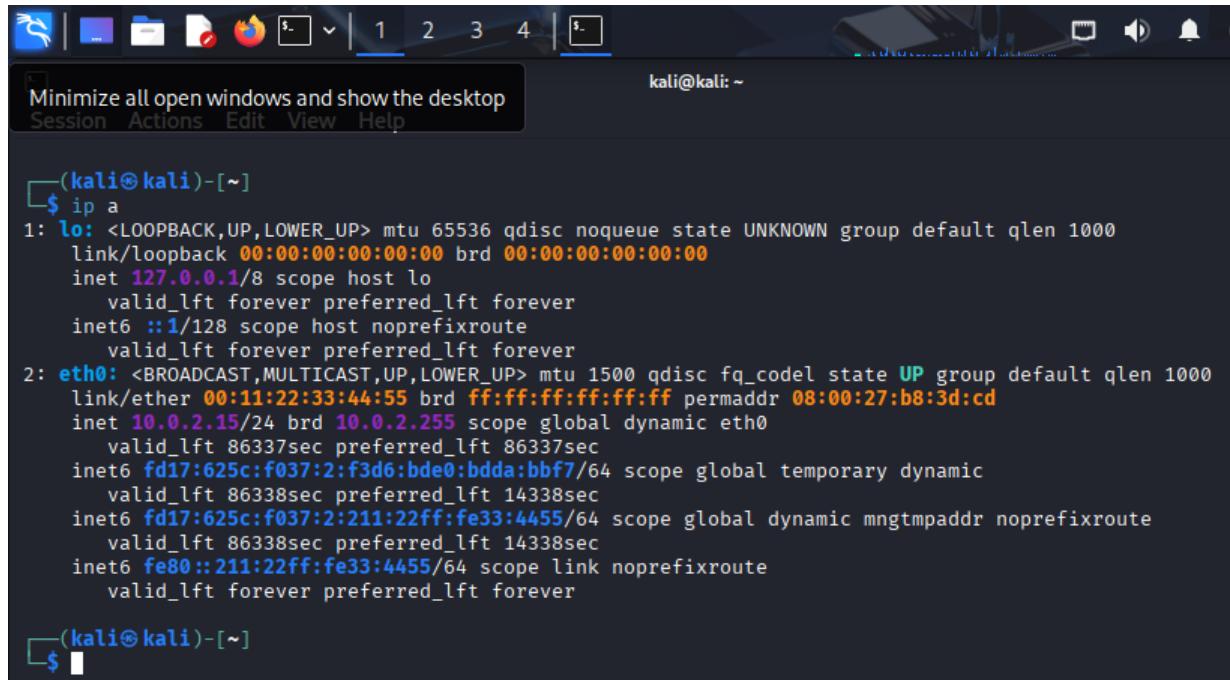
```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host 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 00:11:22:33:44:55 brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:b8:3d:cd
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
            valid_lft 86370sec preferred_lft 86370sec
            inet6 fd17:625c:f037:2:4341:910d:fd2c:3e48/64 scope global temporary dynamic
                valid_lft 86371sec preferred_lft 14371sec
            inet6 fd17:625c:f037:2:211:22ff:fe33:4455/64 scope global dynamic mngtmpaddr noprefixroute
                valid_lft 86371sec preferred_lft 14371sec
            inet6 fe80::211:22ff:fe33:4455/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
```

Assign IPs manually or using DHCP

Mac address spoofing

To assign IP address on the kali Linux we use the spoofing method to manually force a change of the current Ip address. Mac address spoofing is when we temporarily change the unique hardware identifier assigned to your network interface, which tricks the DHCP server in this case our router into treating your device as a new client. This helped force a new IP address because on our previous attempts the server keeps reassigning the same one based on our original MAC.

We started by updating the system packages our IP address before spoofing ways 10.0.2.15/24.



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal title is '(kali㉿kali)-[~]'. The user has run the command 'ip a' to list network interfaces. The output shows two interfaces: 'lo' (loopback) and 'eth0' (ethernet). The 'lo' interface has an IP of 127.0.0.1/8. The 'eth0' interface has an IP of 10.0.2.15/24. The terminal window also includes a menu bar with options like Minimize all open windows and show the desktop, Session, Actions, Edit, View, Help, and a status bar showing 'kali@kali: ~'.

```
(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 00:11:22:33:44:55 brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:b8:3d:cd
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
            valid_lft 86337sec preferred_lft 86337sec
        inet6 fd17:625c:f037:2:f3d6:bde0:bbf7/64 scope global temporary dynamic
            valid_lft 86338sec preferred_lft 14338sec
        inet6 fd17:625c:f037:2:211:22ff:fe33:4455/64 scope global dynamic mngtmpaddr noprefixroute
            valid_lft 86338sec preferred_lft 14338sec
        inet6 fe80::211:22ff:fe33:4455/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

The sequence of commands starts by bringing the interface down (Ip link set dev eth0 down), then assigns a new MAC address (Ip link set dev eth0 address 00:11:22:33:44:55), and finally brings the interface back up (Ip link set dev eth0 up). Once the new MAC is active, dhclient -v eth0 requests a fresh IP lease from the DHCP server, often resulting in a different IP address than before. This method is especially useful in environments like Kali Linux where traditional network management tools may be unavailable.

```
(kali㉿kali)-[~]
$ sudo ip link set dev eth0 down
(kali㉿kali)-[~]
$ sudo ip link set dev eth0 address 00:11:22:33:44:55
(kali㉿kali)-[~]
$ sudo ip link set dev eth0
(kali㉿kali)-[~]
$ sudo dhclient -v eth0
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth0/00:11:22:33:44:55
Sending on LPF/eth0/00:11:22:33:44:55
Sending on Socket/fallback
DHCPREQUEST for 10.0.2.16 on eth0 to 255.255.255.255 port 67
DHCPNAK from 10.0.2.2
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 5
DHCPOffer of 10.0.2.15 from 10.0.2.2
DHCPREQUEST for 10.0.2.15 on eth0 to 255.255.255.255 port 67
DHCPACK of 10.0.2.15 from 10.0.2.2
bound to 10.0.2.15 -- renewal in 41331 seconds.

(kali㉿kali)-[~]
```

After successfully spoofing the mac address we check our ip address if it had changed and it had changed to 10.0.2.16/24.

```
(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 00:11:22:33:44:55 brd ff:ff:ff:ff:ff:ff permaddr 08:00:27:b8:3d:cd
    inet 10.0.2.16/24 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 86199sec preferred_lft 86199sec
    inet6 fd17:625c:f037:2:c5ce:7982:22f:179d/64 scope global temporary dynamic
        valid_lft 86201sec preferred_lft 14201sec
    inet6 fd17:625c:f037:2:211:22ff:fe33:4455/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 86201sec preferred_lft 14201sec
    inet6 fe80::211:22ff:fe33:4455/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

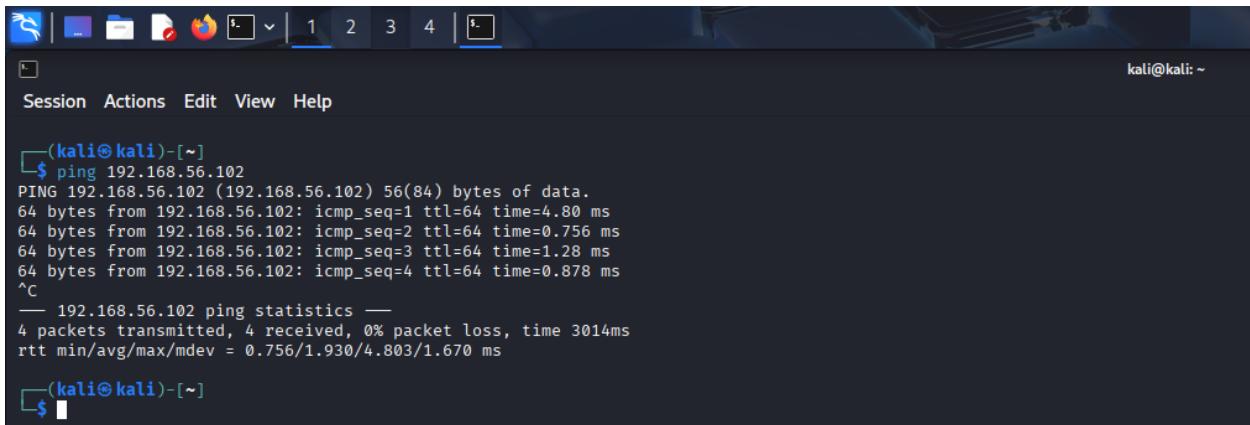
(kali㉿kali)-[~]
```

This method is useful in situations where you don't have other kali utilities available.

Basic configuration

Ping test

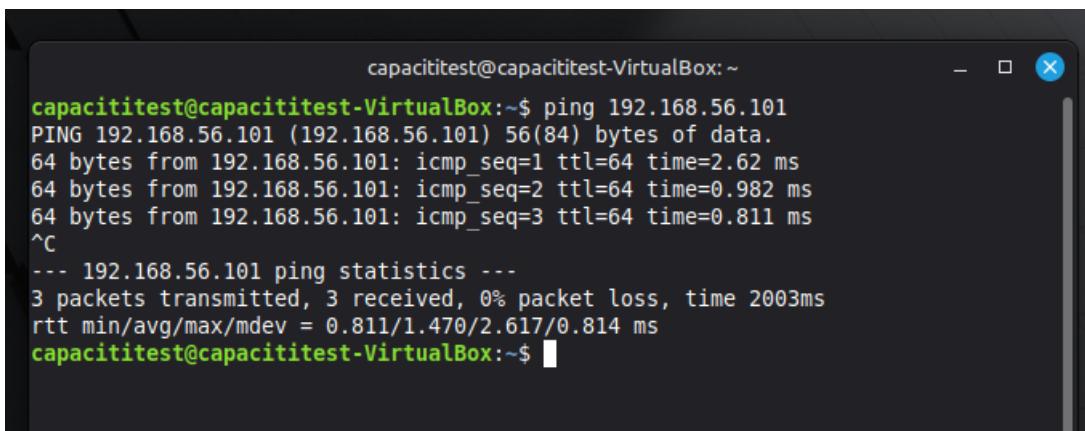
We used the kali Linux virtual machine to ping the Linux Mint virtual machine the Kali Linux virtual machine has the following Ip address 192.168.56.102.



```
(kali㉿kali)-[~]
$ ping 192.168.56.102
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data.
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=4.80 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=0.756 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=1.28 ms
64 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=0.878 ms
^C
--- 192.168.56.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3014ms
rtt min/avg/max/mdev = 0.756/1.930/4.803/1.670 ms

(kali㉿kali)-[~]
$
```

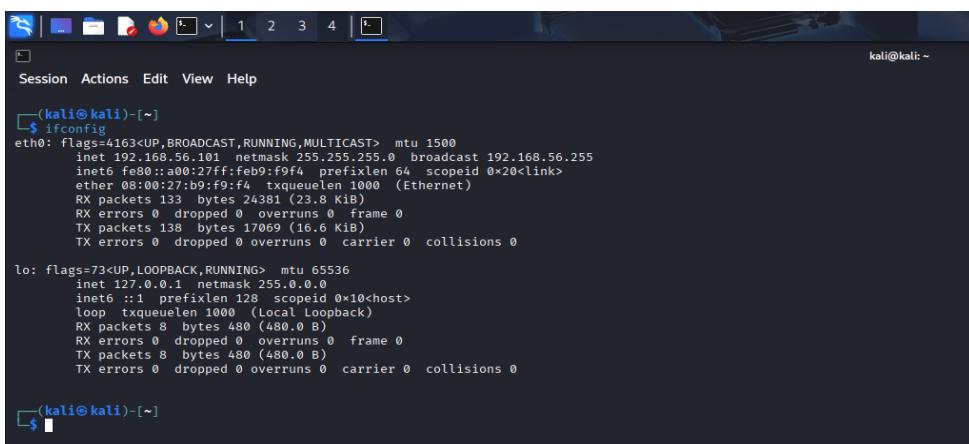
We then went on to ping the Linux mint virtual machine.



