TE

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

Operating system Configuration

Desktop: Asus AIO

Operating system: kali Linux 2025.3

Processor: 12th gen intel(R)

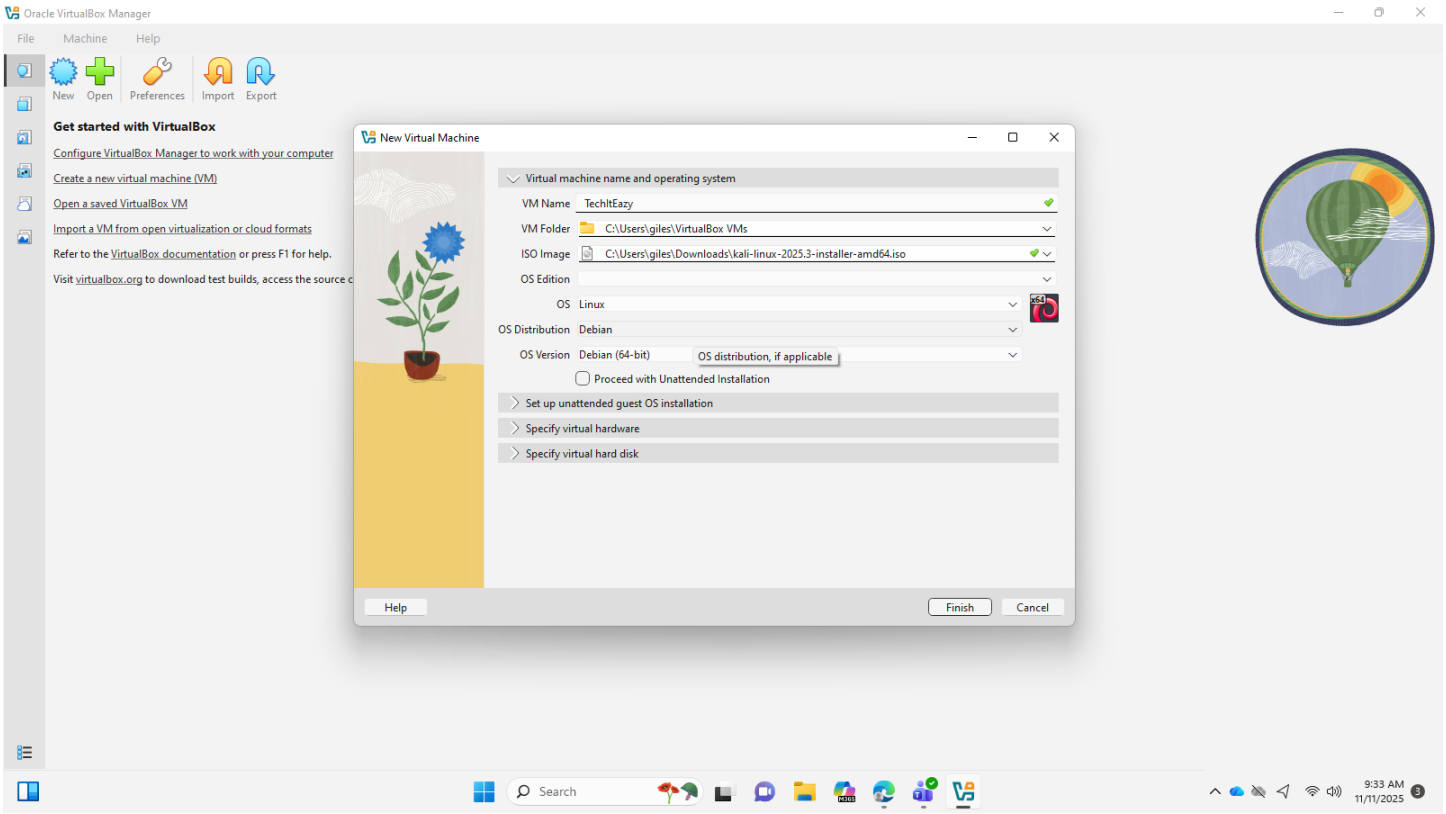
Installed ram: 8.00 GB

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.