```
capacititest@capacititest-VirtualBox: ~
capacititest@capacititest-VirtualBox:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=2.62 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.982 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.811 ms
^C
--- 192.168.56.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.811/1.470/2.617/0.814 ms
capacititest@capacititest-VirtualBox:~$
```

ifconfig

We used the ipconfig command to get the IP address of the and the Linux.

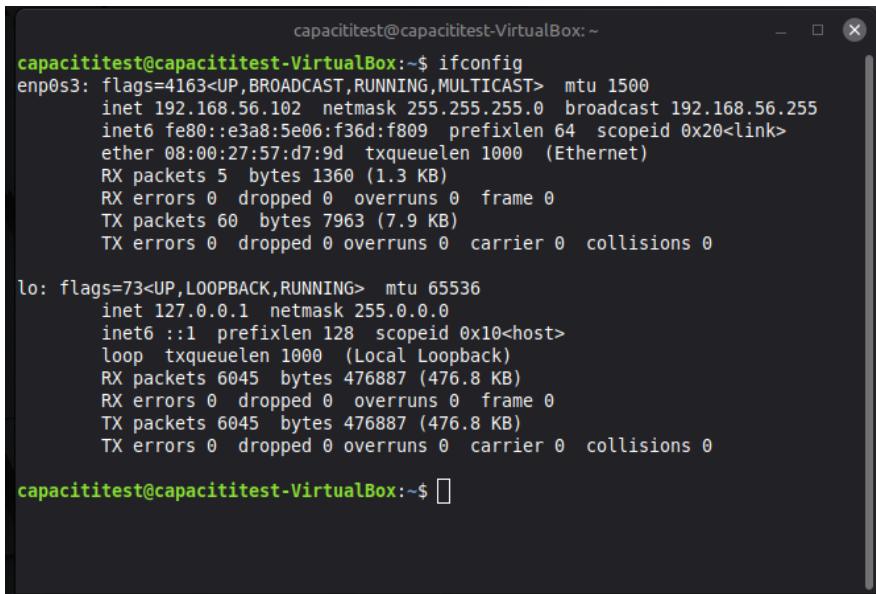


```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 192.168.56.101  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::a00:27ff:feb0:f9f4  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:b0:f9:f4  txqueuelen 1000  (Ethernet)
            RX packets 133  bytes 24381 (23.8 Kib)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 138  bytes 17069 (16.6 Kib)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
          loop  txqueuelen 1000  (Local Loopback)
            RX packets 8  bytes 480 (480.0 B)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 8  bytes 480 (480.0 B)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

(kali㉿kali)-[~]
$
```

The ifconfig command in Kali Linux is used to display and configure network interfaces. It shows important details like IP address, MAC address, netmask, and interface status (up or down). You can use it to manually assign IP addresses, enable or disable interfaces, and troubleshoot network connectivity.



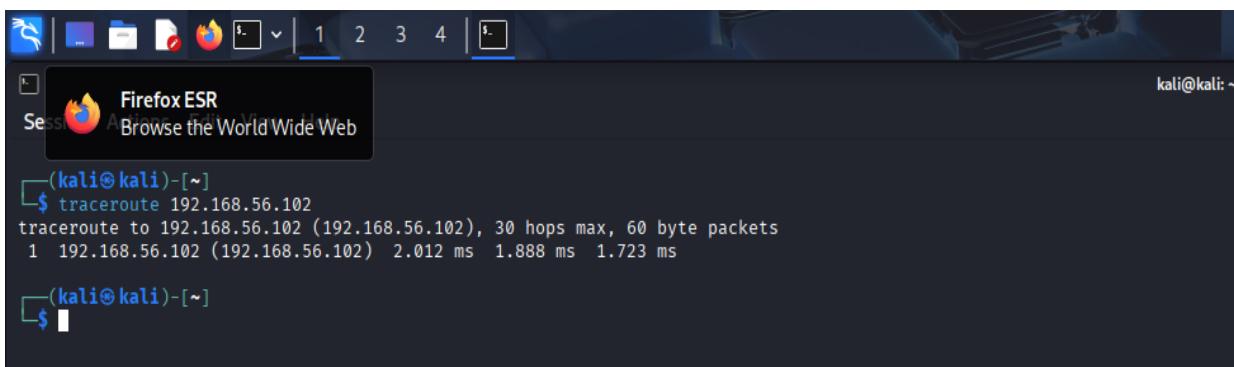
```
capacititest@capacititest-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
        inet6 fe80::e3a8:5e06:f36d:f809 prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:57:d7:9d txqueuelen 1000 (Ethernet)
            RX packets 5 bytes 1360 (1.3 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 60 bytes 7963 (7.9 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 6045 bytes 476887 (476.8 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 6045 bytes 476887 (476.8 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

capacititest@capacititest-VirtualBox:~$
```

Traceroute

We then used traceroute on the kali virtual machine, traceroute is a network diagnostic tool in Kali Linux that shows the path your data packets take to reach a specific destination. It lists each hop (router or gateway) along the route and measures how long it takes to reach each one. This helps identify where delays, failures, or bottlenecks occur in the network.



```
kali@kali:~$ traceroute 192.168.56.102
traceroute to 192.168.56.102 (192.168.56.102), 30 hops max, 60 byte packets
 1  192.168.56.102 (192.168.56.102)  2.012 ms  1.888 ms  1.723 ms

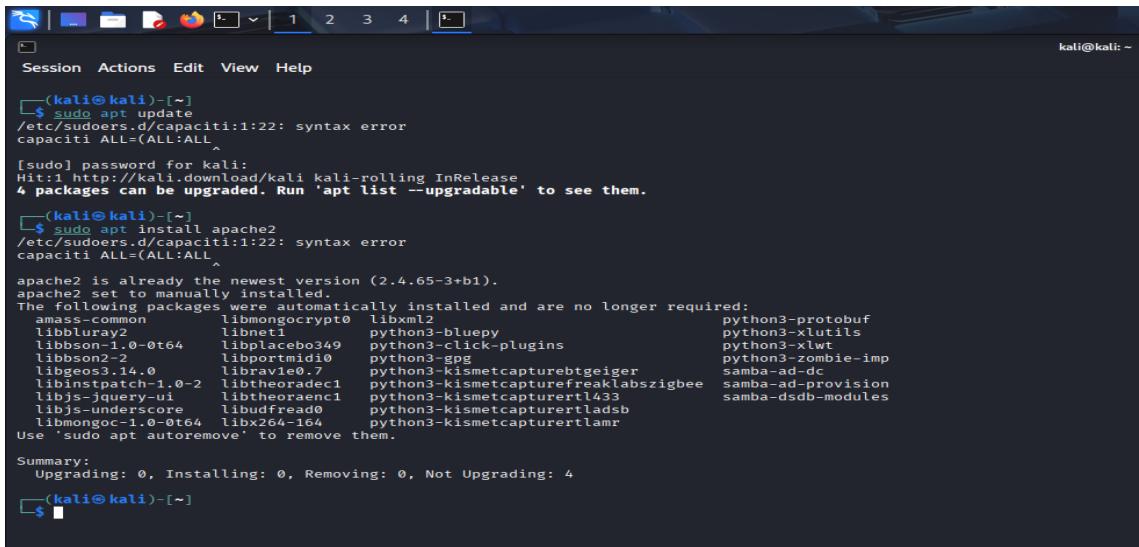
kali@kali:~$
```

Server & Services

Server and layout

Installing the Apache server

We used the following commands **sudo apt update** which updates the software packages we then went on to use the **sudo apt install apache2** which installs Apache



```
(kali㉿kali)-[~]
$ sudo apt update
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)
[sudo] password for kali:
Hit:1 http://kali.download/kali kali-rolling InRelease
4 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali㉿kali)-[~]
$ sudo apt install apache2
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)

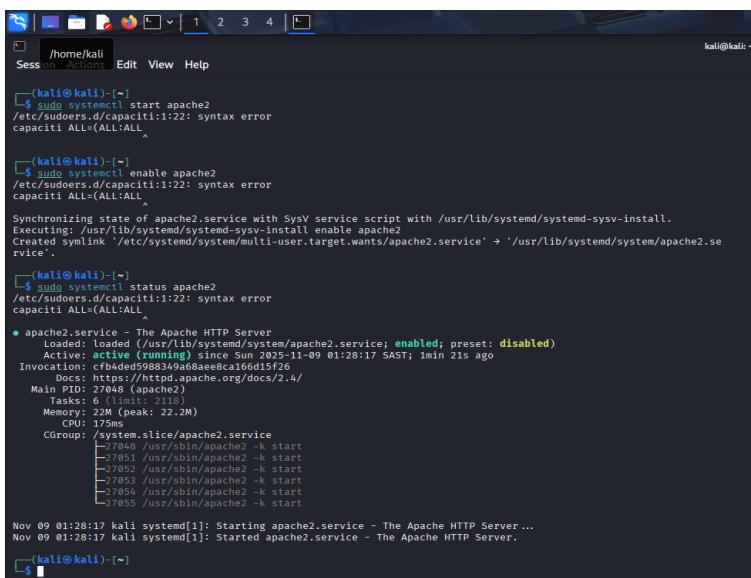
apache2 is already the newest version (2.4.65-3+b1).
apache2 set to manually installed.
The following packages were automatically installed and are no longer required:
amass-common libmongocrypt0 libxml2 python3-protoBuf
liblbburay2 libnet1 python3-bluepy python3-xlutils
libbison-1.0-64 libplacebo349 python3-click-plugins python3-xlw
libbison2-2 libportmid10 python3-gpg python3-zombie-imp
libgeos3-14_0 libravie0.7 python3-kismetcapturebtgeiger samba-ad-dc
libinstalld-1.0-2 librtacrc1 python3-kismetcapturefrakabszigbee samba-ad-provision
libjs-jquery-ui libtheoraen1 python3-kismetcapturertl33 samba-dsdb-modules
libjs-underScore libudfread0 python3-kismetcapturertladsb
libmongoc-1.0-6t64 libx264-164 python3-kismetcapturertlamr
Use 'sudo apt autoremove' to remove them.

Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 4

(kali㉿kali)-[~]
$
```

Starting and enabling Apache

We used the following commands to start and enable Apache **sudo systemctl start apache2**, **sudo systemctl enable apache2**, **sudo systemctl status apache2** these commands are used to manage the Apache web server on Linux with systemctl. Running sudo systemctl start apache2 starts Apache immediately, while sudo systemctl enable apache2 ensures it will automatically launch every time the system boots. Finally, sudo systemctl status apache2 shows whether Apache is currently running, if it's enabled, and provides recent log details for troubleshooting.



```
(kali㉿kali)-[~]
$ /home/kali
Session Actions Edit View Help

(kali㉿kali)-[~]
$ sudo systemctl start apache2
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)

(kali㉿kali)-[~]
$ sudo systemctl enable apache2
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)

Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
Created symlink '/etc/systemd/system/multi-user.target.wants/apache2.service' → '/usr/lib/systemd/system/apache2.service'.