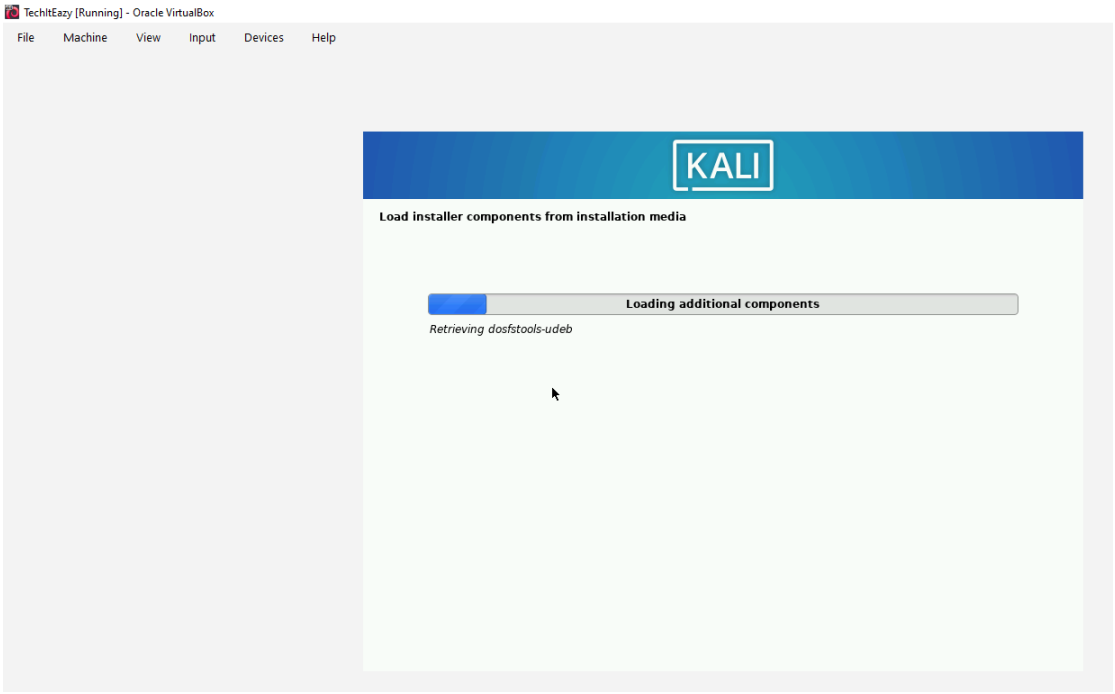
Kali Linux configuration

We used VirtualBox to install the kali Linux 2025.3 operating system

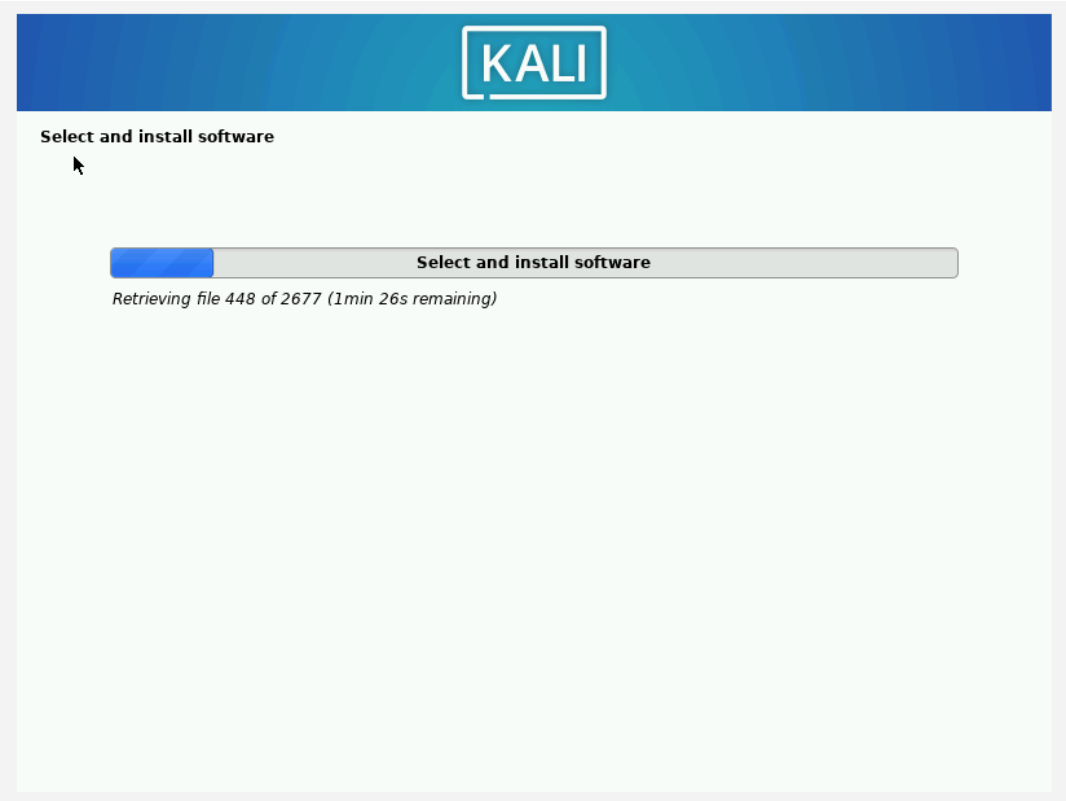


Os image installation

installing the OS using the graphical user interface

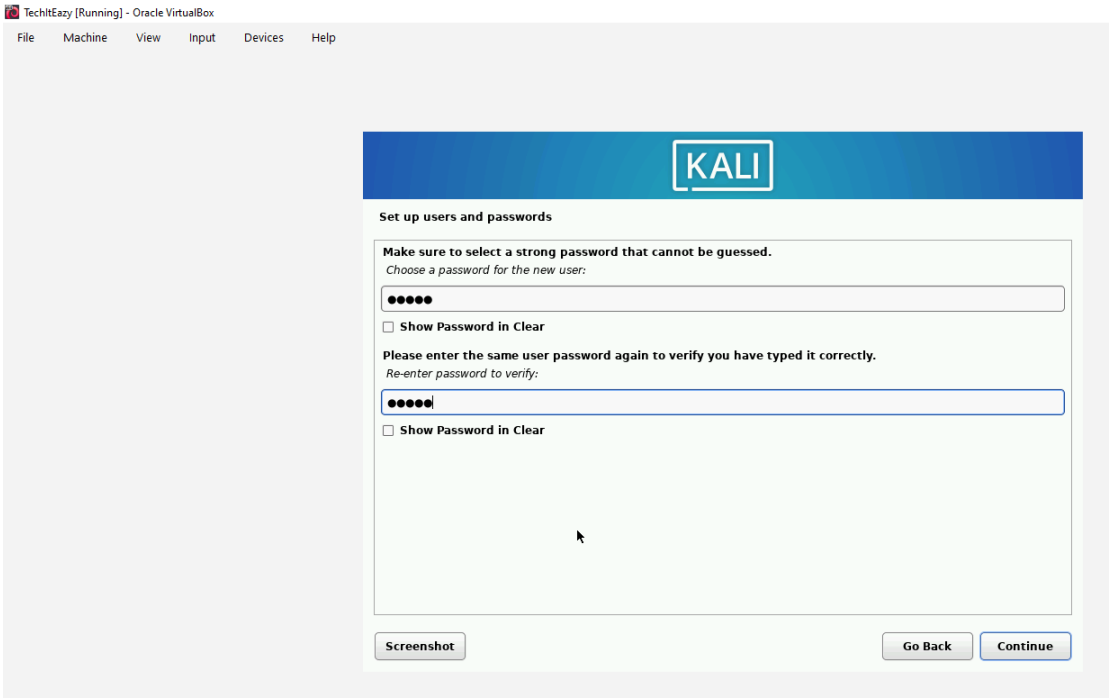


Installing additional system components and software packages



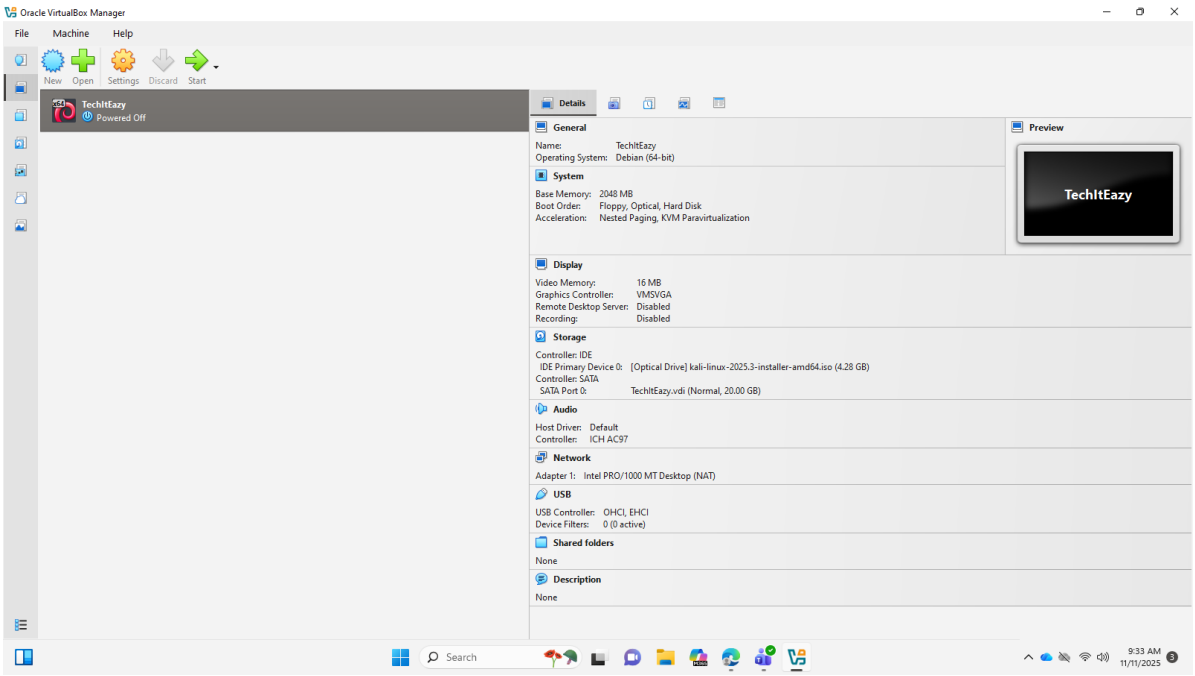
password

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

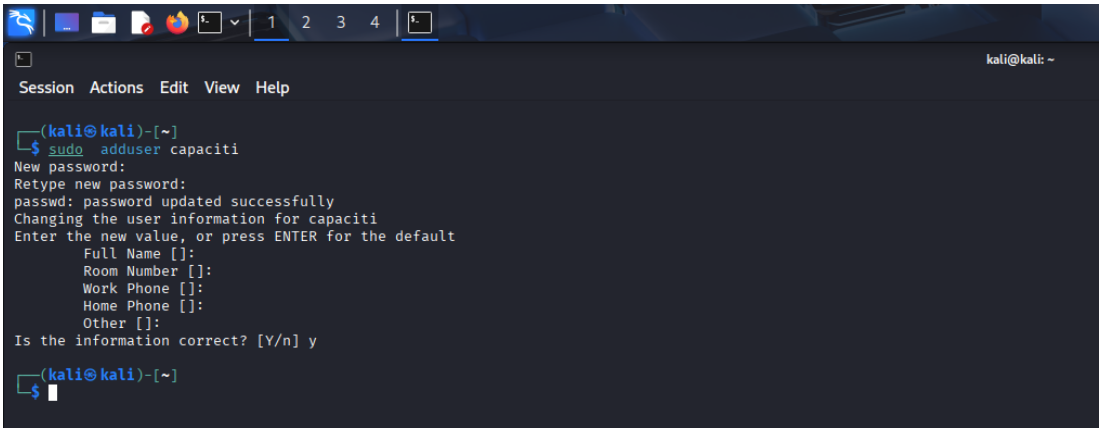
Using kali Linux 2025.3

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:

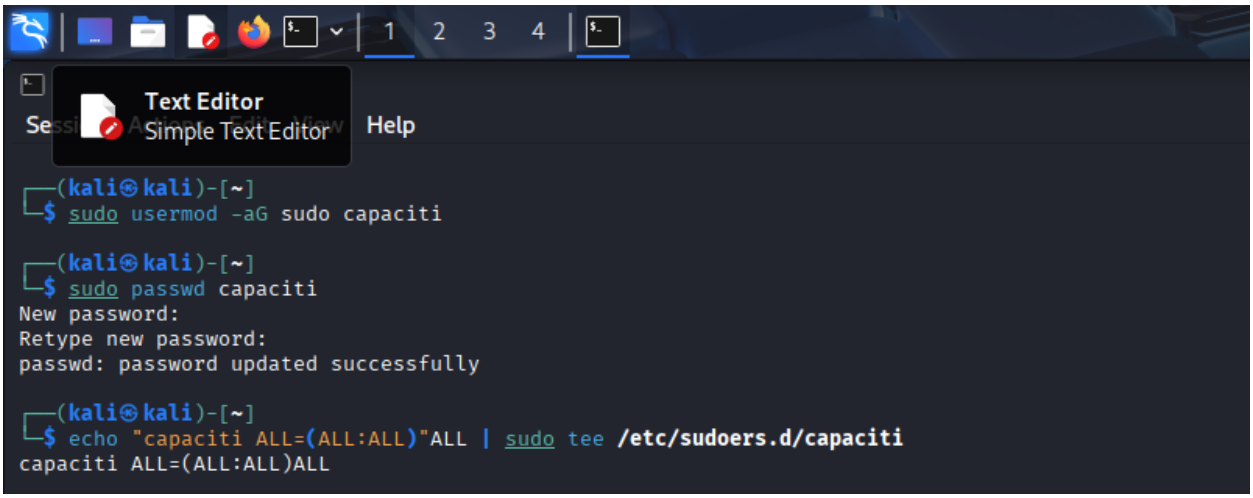
- Creates the user in the /etc/passwd and /etc/shadow databases.
- Prompts you to set a secure password for the account.
- Creates a new home directory (/home/newusername).
- Copies default configuration files into the new home directory.



Grant Administrative Privileges

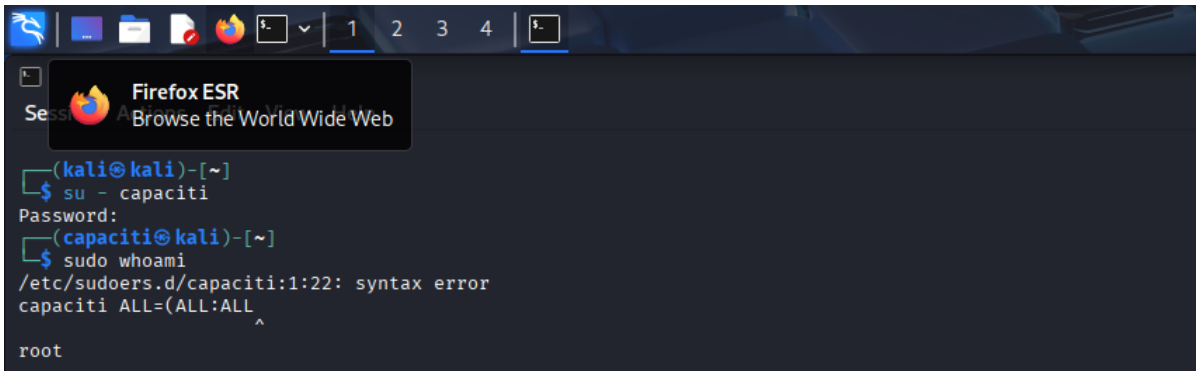
We then went on to grant administrative permissions (sudo permissions) to the capaciti user using the following command **sudo usermod -aG capaciti** this command modifies the user's group memberships to grant them administrative rights. The **sudo usermod -aG command** does the following:

- **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.



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.

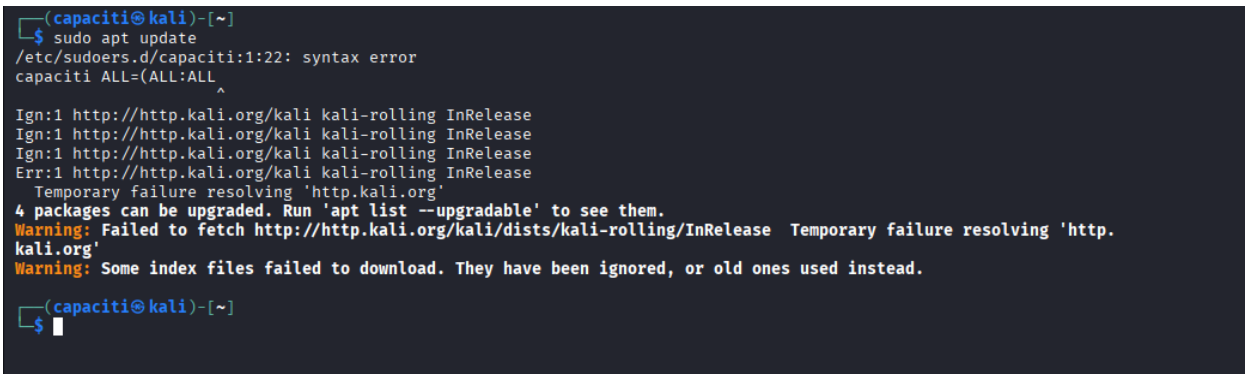


Validate Sudo Access

- 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.

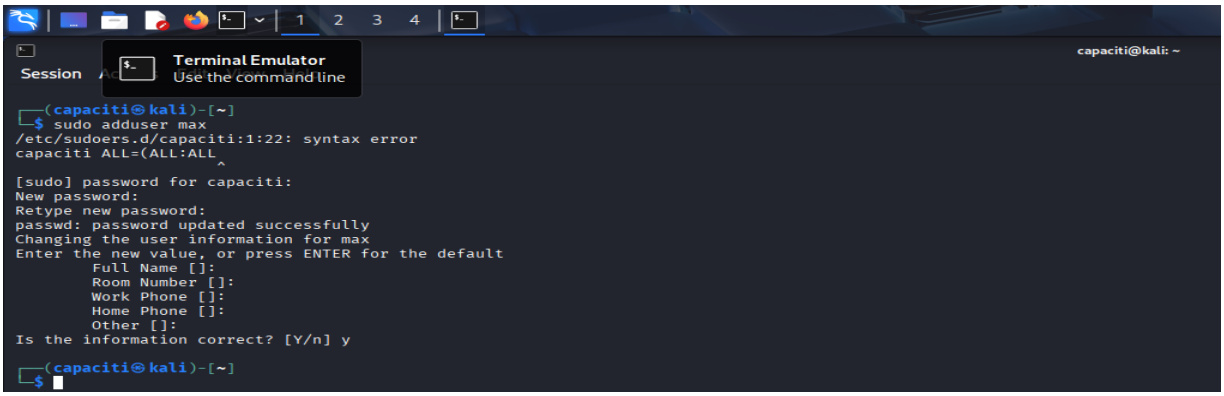
Testing user permissions

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



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



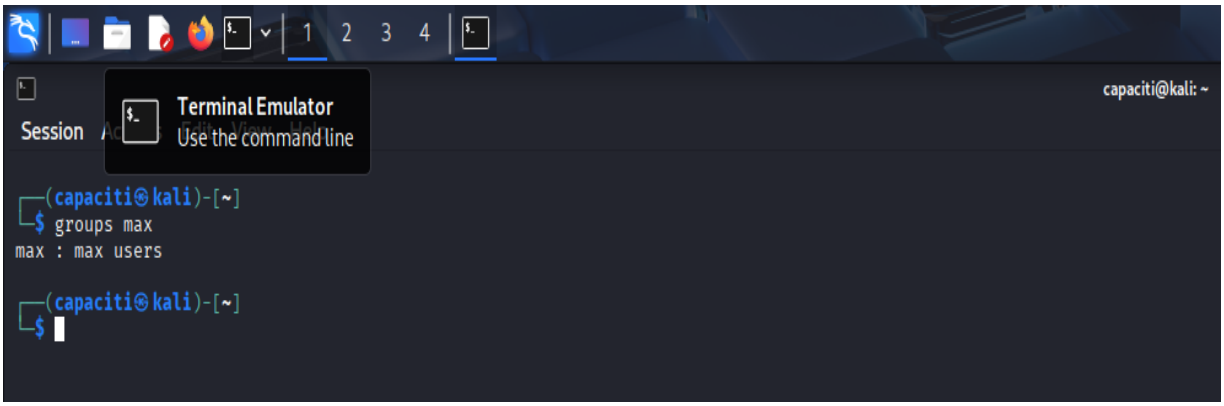
The purpose of the sudo adduser max command creates a new, standard user account, this standard user account doesn't have any sudo permissions.

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.

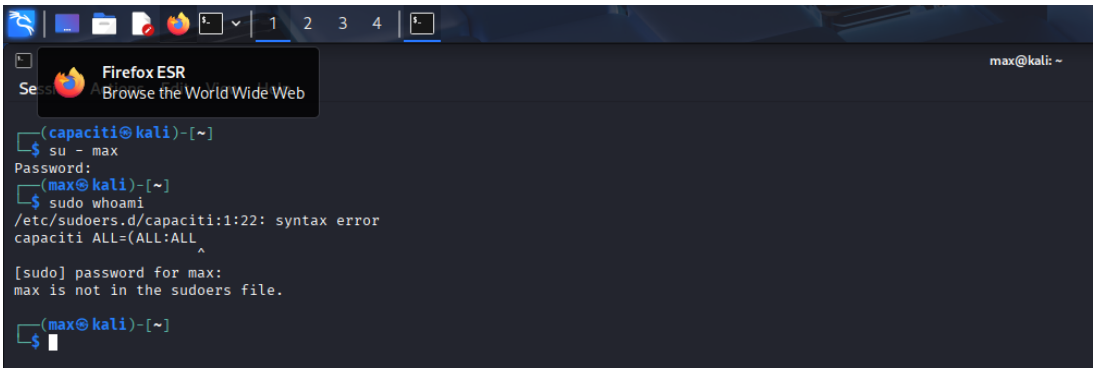
Verify Account Creation and Privileges

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.



Test the Account

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



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.

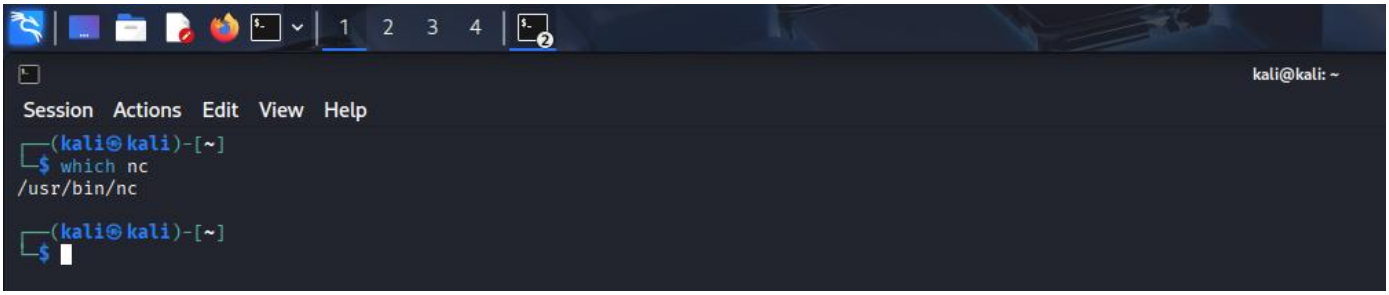
Networking

Virtual network

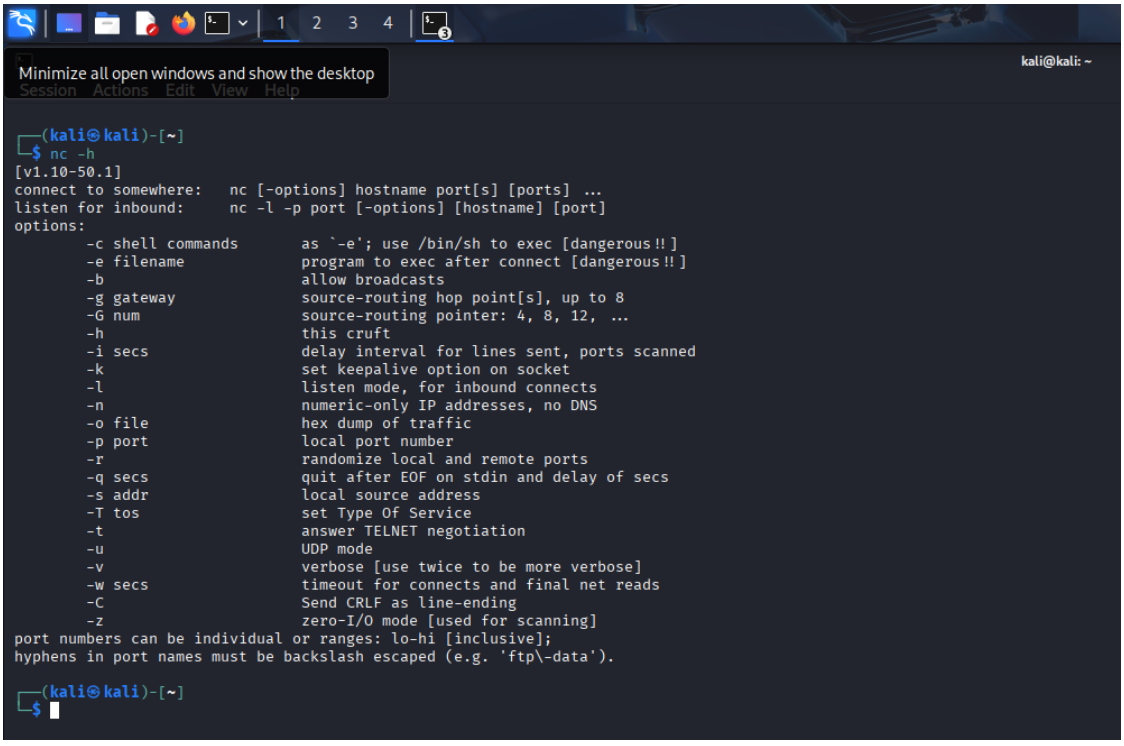
Netcat to send messages

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.

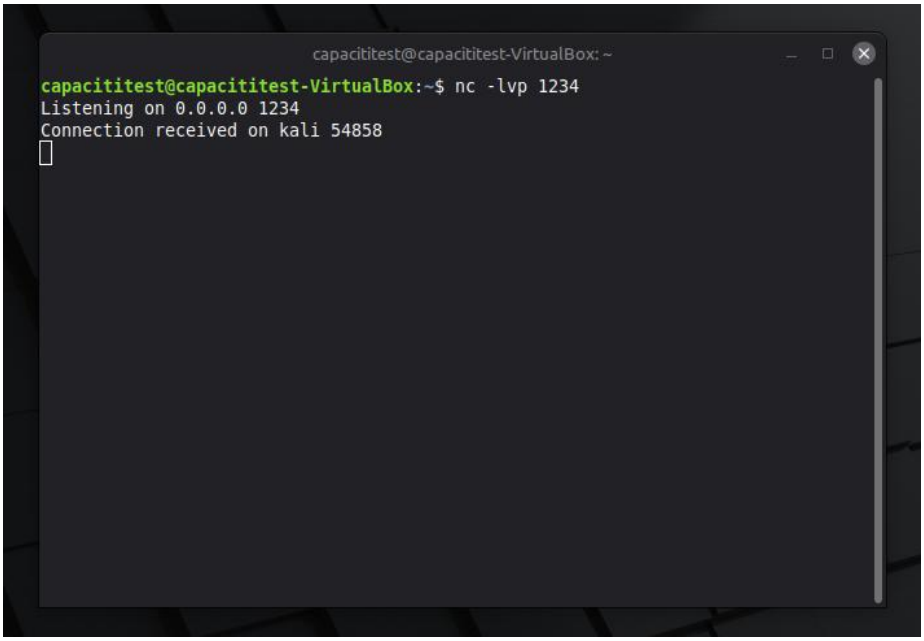


We then used the following command **nc -h** to confirm if Netcat is installed and working

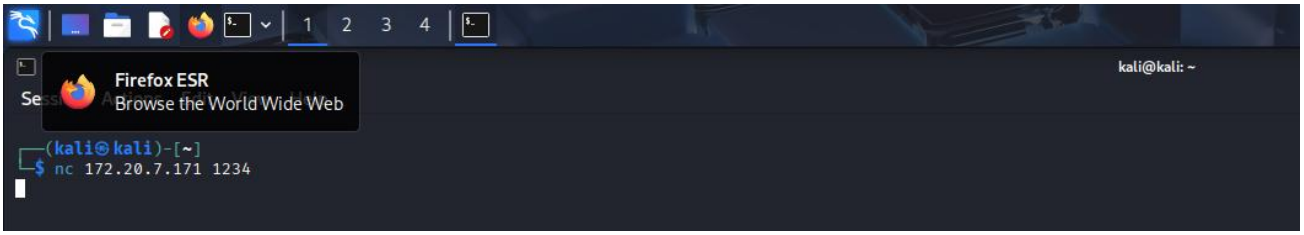


The results confirmed that Netcat is working.

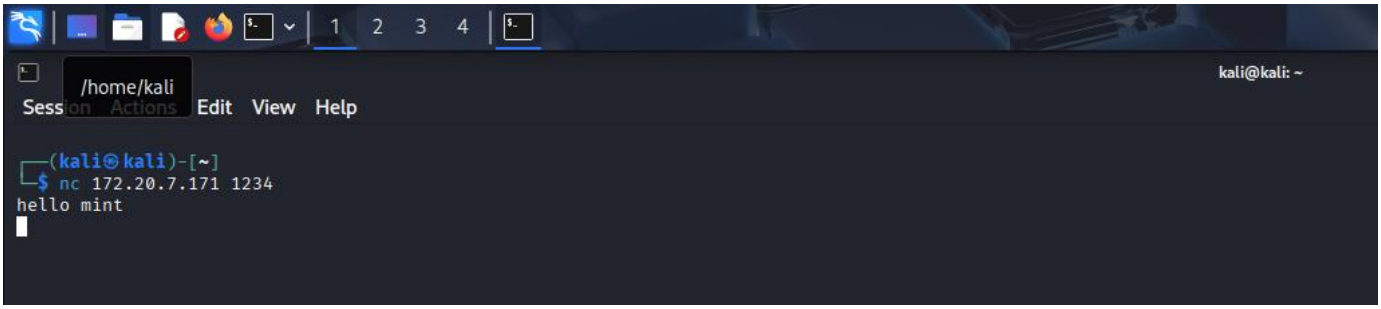
We then used the **nc -lvp 1234** on Linux mint using Netcat to start a server that listens for incoming connections on port 1234.



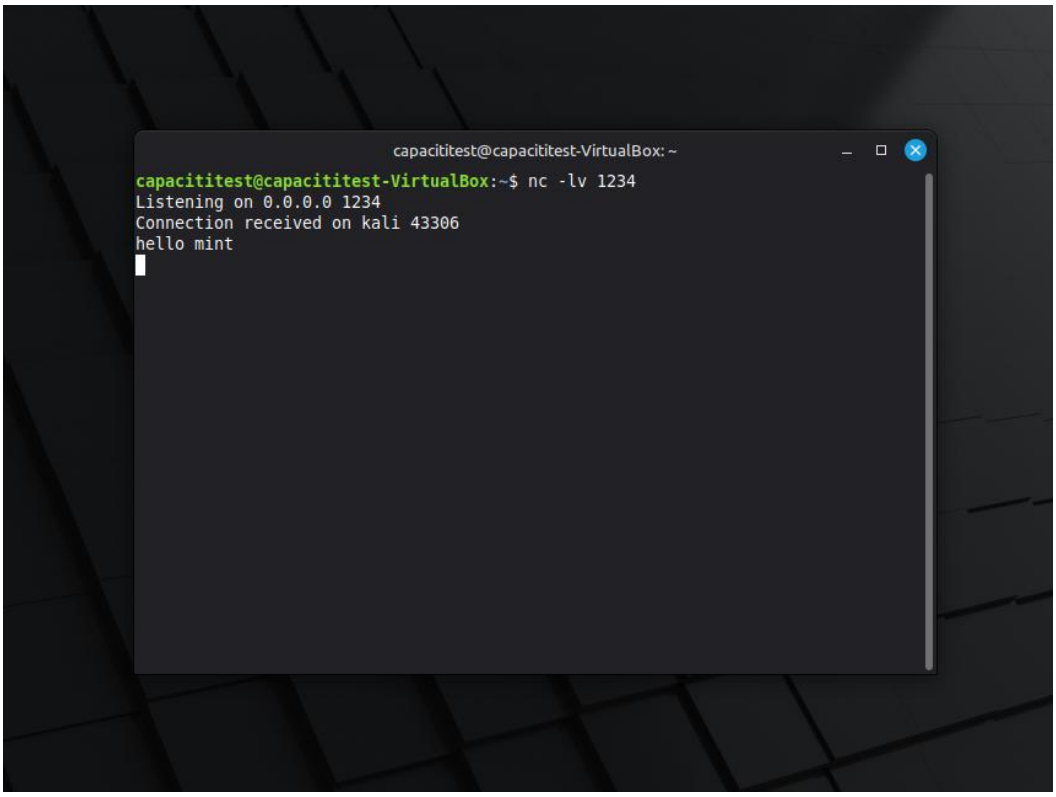
Communicating with Linux mint VM using the kali Linux virtual machine which was successful



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

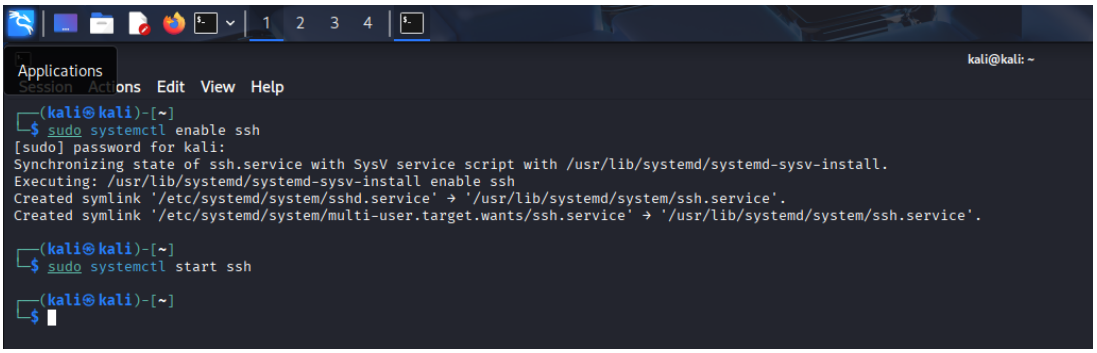


The message was successfully received by the Linux mint virtual machine.