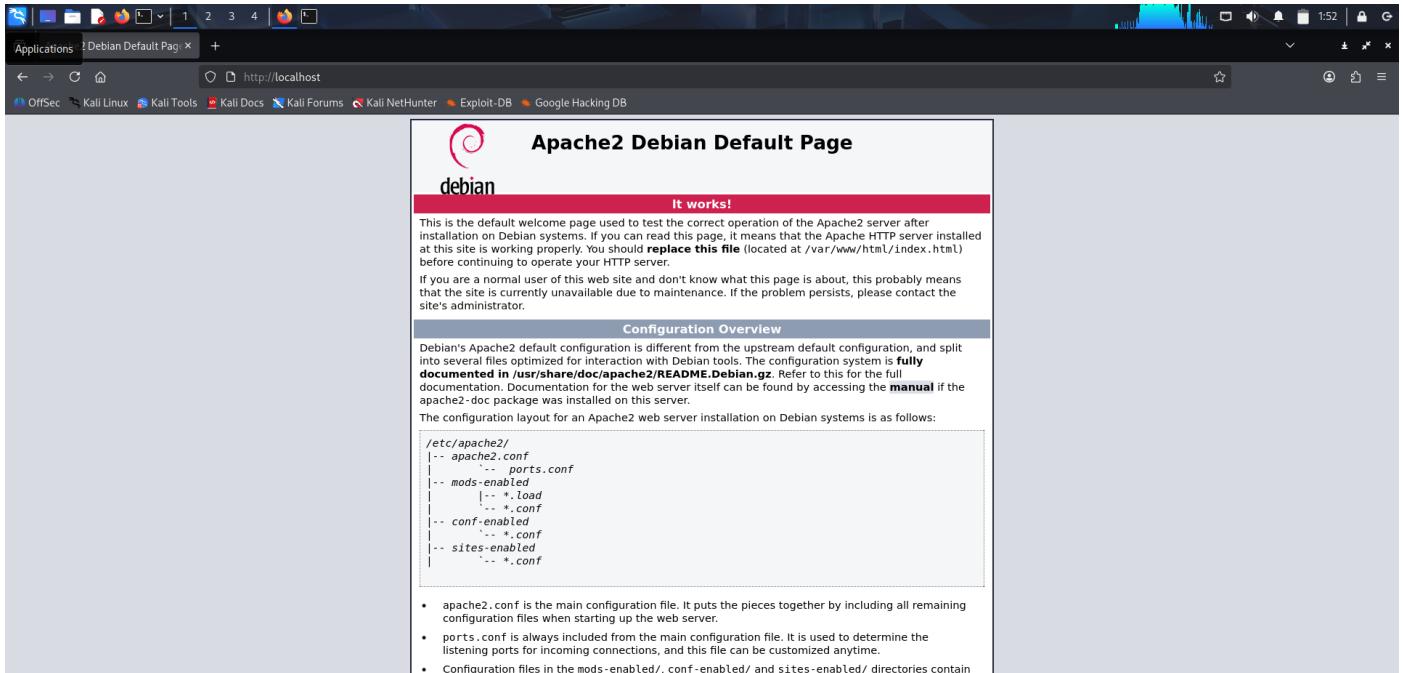
(kali㉿kali)-[~]
$ sudo systemctl status apache2
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: disabled)
     Active: active (running) since Sun 2025-11-09 01:28:17 SAST; 1min 21s ago
       Invocation: cfb4ded5988349a68aeec8a166d15f26
      Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 27048 (apache2)
      Tasks: 22 (limit: 2100)
     Memory: 22M (peak: 22.2M)
        CPU: 175ms
       CGroup: /system.slice/apache2.service
           ├─27048 /usr/sbin/apache2 -k start
           ├─27049 /usr/sbin/apache2 -k start
           ├─27051 /usr/sbin/apache2 -k start
           ├─27053 /usr/sbin/apache2 -k start
           └─27054 /usr/sbin/apache2 -k start
Nov 09 01:28:17 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Nov 09 01:28:17 kali systemd[1]: Started apache2.service - The Apache HTTP Server.

(kali㉿kali)-[~]
$
```

Testing Apache

We opened the Firefox web browser and typed <https://localhost> which previewed the Apache Debian default page this is a local intranet page



Custom intranet page

we typed the following command **sudo rm /var/www/html/index.html** which removed the default Apache index file and then followed by the **sudo nano /var/www/html/index.html** which allowed us to create a new index file a customized index file we saved the file.

```
(kali㉿kali)-[~]
└─$ sudo rm /var/www/html/index.html
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)

(kali㉿kali)-[~]
└─$ sudo nano /var/www/html/index.html
/etc/sudoers.d/capaciti:1:22: syntax error
capaciti ALL=(ALL:ALL)

(kali㉿kali)-[~]
└─$
```

```

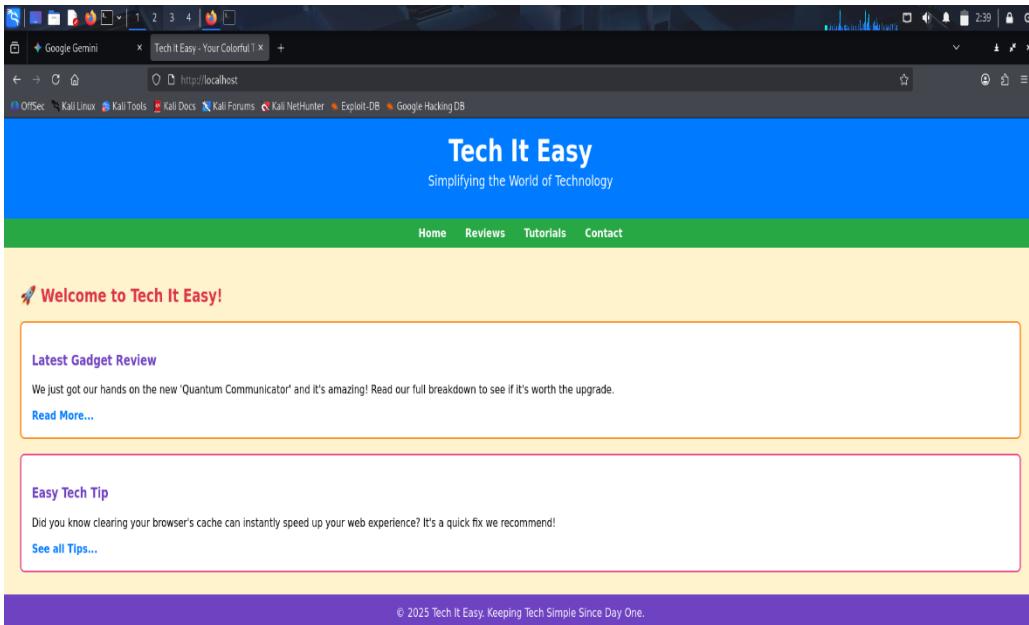
nano index.html
^D
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tech It Easy - Your Colorful Tech Guide</title>
</head>
<body style="margin: 0; font-family: sans-serif;">
    <header style="background-color: #007bff; color: white; padding: 20px; text-align: center;">
        <a href="#" style="margin: 0; font-size: 2.5em; font-weight: bold;">Home</a>
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Reviews</a>
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Tutorials</a>
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Contact</a>
    </header>
    <nav style="background-color: #28a745; padding: 10px 0; text-align: center;">
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Home</a>
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Reviews</a>
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Tutorials</a>
        <a href="#" style="color: white; text-decoration: none; margin: 0 15px; font-weight: bold;">Contact</a>
    </nav>
    <main style="background-color: #fff3cd; padding: 30px; min-height: 400px;">
        <h2 style="color: #dc3545;">Welcome to Tech It Easy!</h2>
        <article style="background-color: white; border: 3px solid #fd7e14; padding: 20px; margin-bottom: 20px; border-radius: 8px;">
            <h3 style="color: #f6f4c1;">Latest Gadget Review</h3>
            <p>We just got our hands on the new 'Quantum Communicator' and it's amazing! Read our full breakdown to see if it's worth the upgrade.</p>
            <a href="#" style="color: #007bff; text-decoration: none; font-weight: bold;">Read More ... </a>
        </article>
        <article style="background-color: white; border: 2px solid #e83e8c; padding: 20px; border-radius: 8px;">
            <h3 style="color: #f6f4c1;">Easy Tech Tip</h3>
            <p>Did you know clearing your browser's cache can instantly speed up your web experience? It's a quick fix we recommend!</p>
            <a href="#" style="color: #007bff; text-decoration: none; font-weight: bold;">See all Tips ... </a>
        </article>
    </main>
    <footer style="background-color: #d8f4c1; color: white; text-align: center; padding: 15px;">
        <p style="margin: 0;">© 2025 Tech It Easy. Keeping Tech Simple Since Day One.</p>
    </footer>
</body>
</html>

```

The screenshot shows a terminal window titled 'Firefox ESR' with the command 'nano index.html' entered. The content of the file is displayed, showing an HTML document structure with a header, navigation bar, main content area containing two articles (one for reviews and one for tech tips), and a footer. The nano text editor interface is visible at the bottom.

We then created our custom page on the nano text editor.

We then tested the custom page on the browser by typing <https://localhost> which then displayed our custom intranet page.



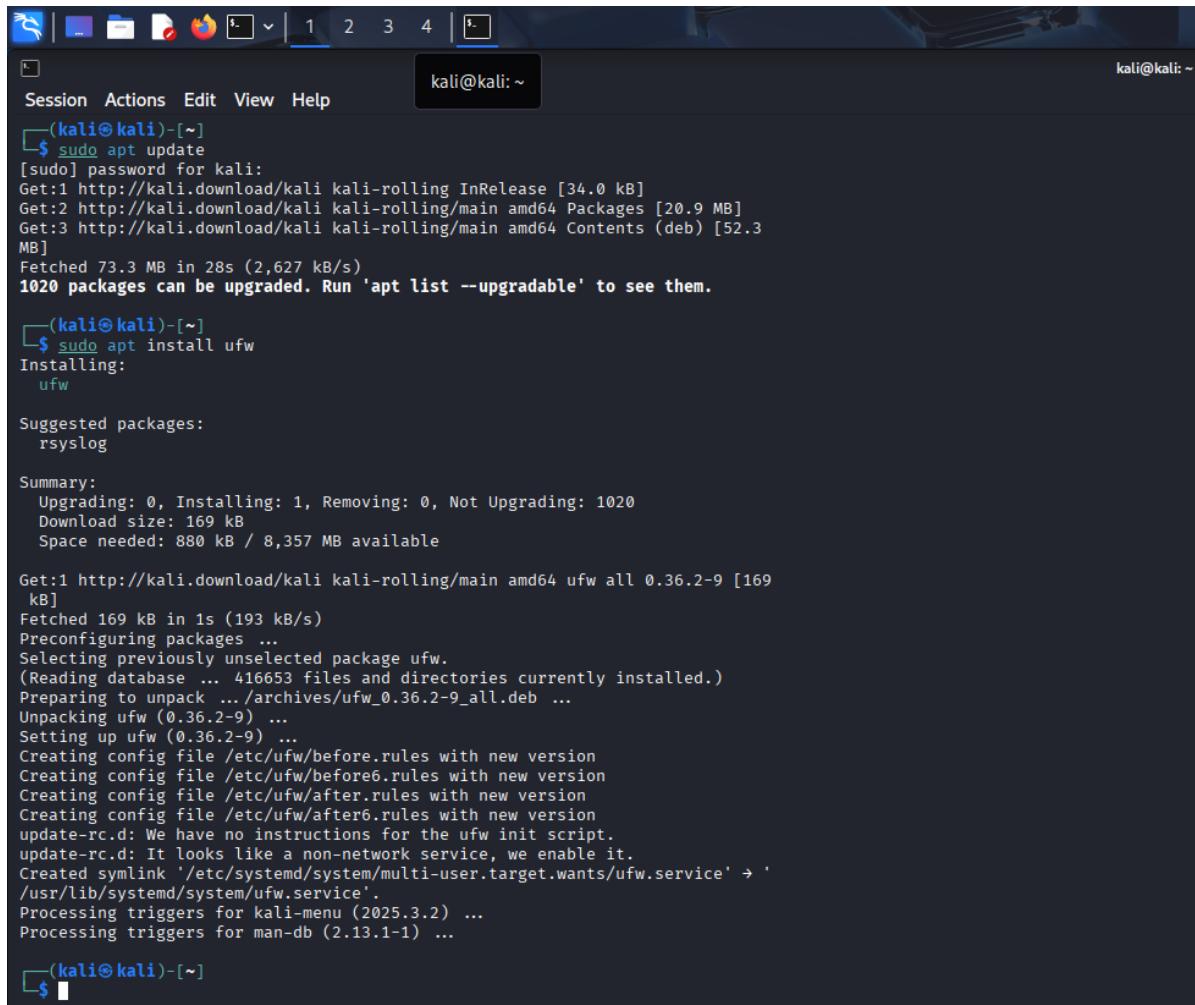
Security

Enabling firewall

We are going to install the UFW (Uncomplicated Firewall), which is a front-end for iptables. The UFW (Uncomplicated Firewall) is a user-friendly front-end for iptables, designed to simplify firewall management on Linux systems like Kali. It allows users to easily configure rules for allowing or denying network traffic using straightforward commands, while iptables handles the complex backend processing. UFW is ideal for beginners and sysadmins who want quick, effective control over their system's security.

Installing ufw

We started by updating the packages, **using sudo apt update** we then installed the uncomplicated firewall (ufw).



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar says "kali@kali: ~". The terminal content is as follows:

```
(kali㉿kali)-[~]
$ sudo apt update
[sudo] password for kali:
Get:1 http://kali.download/kali kali-rolling InRelease [34.0 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [20.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [52.3
MB]
Fetched 73.3 MB in 28s (2,627 kB/s)
1020 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali㉿kali)-[~]
$ sudo apt install ufw
Installing:
  ufw

Suggested packages:
  rsyslog

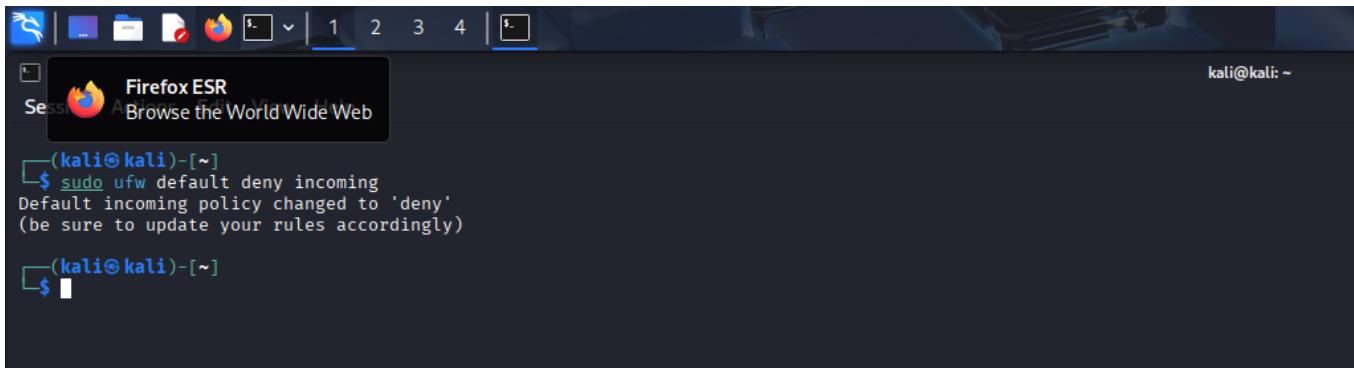
Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 1020
  Download size: 169 kB
  Space needed: 880 kB / 8,357 MB available

Get:1 http://kali.download/kali kali-rolling/main amd64 ufw all 0.36.2-9 [169
kB]
Fetched 169 kB in 1s (193 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ufw.
(Reading database ... 416653 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36.2-9_all.deb ...
Unpacking ufw (0.36.2-9) ...
Setting up ufw (0.36.2-9) ...
Creating config file /etc/ufw/before.rules with new version
Creating config file /etc/ufw/before6.rules with new version
Creating config file /etc/ufw/after.rules with new version
Creating config file /etc/ufw/after6.rules with new version
update-rc.d: We have no instructions for the ufw init script.
update-rc.d: It looks like a non-network service, we enable it.
Created symlink '/etc/systemd/system/multi-user.target.wants/ufw.service' → '/usr/lib/systemd/system/ufw.service'.
Processing triggers for kali-menu (2025.3.2) ...
Processing triggers for man-db (2.13.1-1) ...

(kali㉿kali)-[~]
$
```

Enable UFW and set default rules

We used the command **sudo ufw default deny incoming** sets the default policy for incoming network traffic to deny. This means that unless you explicitly allow a connection (e.g., via sudo ufw allow 22 for SSH), all incoming traffic will be blocked. It's a security-first approach that helps protect your system from unauthorized access by only permitting traffic you specifically approve. This step is part of hardening your firewall before enabling it with sudo ufw enable.

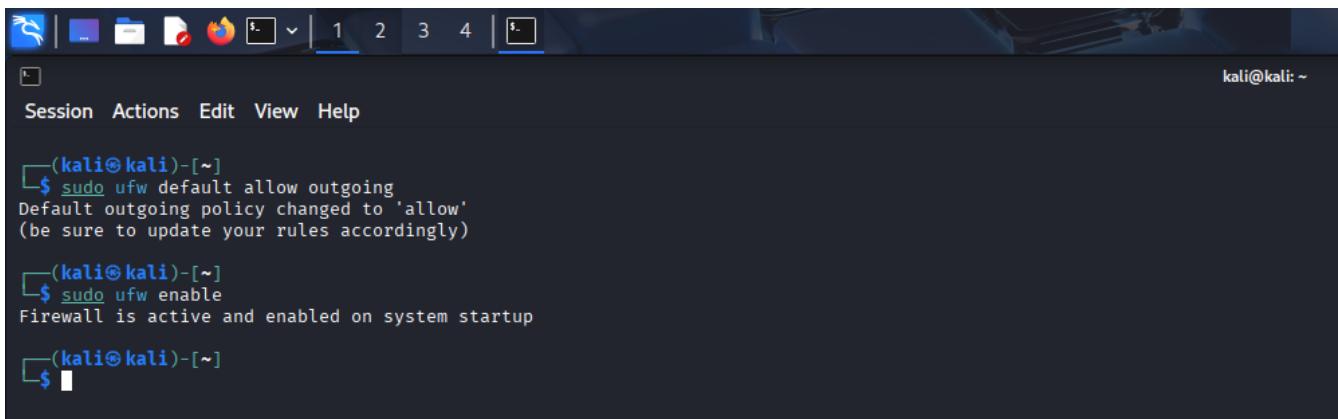


```
(kali㉿kali)-[~]
$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)

(kali㉿kali)-[~]
$
```

We then used the command **sudo ufw default allow outgoing** sets the default policy to allow all outgoing traffic from your system. This means your computer can initiate connections to other devices or servers (like browsing websites or sending emails) without restriction. It's a common setting because outgoing traffic is generally safe.

Then, **sudo ufw enable** activates the firewall with the rules you've configured. After this step, UFW starts enforcing your default policies and any specific rules you've added. So with these two commands, we are allowing your system to freely send data out while blocking incoming traffic unless explicitly permitted.



```
Session Actions Edit View Help

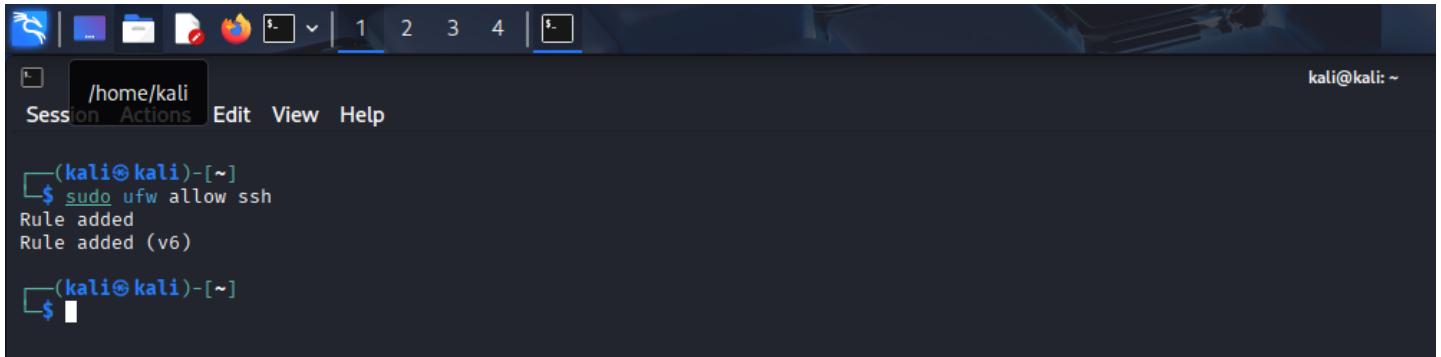
(kali㉿kali)-[~]
$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)

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

(kali㉿kali)-[~]
$
```

Allow specific services

We used the command **sudo ufw allow ssh** creates a firewall rule that permits incoming SSH connections, typically on port 22. This is essential if you want to remotely access your system via SSH after enabling UFW, since the default policy usually blocks all incoming traffic. By allowing SSH, we are telling UFW to make an exception for secure remote login, ensuring we don't lock our out of your system.

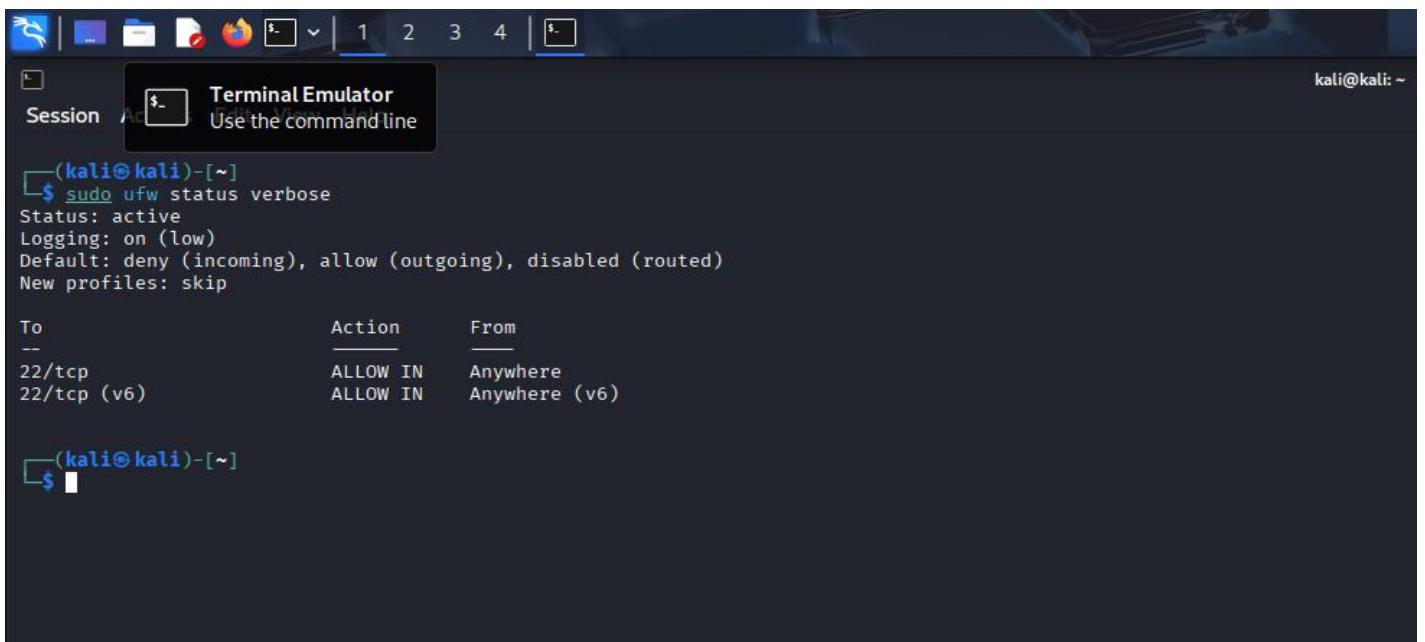


```
(kali㉿kali)-[~]
$ sudo ufw allow ssh
Rule added
Rule added (v6)

(kali㉿kali)-[~]
$
```

Checking status

As a team, we ran the command **sudo ufw status verbose** to assess the current state of our system's firewall. This gave us a comprehensive overview of UFW's configuration, confirming whether it was active and detailing the default policies for incoming and outgoing traffic. We reviewed the list of active rules to ensure that only authorized connections are permitted and verified that logging is appropriately configured for monitoring purposes. This collective check helps us maintain a secure network environment and ensures that our firewall settings align with our operational and security protocols.



```
Terminal Emulator
Use the command line

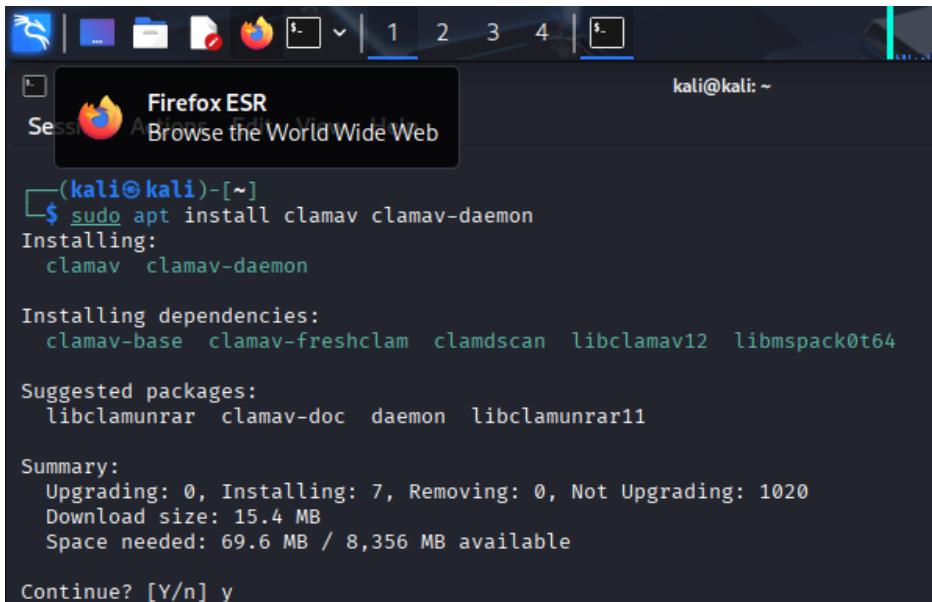
(kali㉿kali)-[~]
$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

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

(kali㉿kali)-[~]
$
```

Set Up Basic Antivirus

As a team, we proceeded to enhance our system's security by installing ClamAV, an open-source antivirus solution, using the command **sudo apt install clamav clamav-daemon**. This step ensures we have real-time scanning capabilities and a reliable toolset for detecting and mitigating potential threats across our environment. It reflects our collective commitment to maintaining a secure and resilient infrastructure.



```
(kali㉿kali)-[~]
$ sudo apt install clamav clamav-daemon
Installing:
  clamav  clamav-daemon

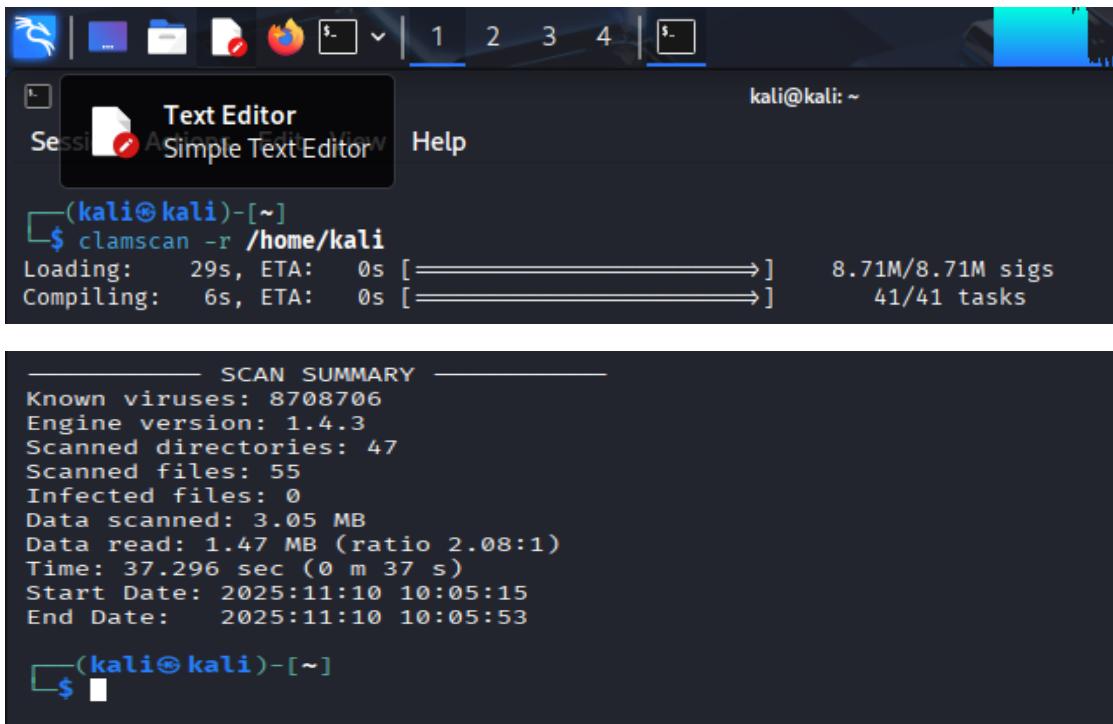
Installing dependencies:
  clamav-base  clamav-freshclam  clamdscan  libclamav12  libmspack0t64

Suggested packages:
  libclamunrar  clamav-doc  daemon  libclamunrar11

Summary:
  Upgrading: 0, Installing: 7, Removing: 0, Not Upgrading: 1020
  Download size: 15.4 MB
  Space needed: 69.6 MB / 8,356 MB available

Continue? [Y/n] y
```

We then went forward with scanning the kali directory using the antivirus.

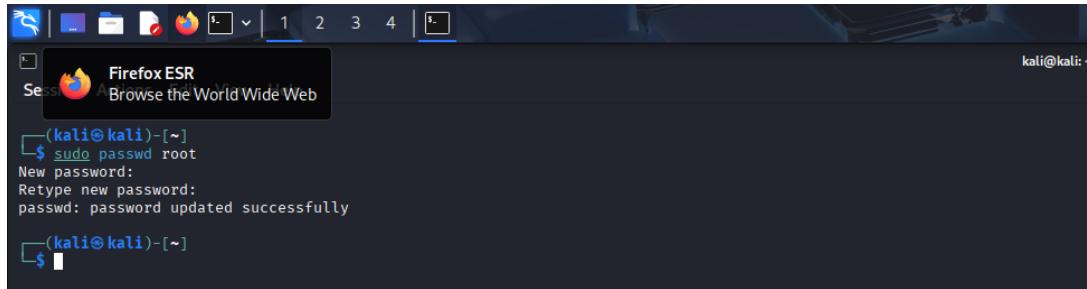


```
(kali㉿kali)-[~]
$ clamscan -r /home/kali
Loading: 29s, ETA: 0s [=====] 8.71M/8.71M sigs
Compiling: 6s, ETA: 0s [=====] 41/41 tasks
```



```
----- SCAN SUMMARY -----
Known viruses: 8708706
Engine version: 1.4.3
Scanned directories: 47
Scanned files: 55
Infected files: 0
Data scanned: 3.05 MB
Data read: 1.47 MB (ratio 2.08:1)
Time: 37.296 sec (0 m 37 s)
Start Date: 2025:11:10 10:05:15
End Date: 2025:11:10 10:05:53
```

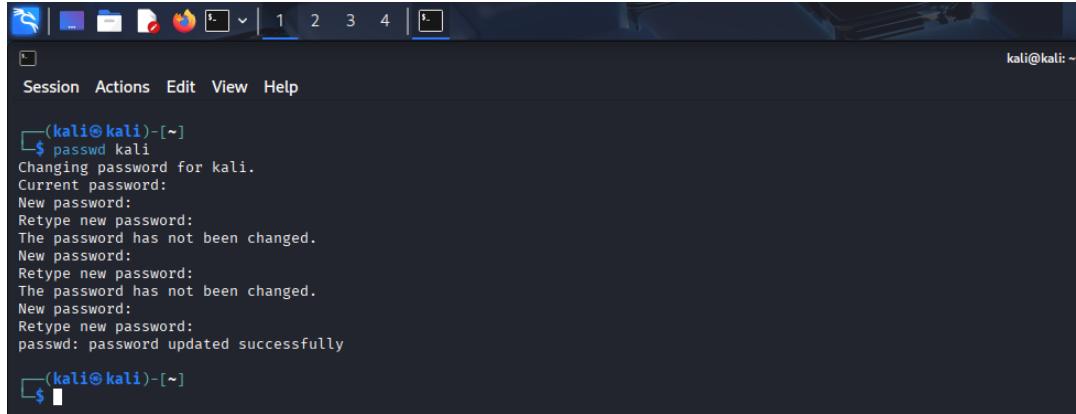
Change Default Passwords



A screenshot of a terminal window titled "Firefox ESR" showing a root shell session. The command `sudo passwd root` is run, followed by entering a new password and re-entering it. The output shows the password was updated successfully.