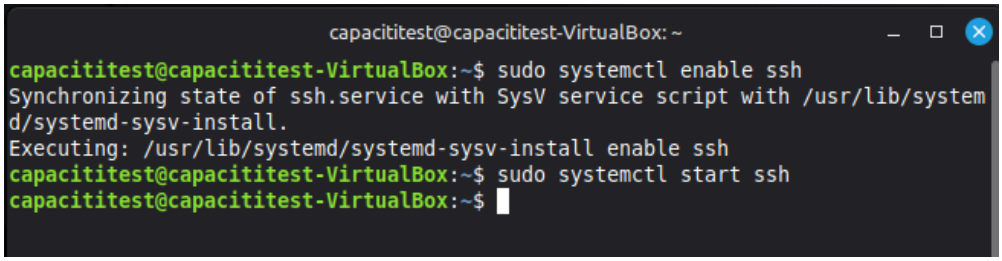
File transfer between two virtual machines

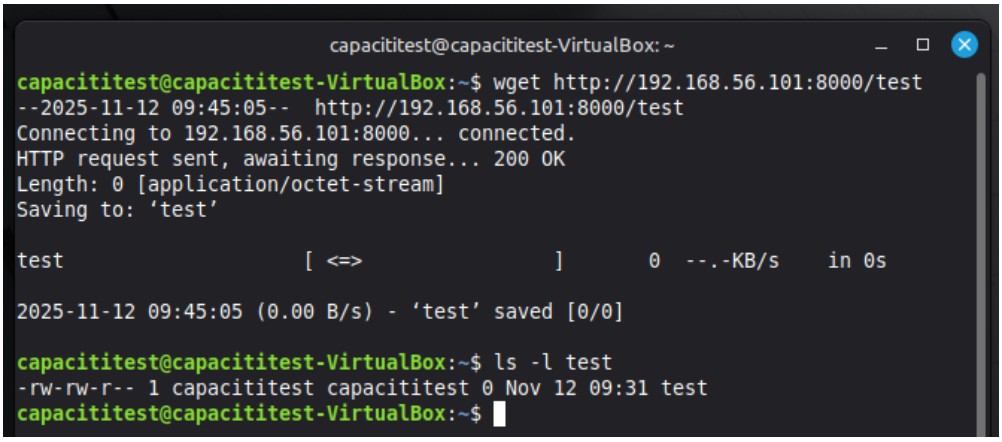
We sent a file from the kali virtual machine to the Linux mint virtual machine using ssh the name of the file is test. We used the following command to enable ssh on the virtual machine **sudo systemctl enable ssh** which enabled ssh and followed by the **sudo systemctl start ssh** which starts the ssh.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



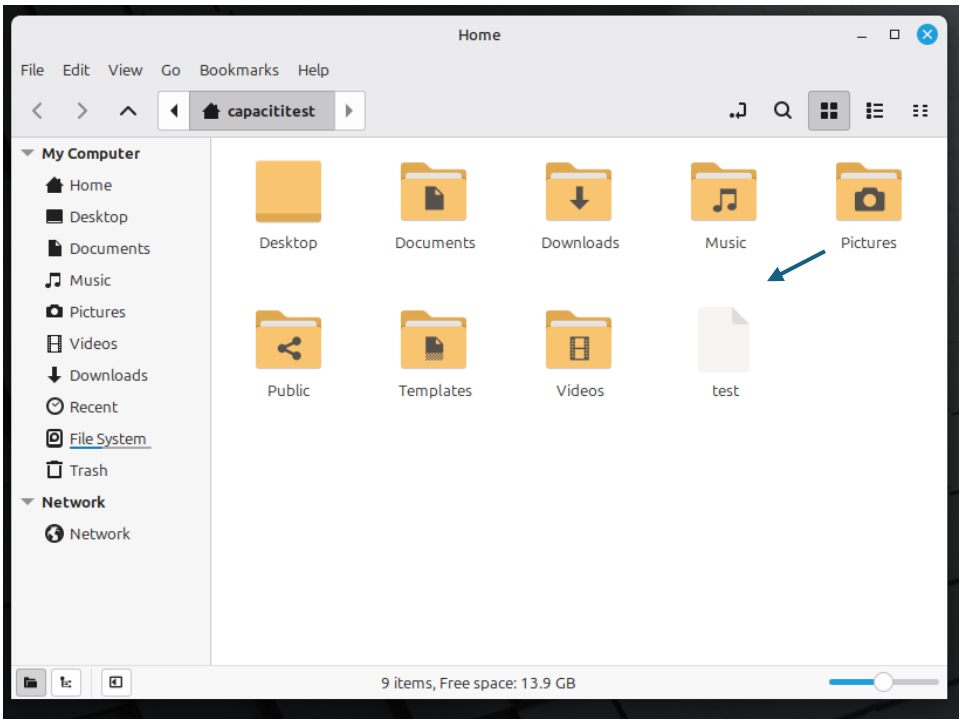
To transfer the file from the kali VM to the Linux mint VM we opened the terminal and navigated it to the downloads file using the following command **cd /home/kali/Downloads** and then we went on to start the python http server using the following command **python3 -m http.server 8000** which will be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

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.

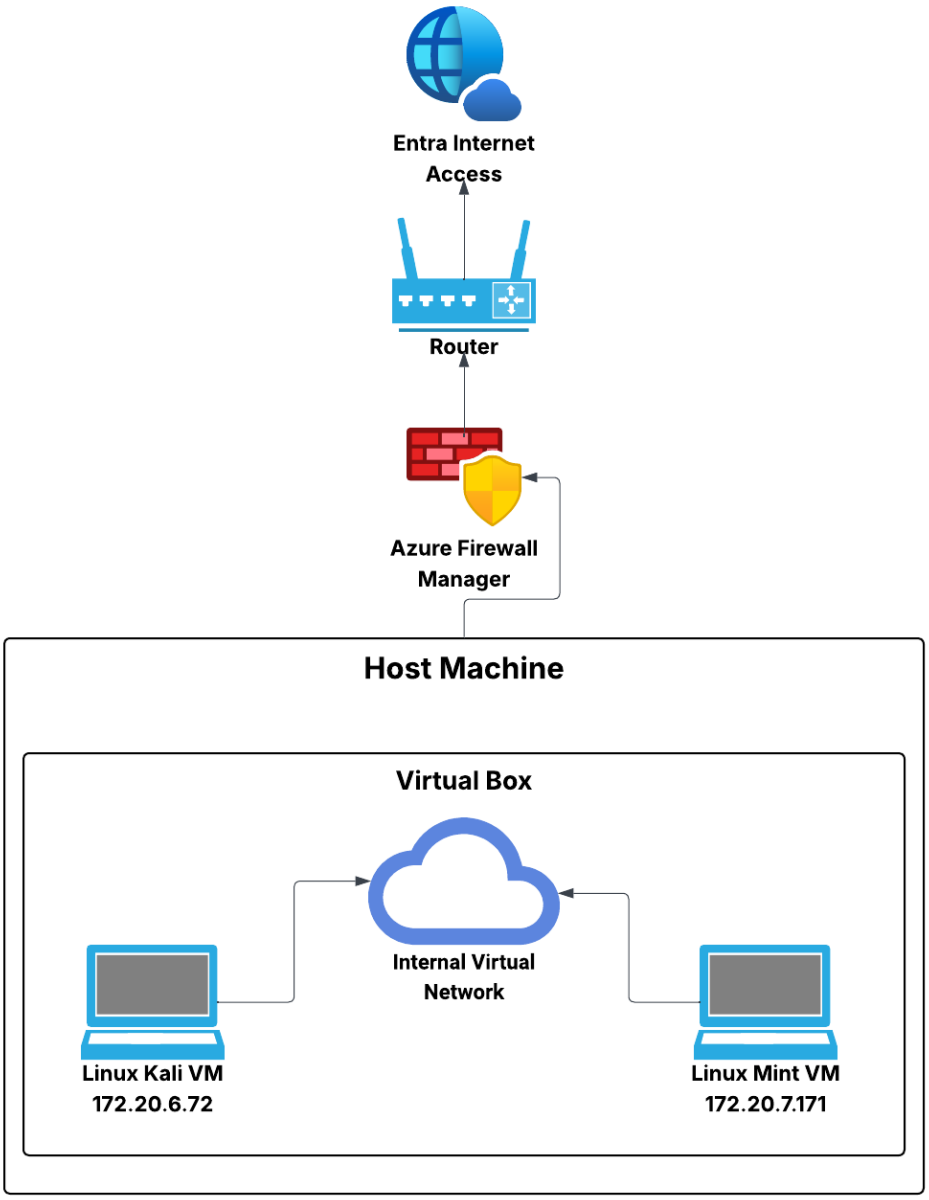




File has been successfully transferred to the mint virtual machine.