```
(kali㉿kali)-[~]
$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
(kali㉿kali)-[~]
$
```

Change user password

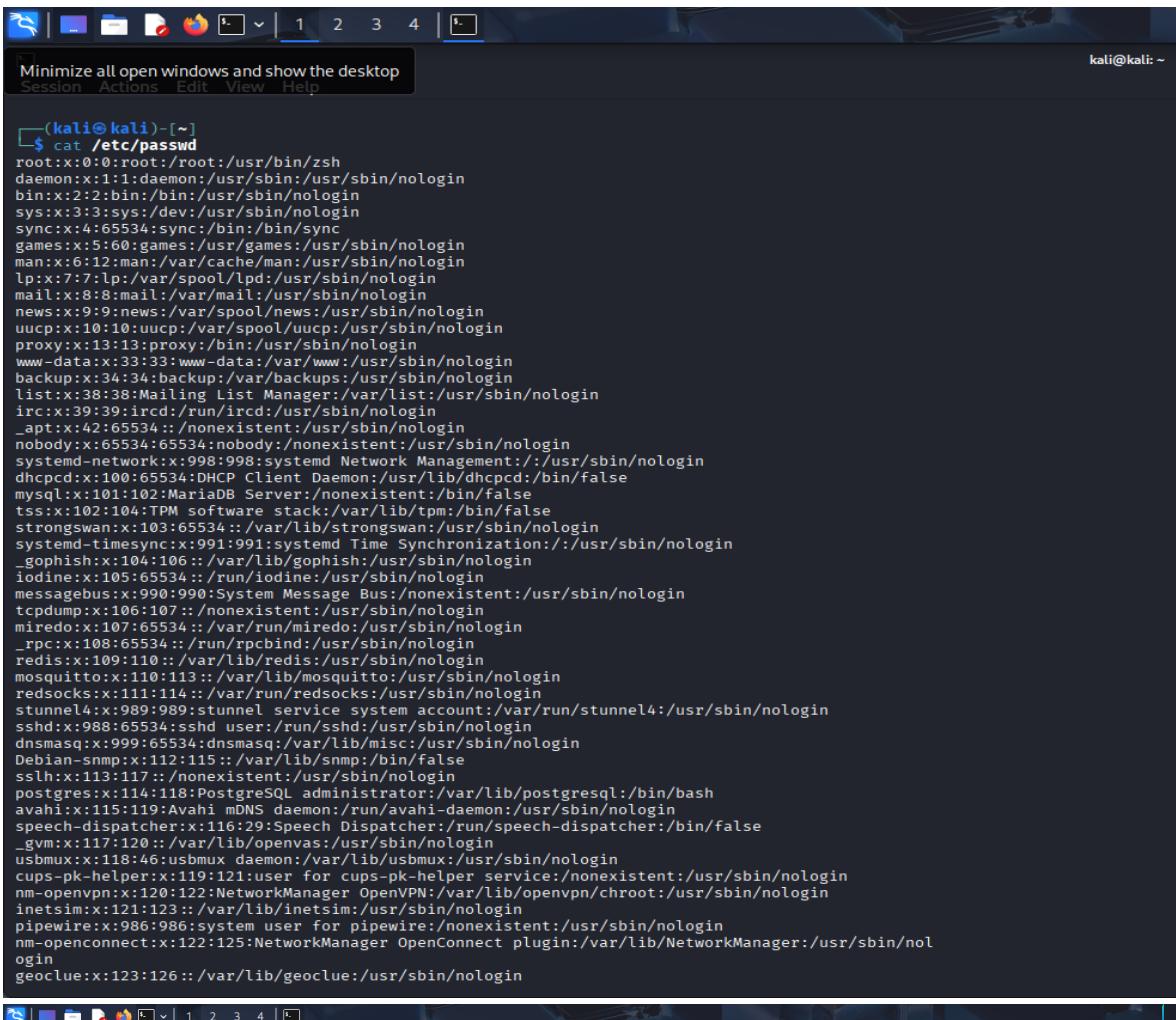


A screenshot of a terminal window showing a user password being changed. The command `passwd kali` is run, followed by entering a new password and re-entering it. The output shows the password was updated successfully.

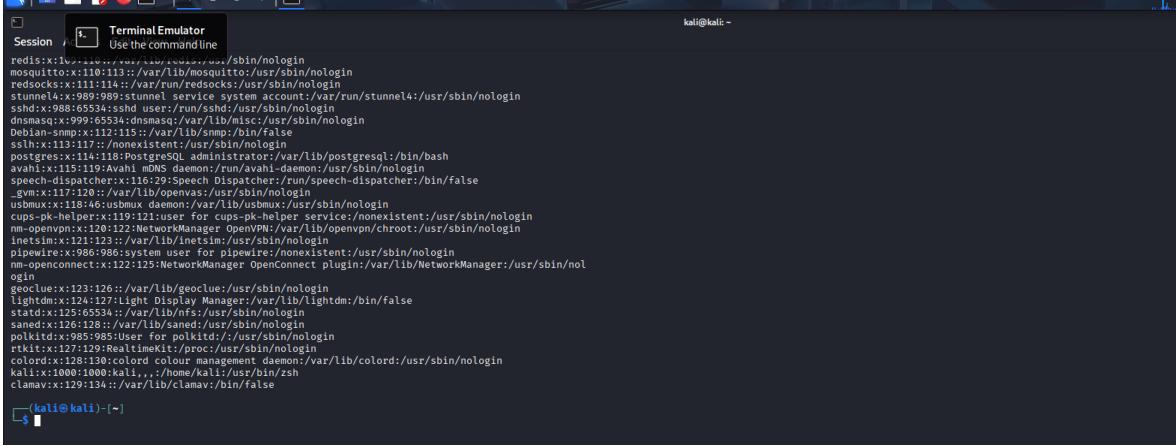
```
Session Actions Edit View Help
(kali㉿kali)-[~]
$ passwd kali
Changing password for kali.
Current password:
New password:
Retype new password:
The password has not been changed.
New password:
Retype new password:
The password has not been changed.
New password:
Retype new password:
The password has not been changed.
passwd: password updated successfully
(kali㉿kali)-[~]
$
```

Audit user accounts

As a group, we audited user accounts on our system by reviewing the contents of `/etc/passwd` using the command **cat /etc/passwd**. This file lists all user accounts and system services, providing details such as usernames, user IDs (UIDs), group IDs (GIDs), home directories, and default shells. By examining these entries together, we ensured that only authorized users are present, identified any legacy or unused accounts, and confirmed that system accounts are properly configured. This collaborative review strengthens our overall system security and accountability.



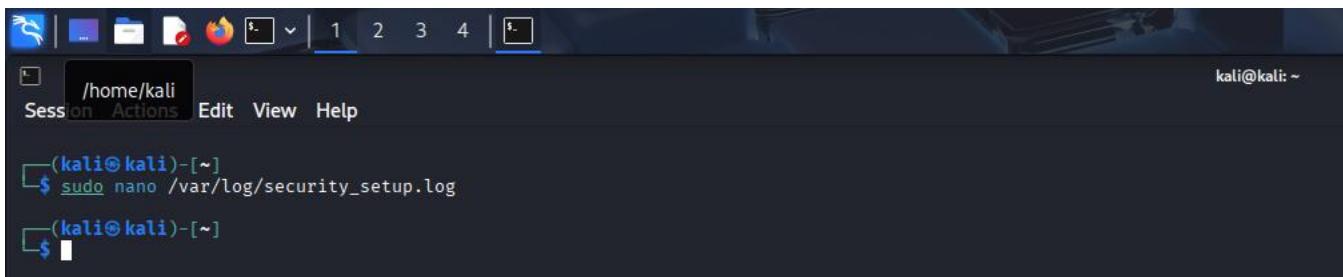
```
(kali㉿kali)-[~]
$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backupr:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon:/usr/lib/dhcpcd:/bin/false
mysql:x:101:102:MariaDB Server:/nonexistent:/bin/false
tss:x:102:103:TPM software stack:/var/lib/tpm:/bin/false
strongswan:x:103:65534::/var/lib/strongswan:/usr/sbin/nologin
systemd-timesync:x:991:991:systemd Time Synchronization:/:/usr/sbin/nologin
_gophish:x:104:106::/var/lib/gophish:/usr/sbin/nologin
iodine:x:105:65534::/run/iodine:/usr/sbin/nologin
messagebus:x:990:990:System Message Bus:/nonexistent:/usr/sbin/nologin
tcpdump:x:106:107::/nonexistent:/usr/sbin/nologin
miredo:x:107:65534::/var/run/miredo:/usr/sbin/nologin
_rpc:x:108:65534::/run/rpcbind:/usr/sbin/nologin
redis:x:109:110::/var/lib/redis:/usr/sbin/nologin
mosquitto:x:110:113::/var/lib/mosquitto:/usr/sbin/nologin
redsocks:x:111:114::/var/run/redsocks:/usr/sbin/nologin
stunnel4:x:989:989:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
sshd:x:988:65534:sshd user:/run/sshd:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
Debian-snmp:x:112:115::/var/lib/snmp:/bin/false
sslh:x:113:117::/nonexistent:/usr/sbin/nologin
postgres:x:114:118:PostgreSQL administrator:/var/lib/postgresql:/bin/bash
avahi:x:115:119:Avahi mDNS daemon:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:116:29:Speech Dispatcher:/run/speech-dispatcher:/bin/false
_gvm:x:117:120::/var/lib/openvas:/usr/sbin/nologin
usbmux:x:118:46:usbmux daemon:/var/lib/usbmux:/usr/sbin/nologin
cups-pk-helper:x:119:121:user for cups-pk-helper service:/nonexistent:/usr/sbin/nologin
nm-openvpn:x:120:122:NetworkManager OpenVPN:/var/lib/openvpn/chroot:/usr/sbin/nologin
inetSim:x:121:123::/var/lib/inetSim:/usr/sbin/nologin
pipewire:x:986:986:system user for pipewire:/nonexistent:/usr/sbin/nologin
nm-openconnect:x:122:125:NetworkManager OpenConnect plugin:/var/lib/NetworkManager:/usr/sbin/nologin
geoclue:x:123:126::/var/lib/geoclue:/usr/sbin/nologin
```



```
(kali㉿kali)-[~]
$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backupr:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon:/usr/lib/dhcpcd:/bin/false
mysql:x:101:102:MariaDB Server:/nonexistent:/bin/false
tss:x:102:103:TPM software stack:/var/lib/tpm:/bin/false
strongswan:x:103:65534::/var/lib/strongswan:/usr/sbin/nologin
systemd-timesync:x:991:991:systemd Time Synchronization:/:/usr/sbin/nologin
_gophish:x:104:106::/var/lib/gophish:/usr/sbin/nologin
iodine:x:105:65534::/run/iodine:/usr/sbin/nologin
messagebus:x:990:990:System Message Bus:/nonexistent:/usr/sbin/nologin
tcpdump:x:106:107::/nonexistent:/usr/sbin/nologin
miredo:x:107:65534::/var/run/miredo:/usr/sbin/nologin
_rpc:x:108:65534::/run/rpcbind:/usr/sbin/nologin
redis:x:109:110::/var/lib/redis:/usr/sbin/nologin
mosquitto:x:110:113::/var/lib/mosquitto:/usr/sbin/nologin
redsocks:x:111:114::/var/run/redsocks:/usr/sbin/nologin
stunnel4:x:989:989:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
sshd:x:988:65534:sshd user:/run/sshd:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
Debian-snmp:x:112:115::/var/lib/snmp:/bin/false
sslh:x:113:117::/nonexistent:/usr/sbin/nologin
postgres:x:114:118:PostgreSQL administrator:/var/lib/postgresql:/bin/bash
avahi:x:115:119:Avahi mDNS daemon:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:116:29:Speech Dispatcher:/run/speech-dispatcher:/bin/false
_gvm:x:117:120::/var/lib/openvas:/usr/sbin/nologin
usbmux:x:118:46:usbmux daemon:/var/lib/usbmux:/usr/sbin/nologin
cups-pk-helper:x:119:121:user for cups-pk-helper service:/nonexistent:/usr/sbin/nologin
nm-openvpn:x:120:122:NetworkManager OpenVPN:/var/lib/openvpn/chroot:/usr/sbin/nologin
inetSim:x:121:123::/var/lib/inetSim:/usr/sbin/nologin
pipewire:x:986:986:system user for pipewire:/nonexistent:/usr/sbin/nologin
nm-openconnect:x:122:125:NetworkManager OpenConnect plugin:/var/lib/NetworkManager:/usr/sbin/nologin
geoclue:x:123:126::/var/lib/geoclue:/usr/sbin/nologin
lightdm:x:124:127:Light Display Manager:/var/lib/lightdm:/bin/false
statd:x:125:65534::/var/lib/nfs:/usr/sbin/nologin
saned:x:126:128::/var/lib/saned:/usr/sbin/nologin
polkitd:x:985:985:User for polkitd:/:/usr/sbin/nologin
rtkit:x:127:129:RealtimeKit:/proc:/usr/sbin/nologin
colord:x:128:130:colord colour management daemon:/var/lib/colord:/usr/sbin/nologin
kali:x:1000:1000:kali,,,:/home/kali:/usr/bin/zsh
clamav:x:129:134::/var/lib/clamav:/bin/false
```

Document Security Settings

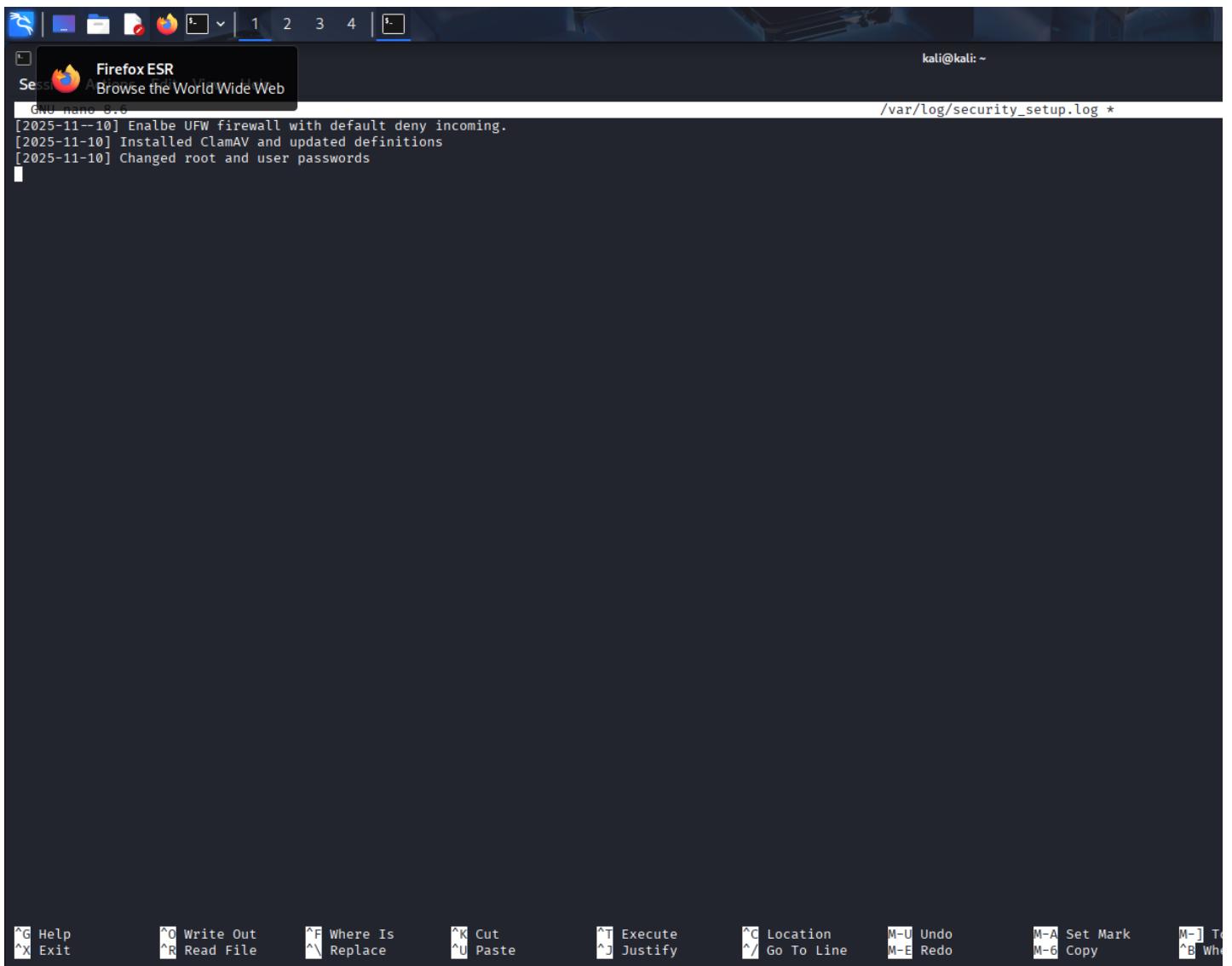
As a team, we documented our system's security configurations by editing the `/var/log/security_setup.log` file using the command `sudo nano /var/log/security_setup.log`. This allowed us to record key actions such as firewall status, antivirus installation, user account audits, and any other relevant security measures we implemented. Maintaining this log ensures transparency, supports future audits, and helps us track changes to our security posture over time.



/home/kali Session Actions Edit View Help