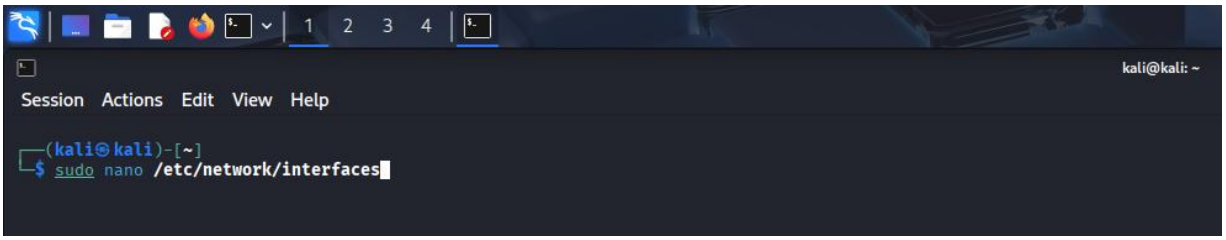
Network diagram



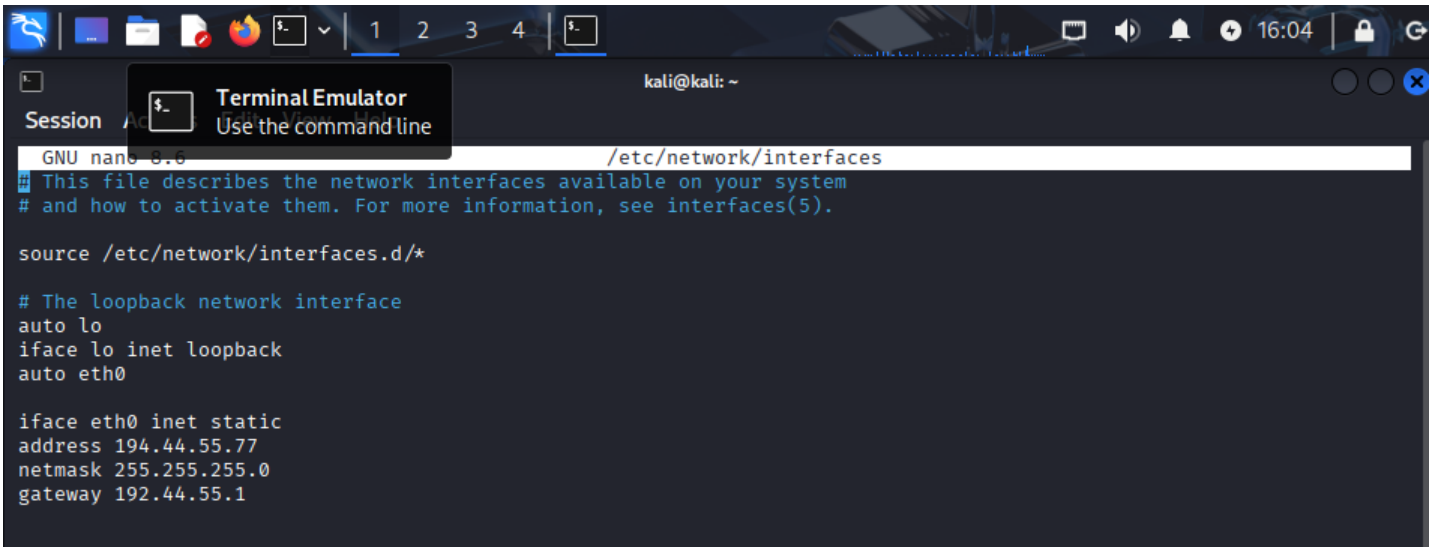
Basic local network setup

Assigning IP by editing interface file

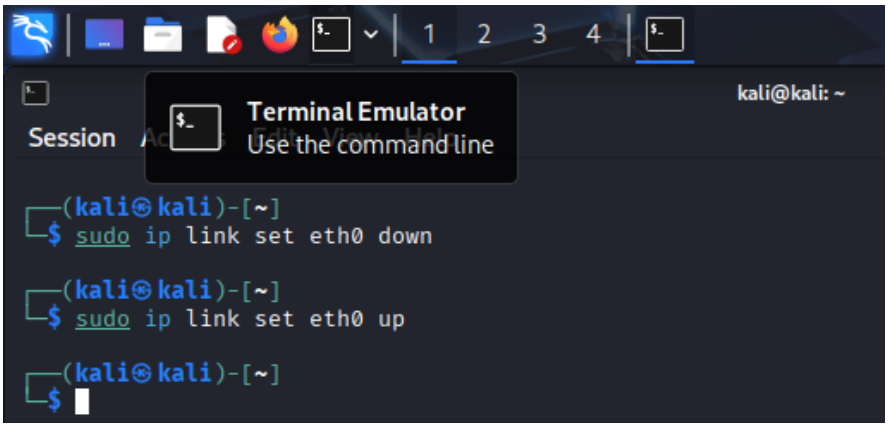
Using the **sudo nano /etc/network/interface** we opened the file and elevated the privileges



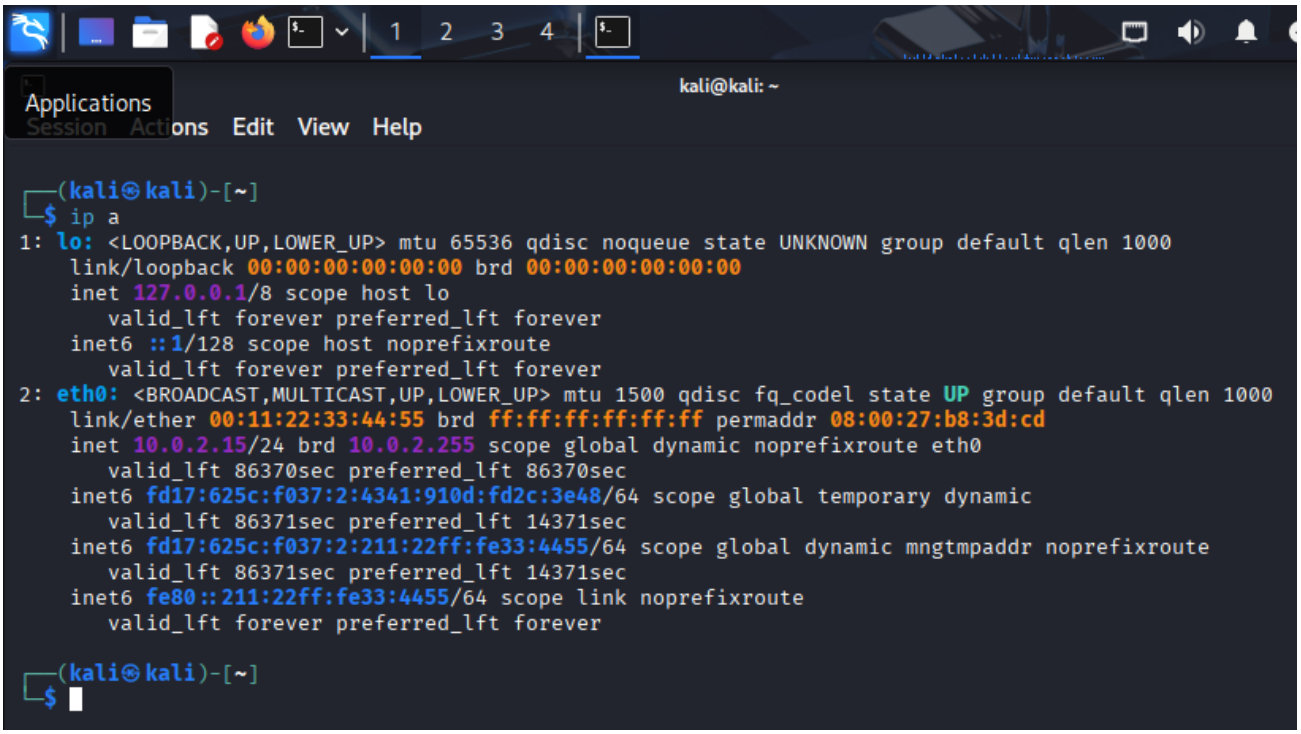
The command opens the Linux open-source GNU nano text editor where we a going to edit the interface document which we will be used to manually change the Ip address and net mask and then gateway.



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.



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.

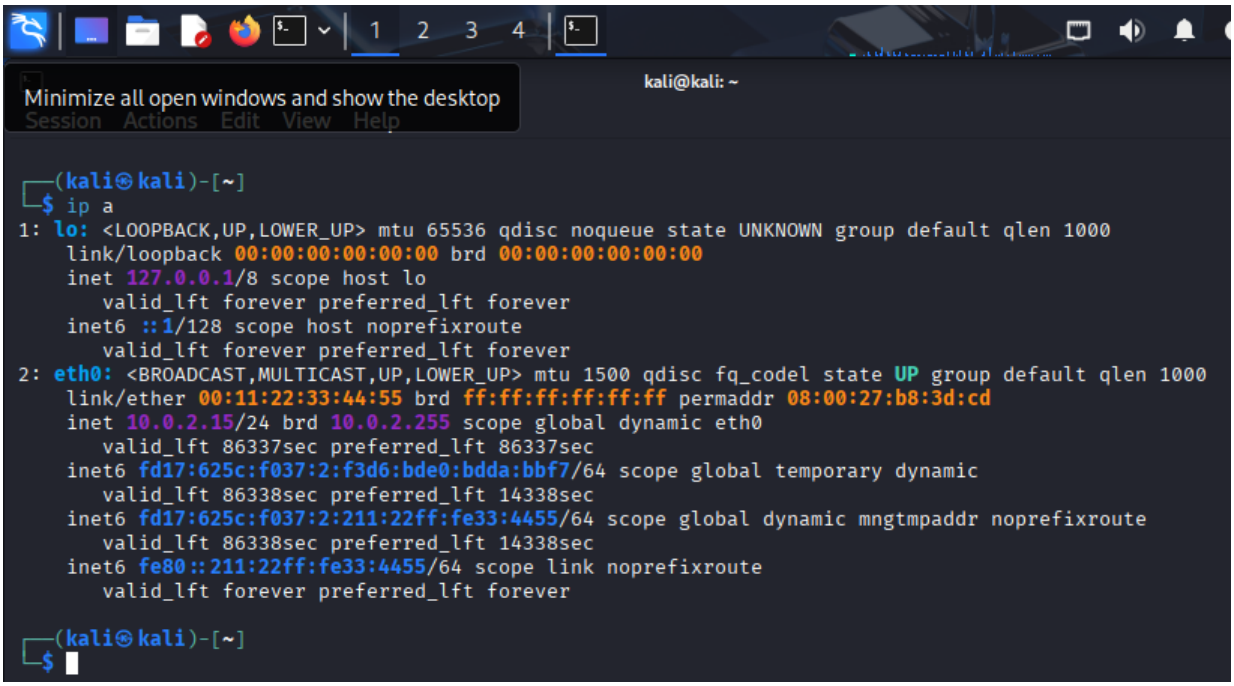


Assign IPs manually or using DHCP

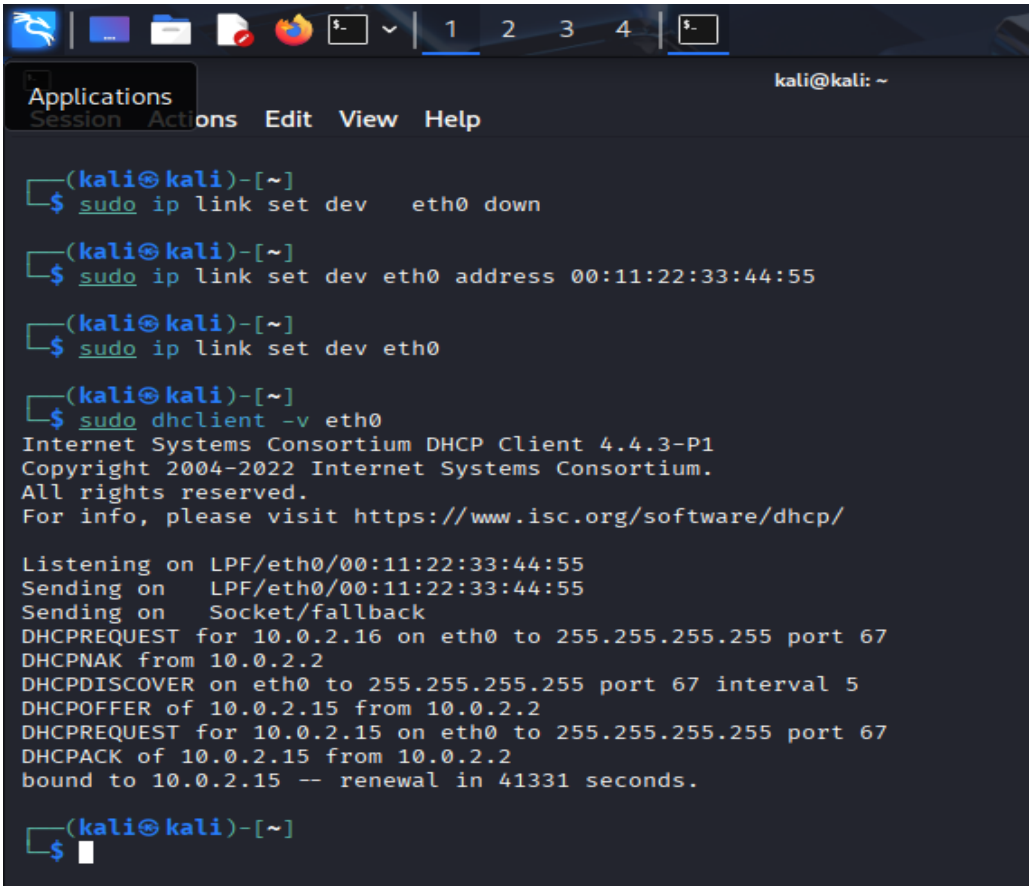
Mac address spoofing

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.

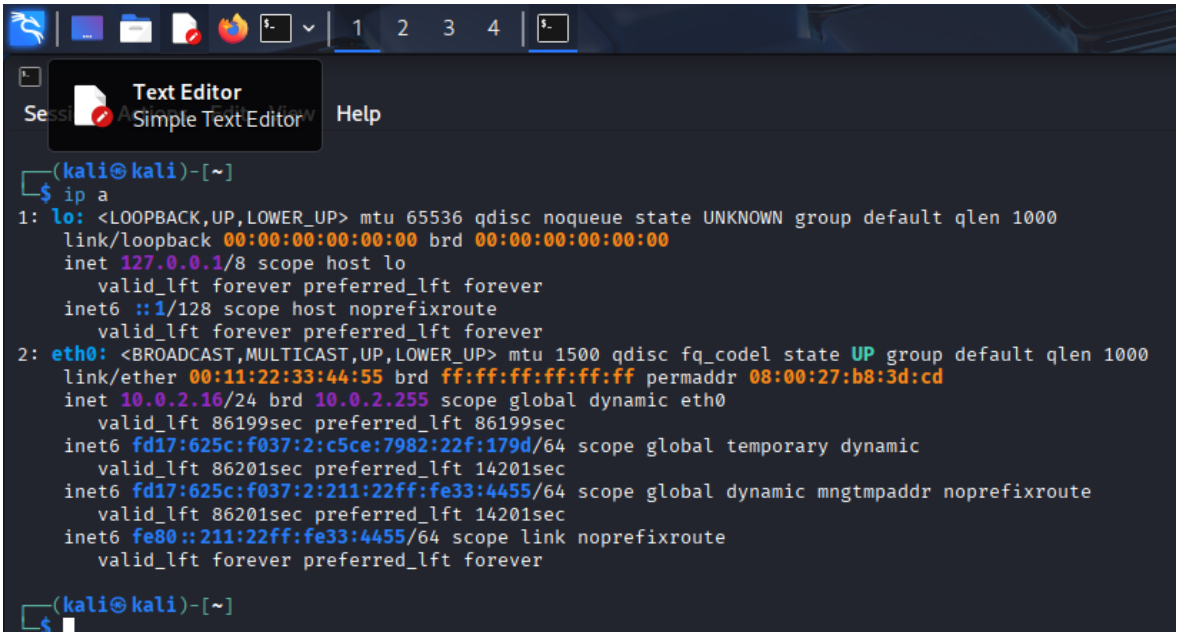
We started by updating the system packages our IP address before spoofing ways 10.0.2.15/24.