```
(kali㉿kali)-[~]
$ sudo nano /var/log/security_setup.log
(kali㉿kali)-[~]
$
```

This command opened the GNU nano text editor on the command line which allowed us to log system changes.



Firefox ESR Session A

kali@kali: ~

/var/log/security_setup.log *

```
GNU nano 8.0
[2025-11-10] Enabled UFW Firewall with default deny incoming.
[2025-11-10] Installed ClamAV and updated definitions
[2025-11-10] Changed root and user passwords
```

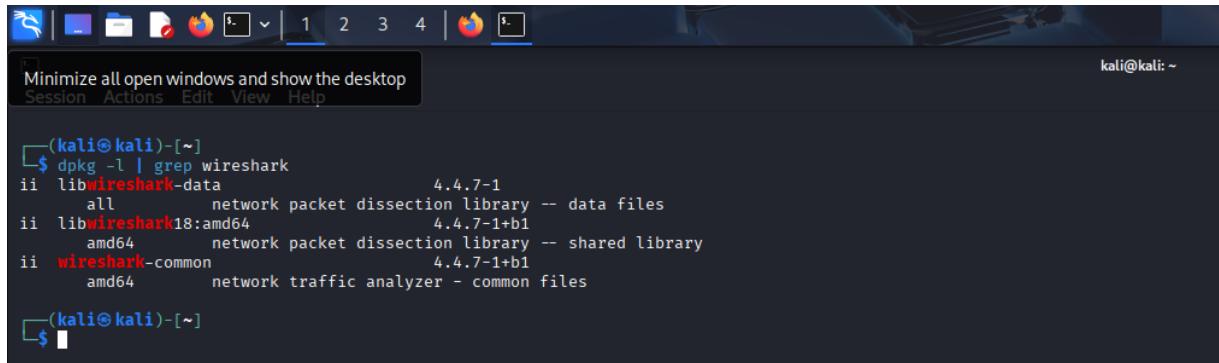
File Edit Insert View Search Plugins Tools Help

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^V Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^B Who

Troubleshooting & Documentation

Problem 1 Wireshark won't launch

User attempts to launch Wireshark from the XFCE menu and terminal, but no GUI appears. We verified Wireshark is installed using this command `dpkg -l | grep Wireshark`, we then discovered that there are core components of Wireshark installed in the system. Meaning Wireshark not installed.

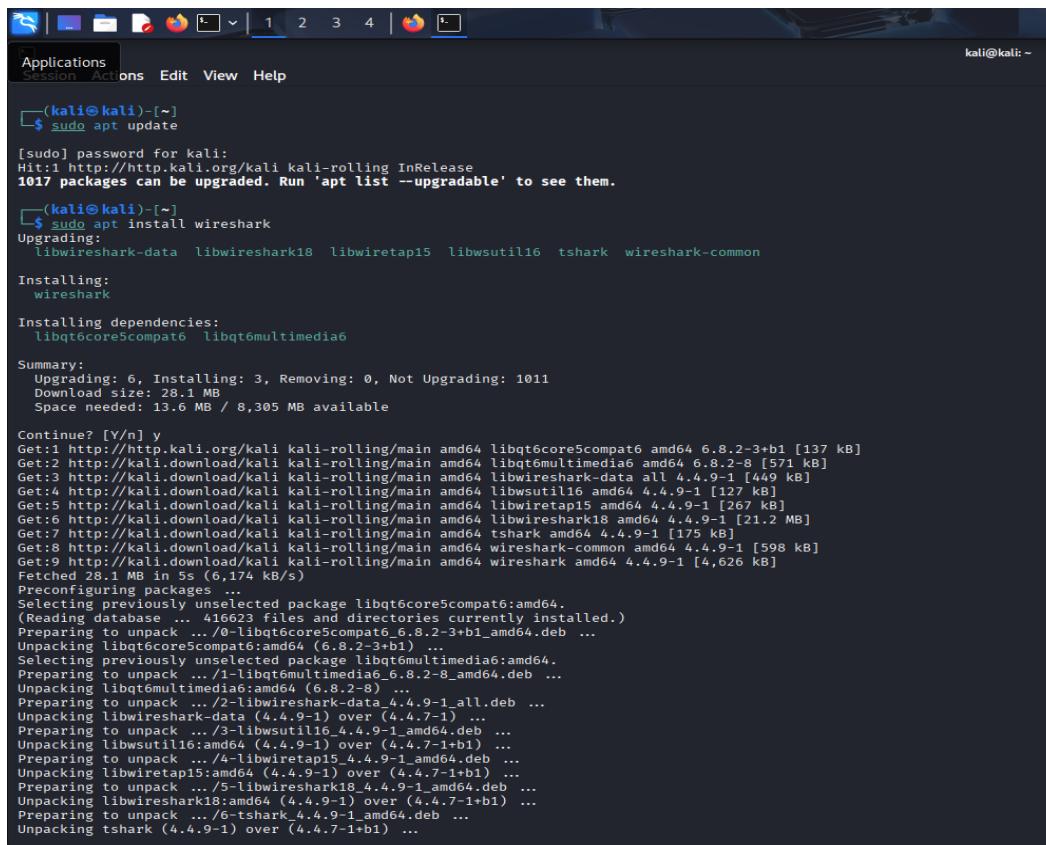


```
(kali㉿kali)-[~]
$ dpkg -l | grep wireshark
ii  libwireshark-data          4.4.7-1
   all      network packet dissection library -- data files
ii  libwireshark18:amd64       4.4.7-1+b1
   amd64    network packet dissection library -- shared library
ii  wireshark-common          4.4.7-1+b1
   amd64    network traffic analyzer - common files

(kali㉿kali)-[~]
$
```

Installing Wireshark

We then updated the packages using the `sudo apt update` command and then went on to install wire shark using the `sudo apt install wireshark` command.



```
[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1017 packages can be upgraded. Run 'apt list --upgradable' to see them.

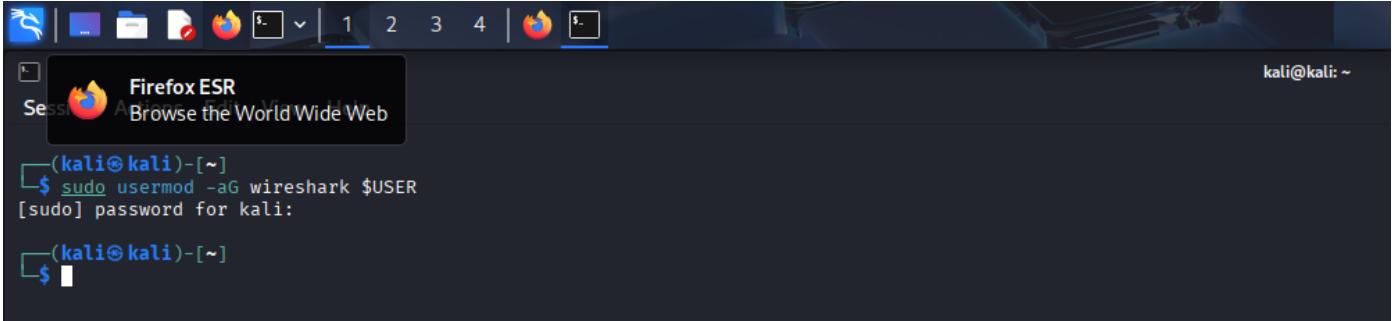
(kali㉿kali)-[~]
$ sudo apt update
[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
1017 packages can be upgraded. Run 'apt list --upgradable' to see them.

(kali㉿kali)-[~]
$ sudo apt install wireshark
Upgrading:
 libwireshark-data libwireshark18 libwiretap15 libwsutil16 tshark wireshark-common
Installing:
 wireshark
Installing dependencies:
 libqt6core5compat libqt6multimedia

Summary:
 Upgrading: 6, Installing: 3, Removing: 0, Not Upgrading: 1011
 Download size: 28.1 MB
 Space needed: 13.6 MB / 8,305 MB available