The 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.



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.

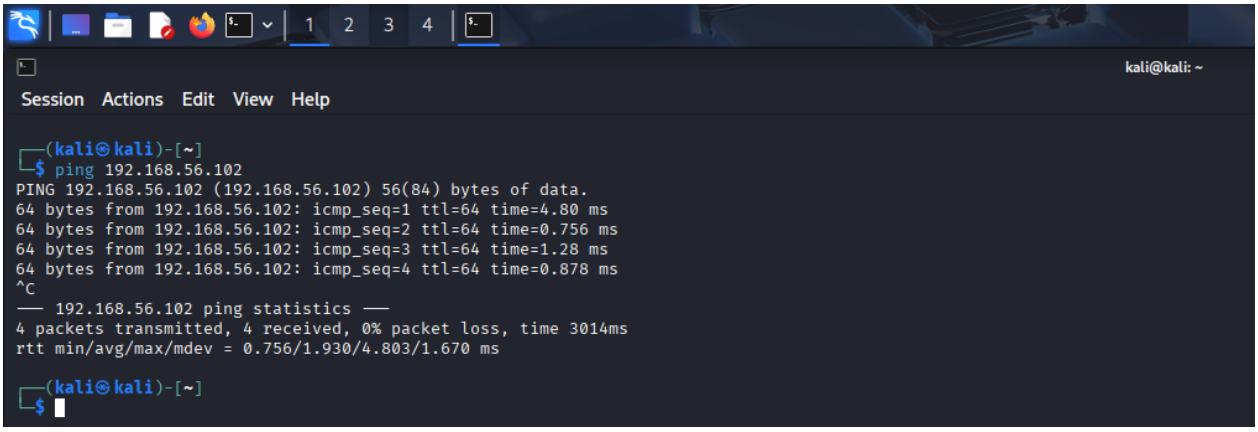


This method is useful in situations where you don’t have other kali utilities available.

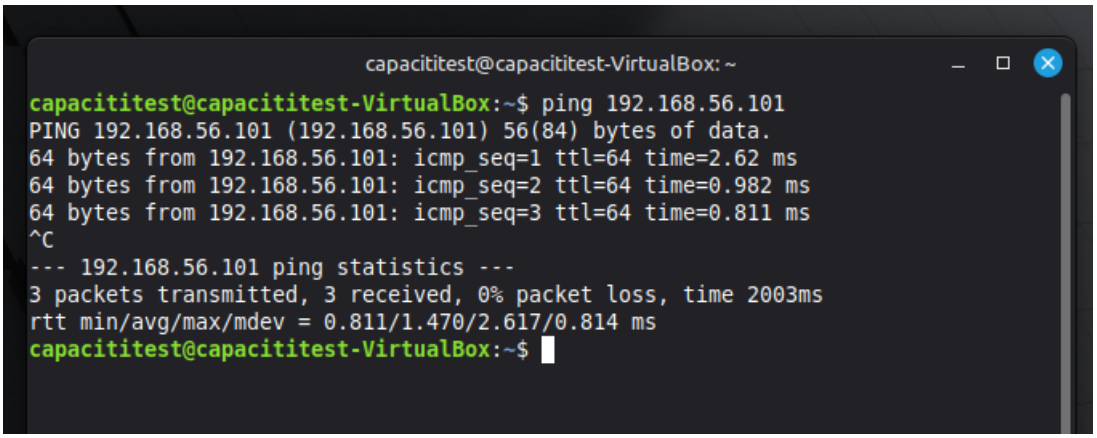
Basic configuration

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.

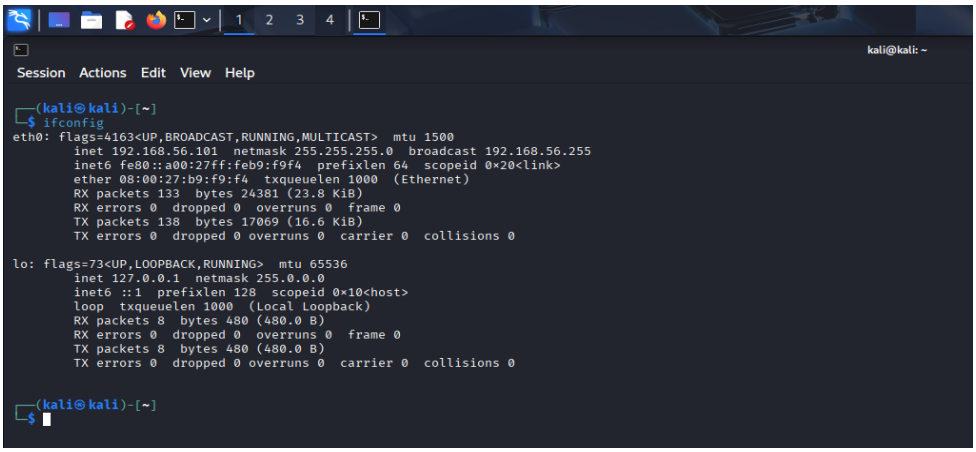


We then went on to ping the Linux mint virtual machine.

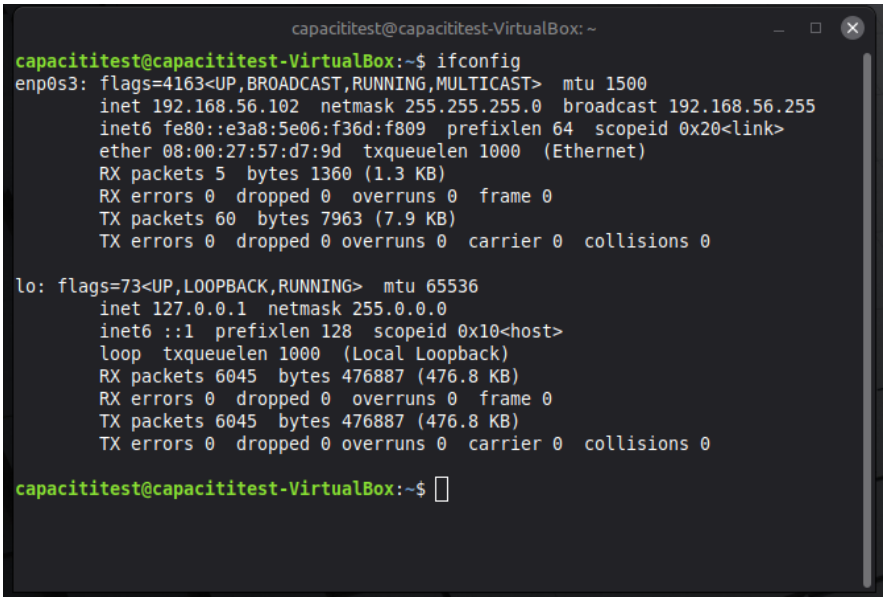


ifconfig

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

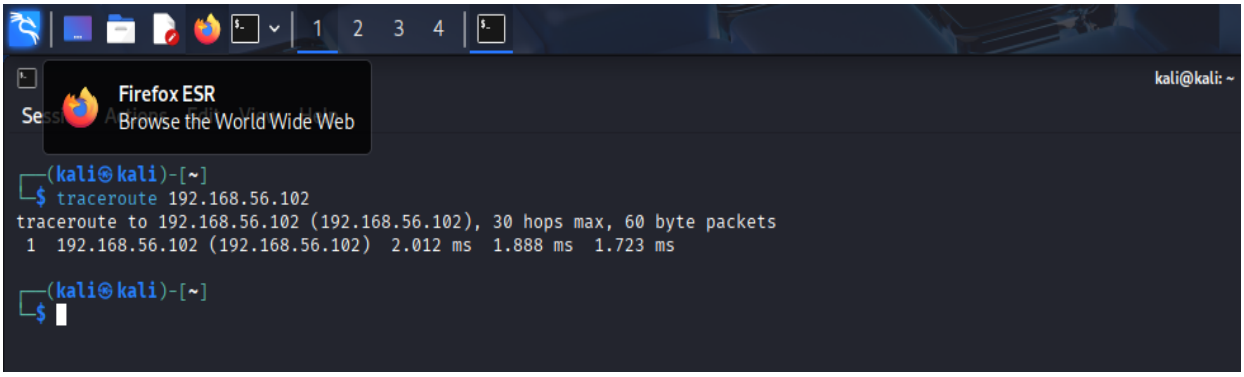


The ifconfig command in Kali Linux is used to display and configure network interfaces. It shows important details like IP address, MAC address, netmask, and interface status (up or down). You can use it to manually assign IP addresses, enable or disable interfaces, and troubleshoot network connectivity.



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.

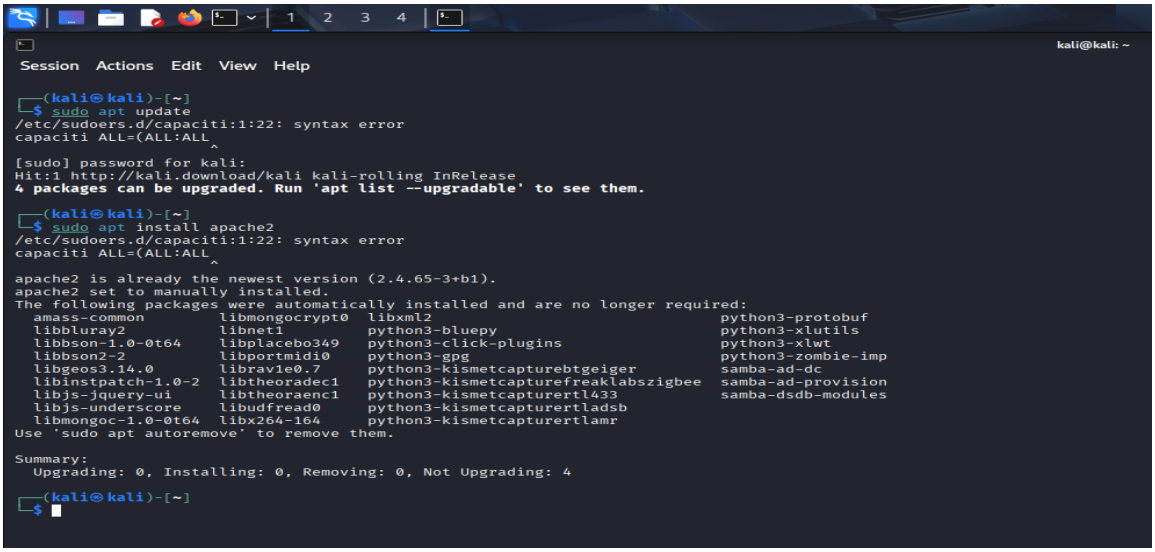


Server & Services

Server and layout

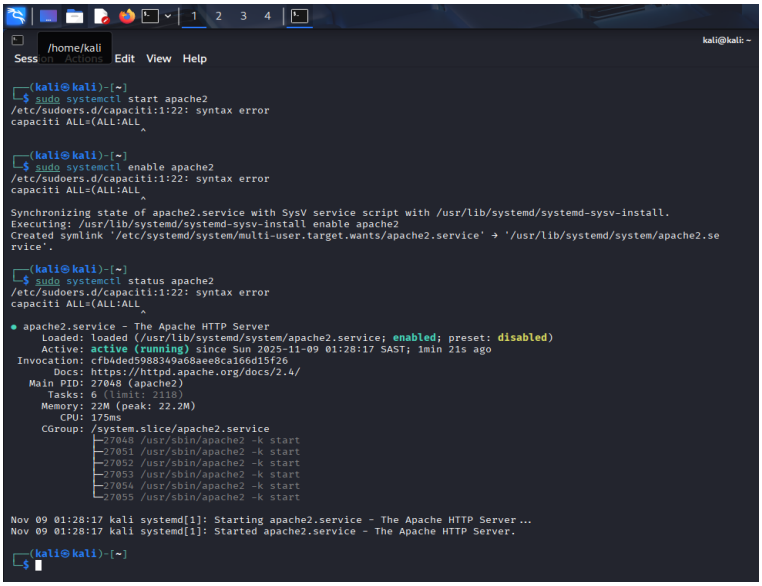
Installing the Apache server

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**



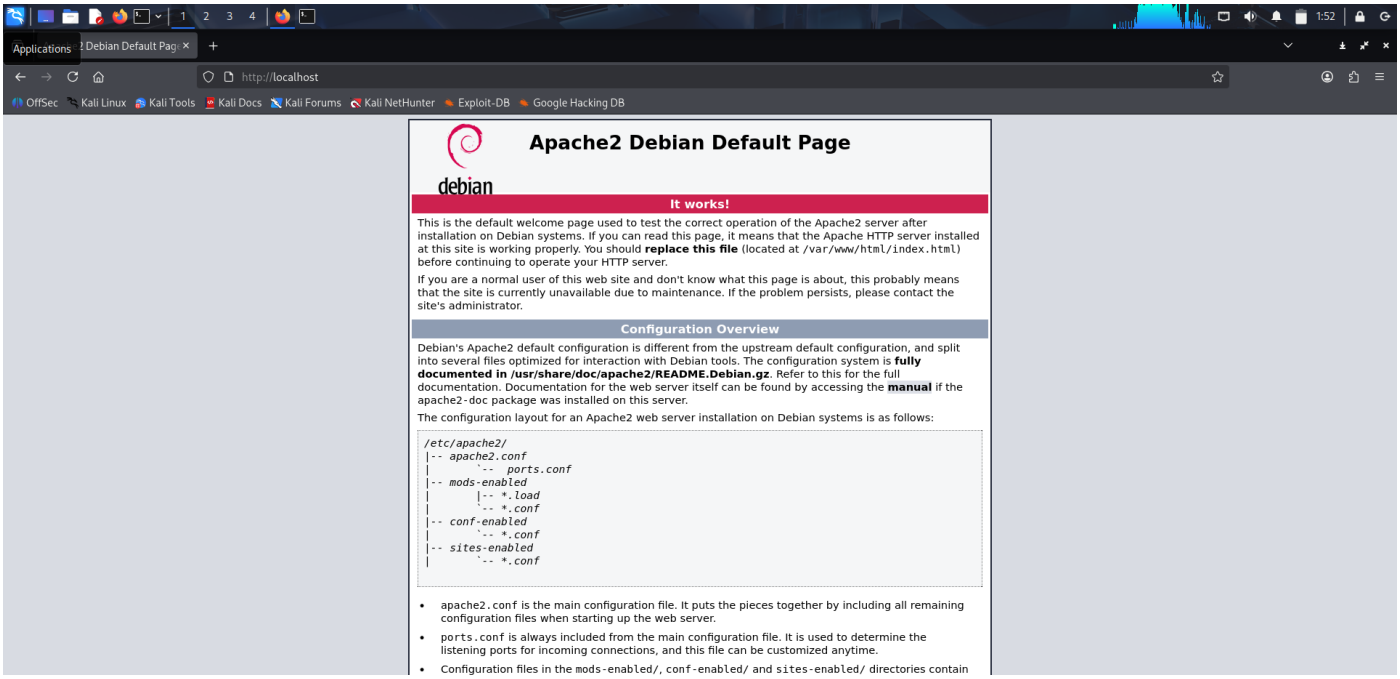
Starting and enabling Apache

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.



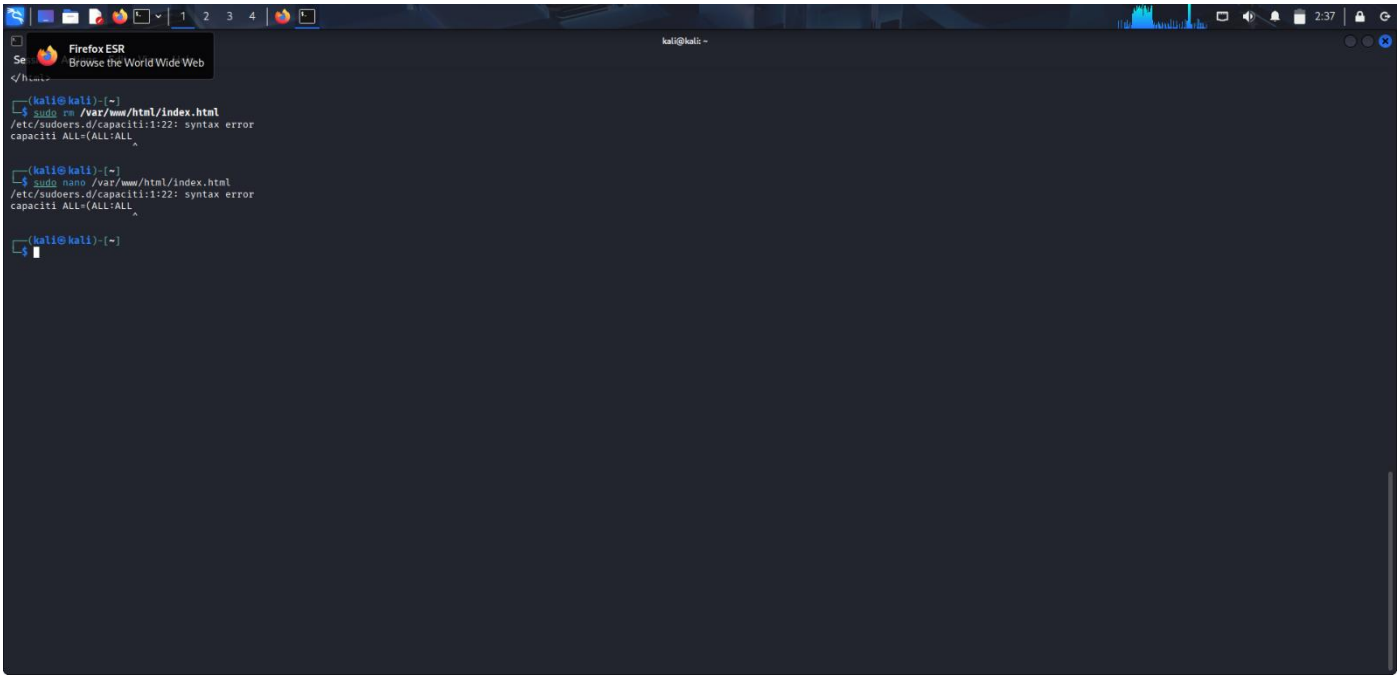
Testing Apache

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



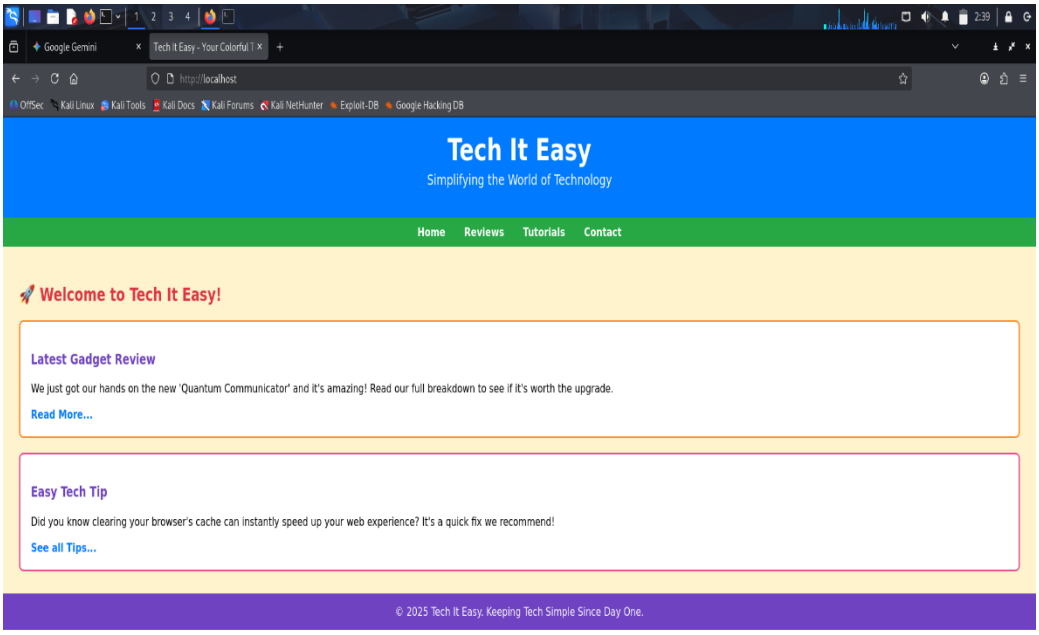
Custom intranet page

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.



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.



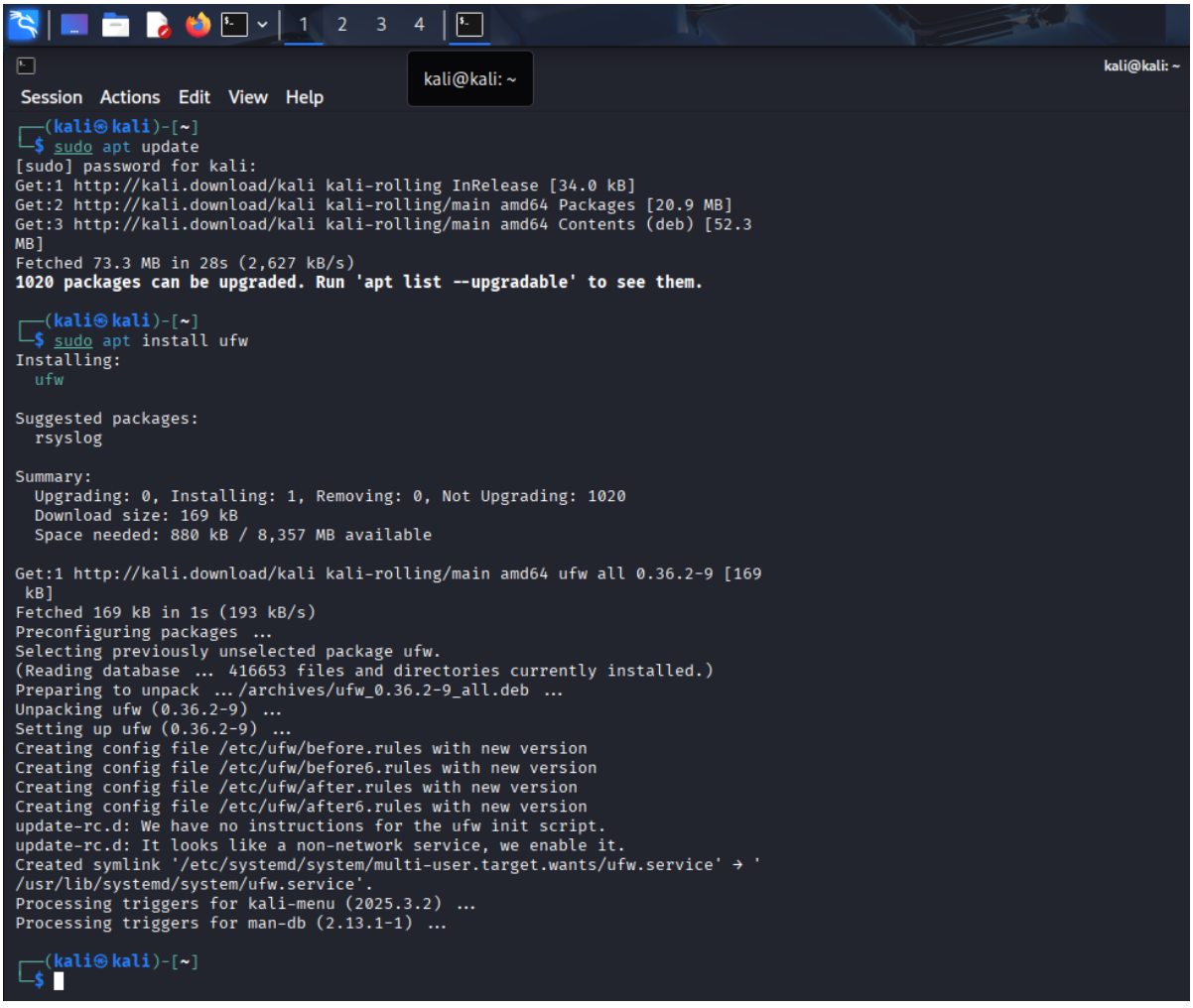
Security

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.