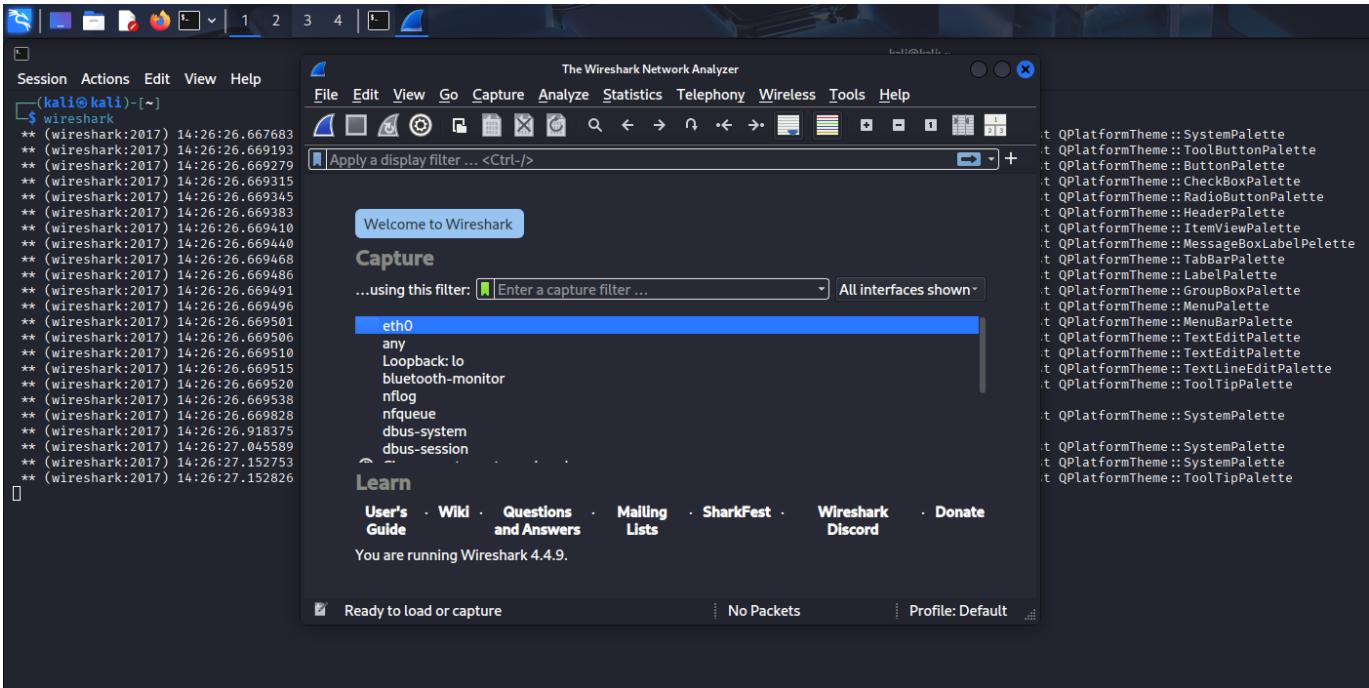
Continue? [Y/n] y
Get:1 http://http.kali.org/kali kali-rolling/main amd64 libqt6core5compat6 amd64 6.8.2-3+b1 [137 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 libqt6multimedia6 amd64 6.8.2-8 [571 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 libwireshark-data all 4.4.9-1 [449 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 libwsutil16 amd64 4.4.9-1 [127 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 libwiretap15 amd64 4.4.9-1 [267 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 libwireshark18 amd64 4.4.9-1 [21.2 MB]
Get:7 http://kali.download/kali kali-rolling/main amd64 tshark amd64 4.4.9-1 [175 kB]
Get:8 http://kali.download/kali kali-rolling/main amd64 wireshark-common amd64 4.4.9-1 [598 kB]
Get:9 http://kali.download/kali kali-rolling/main amd64 wireshark amd64 4.4.9-1 [4,626 kB]
Fetched 28.1 MB in 5s (6,174 kB/s)
Preconfiguring packages...
Selecting previously unselected package libqt6core5compat6:amd64.
(Reading database ... 416623 files and directories currently installed.)
Preparing to unpack .../0-libqt6core5compat6_6.8.2-3+b1_amd64.deb ...
Unpacking libqt6core5compat6:amd64 (6.8.2-3+b1) ...
Selecting previously unselected package libqt6multimedia6:amd64.
Preparing to unpack .../1-libqt6multimedia6_6.8.2-8_amd64.deb ...
Unpacking libqt6multimedia6:amd64 (6.8.2-8)
Preparing to unpack .../2-libwireshark-data_4.4.9-1_all.deb ...
Unpacking libwireshark-data (4.4.9-1) over (4.4.7-1)
Preparing to unpack .../3-libwsutil16_4.4.9-1_amd64.deb ...
Unpacking libwsutil16:amd64 (4.4.9-1) over (4.4.7-1+b1)
Preparing to unpack .../4-libwiretap15_4.4.9-1_amd64.deb ...
Unpacking libwiretap15:amd64 (4.4.9-1) over (4.4.7-1+b1)
Preparing to unpack .../5-libwireshark18_4.4.9-1_amd64.deb ...
Unpacking libwireshark18:amd64 (4.4.9-1) over (4.4.7-1+b1)
Preparing to unpack .../6-tshark_4.4.9-1_amd64.deb ...
Unpacking tshark (4.4.9-1) over (4.4.7-1+b1) ...
```

The command **sudo usermod -aG wireshark \$USER** adds the current user to the wireshark group, allowing them to capture network packets with Wireshark without needing root privileges. The -aG flags ensure the user is added to the group without being removed from others, and sudo is required because modifying user groups needs administrative access. After running it, a logout or reboot may be needed for the change to take effect.



```
Firefox ESR
Session Actions Edit View
(kali㉿kali)-[~]
$ sudo usermod -aG wireshark $USER
[sudo] password for kali:
(kali㉿kali)-[~]
$
```

We then logged out then restarted the computer then opened terminal then typed wireshark which the opened the wireshark GUI

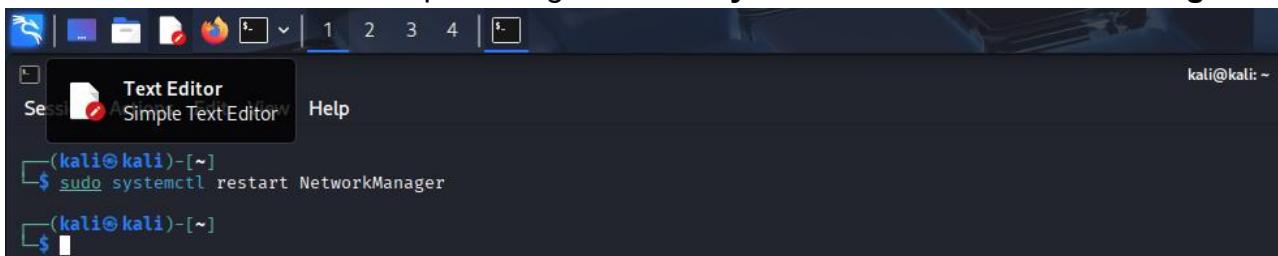


wireshark GUI launched without issue.

problem 2 slow internet access

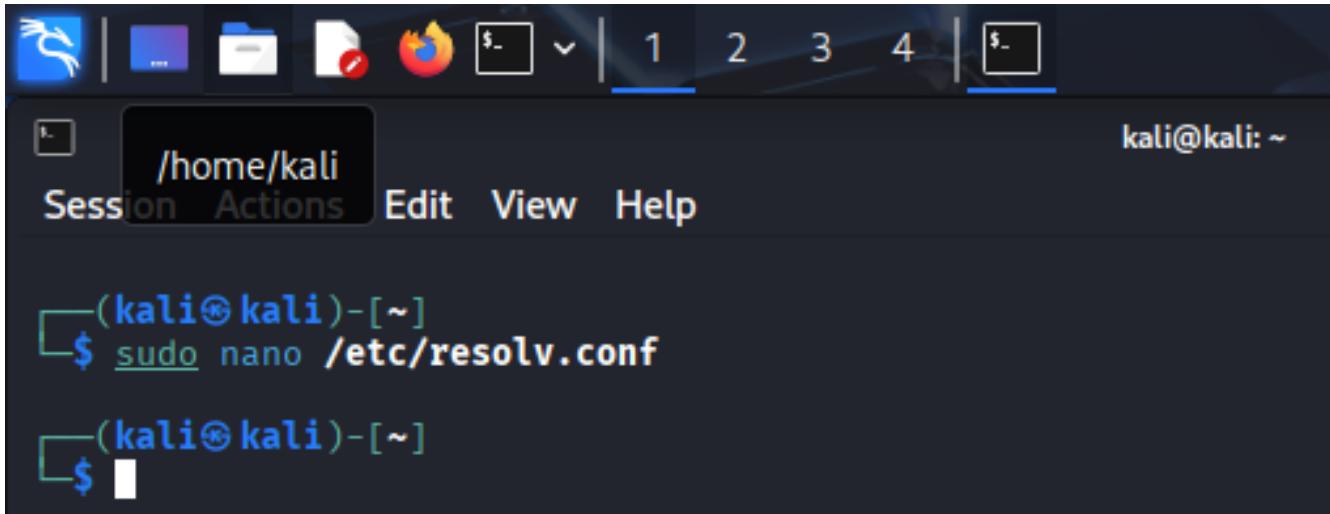
We had sluggish internet connection. Initial network diagnostics revealed high latency and intermittent packet loss when running ping -c 4 google.com, indicating potential connectivity issues. Using nmcli dev wifi, the Wi-Fi signal strength was found to be weak at approximately 30%. After relocating closer to the router, the signal strength significantly improved to around 80%, suggesting that distance from the router was a key factor affecting network performance.

We restarted the network adapter using the **sudo systemctl restart NetworkManager**



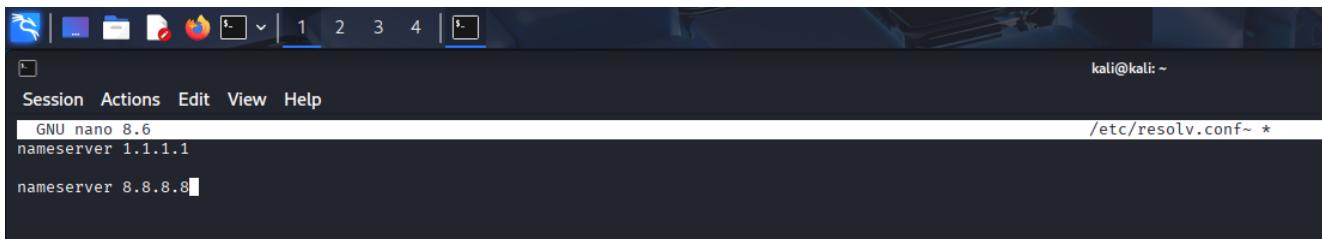
```
kali@kali: ~
Session Actions Edit View Help
(kali㉿kali)-[~]
$ sudo systemctl restart NetworkManager
(kali㉿kali)-[~]
$
```

We opened the DNS configuration file using the **sudo nano /etc/resolv.conf** command



```
kali@kali: ~
Session Actions Edit View Help
(kali㉿kali)-[~]
$ sudo nano /etc/resolv.conf
(kali㉿kali)-[~]
$
```

then opened GNU nano text editor and added cleared the file and added the following nameserver 1.1.1.1 and nameserver 8.8.8.8



```
kali@kali: ~
Session Actions Edit View Help
GNU nano 8.6
/etc/resolv.conf~ *
nameserver 1.1.1.1
nameserver 8.8.8.8
```

This tells your system to use those servers to look up websites, which can speed up browsing and reduce delays.

After running a speed test, we found out the internet speed has improved.

SHARE 

Result ID 18458162871

 RESULTS SETTINGS DOWNLOAD Mbps

42.83

 UPLOAD Mbps

61.48

Ping ms

 7 63 20 Connections

Multi

 FirstNet UrbanXConnect

Johannesburg

[Change Server](#) GO LNX Solutions

102.212.63.248

RATE YOUR PROVIDER

LNX Solutions



Kali Linux 2025.3 XFCE — Troubleshooting Log

Issue	Description	Steps Taken	Outcome
Wireshark Not Launching	Wireshark fails to open from menu or terminal; no GUI appears.	Verified binary with which wireshark then exists Checked installed packages then only core libraries Installed full package: sudo apt install wireshark Allowed non-root capture Added user to wireshark group Logged out/in Launched from terminal	Wireshark launched successfully; packet capture confirmed. Issue resolved.
Slow Network Performance	Internet browsing and downloads are sluggish over Wi-Fi.	Ran ping google.com then high latency Checked signal strength then weak (~30%) Moved closer to router then improved signal Restarted NetworkManager Edited /etc/resolv.conf with fast DNS Reconnected and retested ping	Network speed normalized; browsing and downloads improved. Issue resolved.

What we have learned

System setup fundamentals: Installing Kali Linux 2025.3 reinforced our ability to configure an operating system from scratch, manage user accounts, and apply permission structures that balance usability with security.

Networking essentials: Setting up a local network taught us how to assign IPs both manually and via DHCP, and how diagnostic tools like ping, tracert, and ifconfig provide quick visibility into connectivity issues.

Server and services deployment: Configuring Apache and enabling file sharing gave us practical insight into hosting local resources, and highlighted the importance of documenting service configurations for future troubleshooting.

Security best practices: Enabling firewalls, changing default passwords, and setting up antivirus emphasized the critical role of proactive defense in IT support. We learned that even small oversights (like leaving defaults unchanged) can create vulnerabilities.

Troubleshooting methodology: Simulating issues such as network slowness and offline devices helped us practice structured problem-solving: identify, test, resolve, and document. This reinforced the value of clear logs for accountability and knowledge transfer.

Documentation discipline: Creating screenshots, diagrams, and logs showed us how vital clear communication is in IT support. Technical fixes are only half the job — the other half is making sure others can replicate or understand them

What We'd Do Differently Next Time

Plan documentation earlier: Instead of documenting at the end, we'd integrate notes and screenshots throughout the process to save time and ensure accuracy.

Automate repetitive tasks: Using scripts for user account creation or network diagnostics would streamline setup and reduce human error.

Test under load: We focused on basic functionality, but next time we'd simulate heavier traffic or multiple users to see how the system performs under stress.

Expand security layers: Beyond firewalls and antivirus, we'd add intrusion detection tools or enforce stricter password policies to deepen resilience.

Collaborate more actively: Assigning roles (setup, networking, documentation) could improve efficiency and mirror real-world IT team dynamics.

Reflect on user experience: We'd spend more time thinking from the end-user's perspective — ensuring not just technical correctness but ease of use and clarity