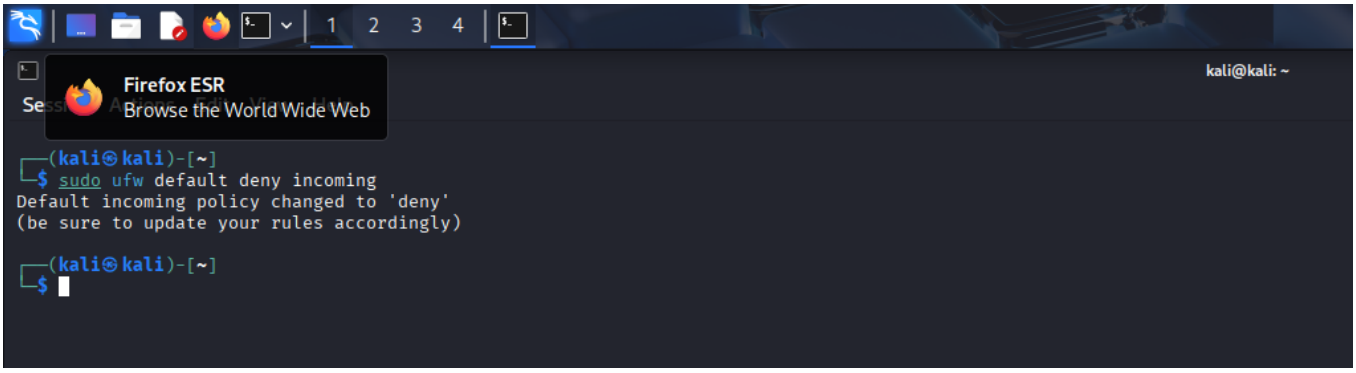
Installing ufw

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



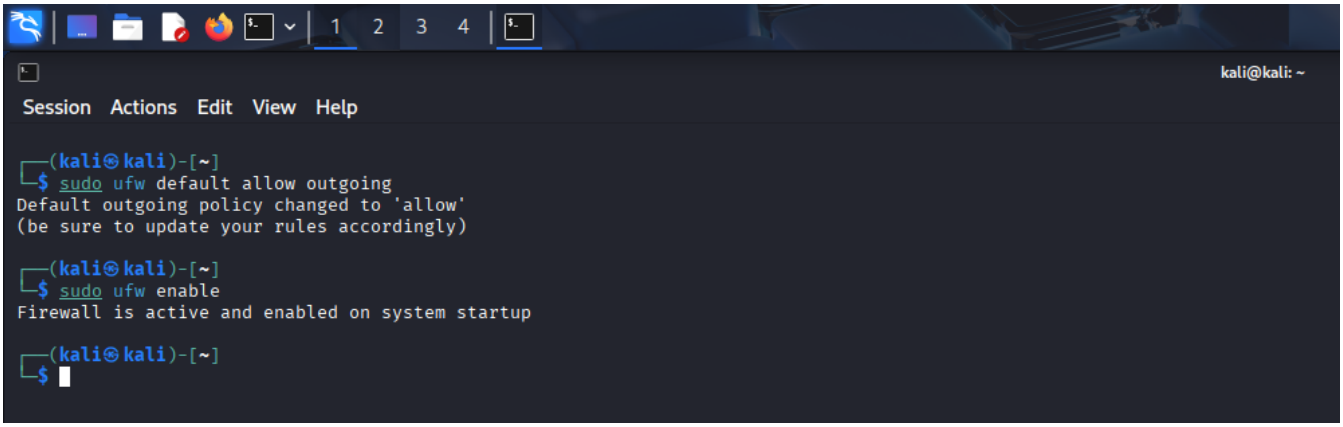
Enable UFW and set default rules

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.



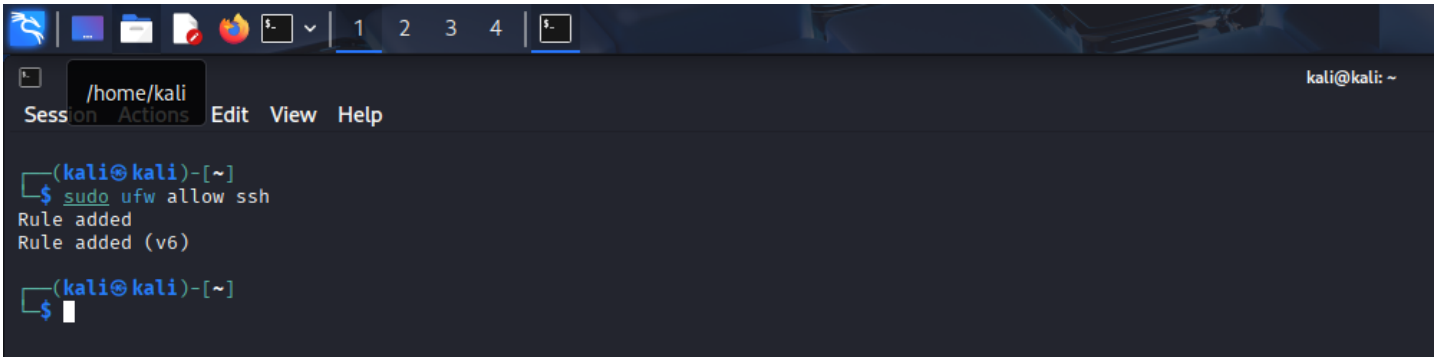
We then used the command **sudo ufw default allow outgoing** sets the default policy to allow all outgoing traffic from your system. This means your computer can initiate connections to other devices or servers (like browsing websites or sending emails) without restriction. It's a common setting because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



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.

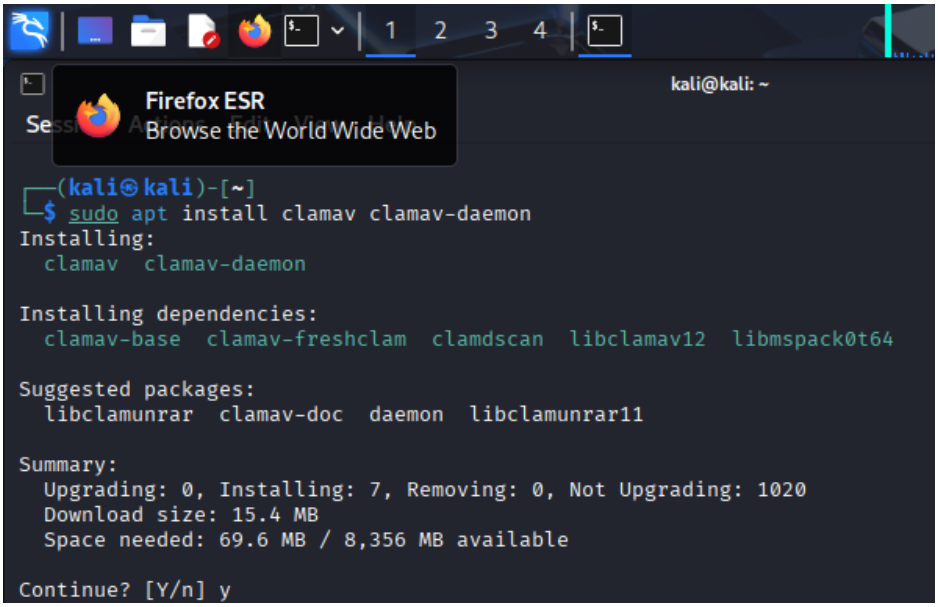


Checking status

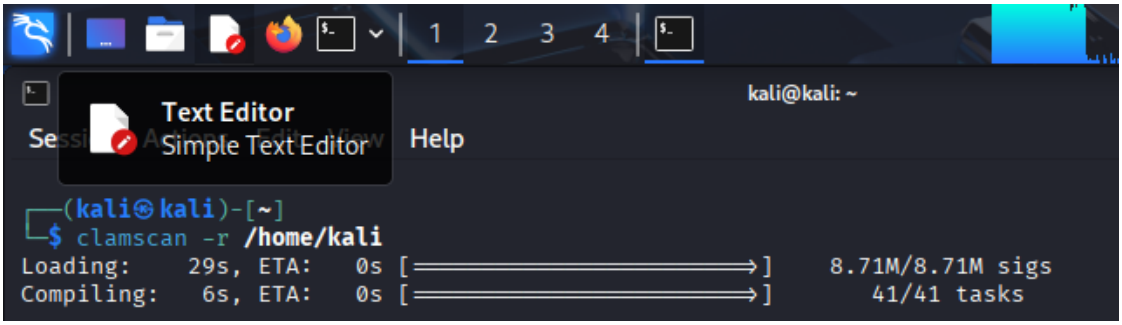
As a team, we ran the command **sudo ufw status verbose** to assess the current state of our system's firewall. This gave us a comprehensive overview of UFW's configuration, confirming whether it was active and detailing the default policies for incoming and outgoing traffic. We reviewed the list of 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.

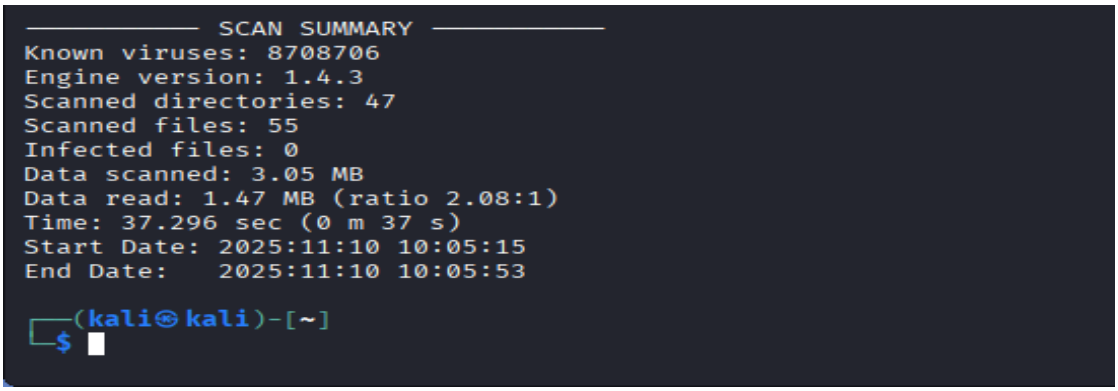
Set Up Basic Antivirus

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.

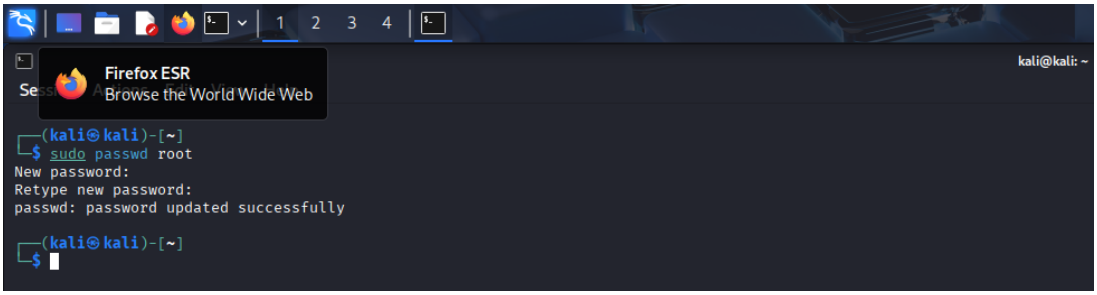


We then went forward with scanning the kali directory using the antivirus.

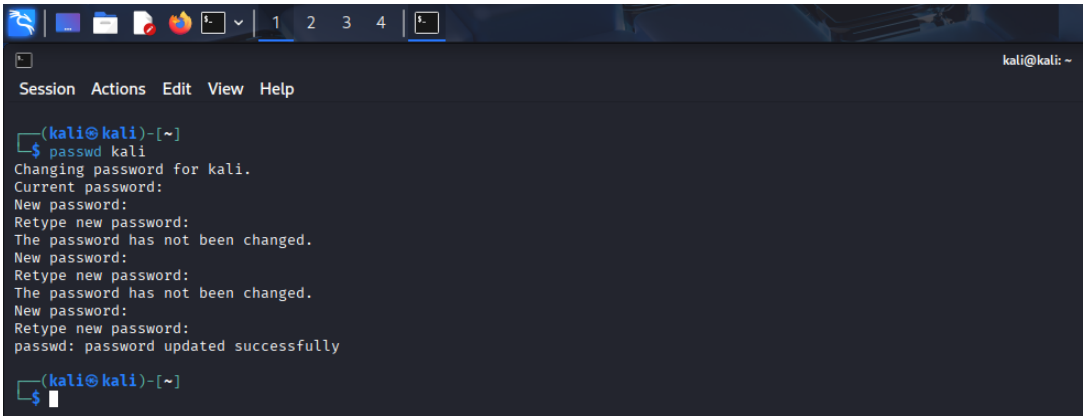




Change Default Passwords

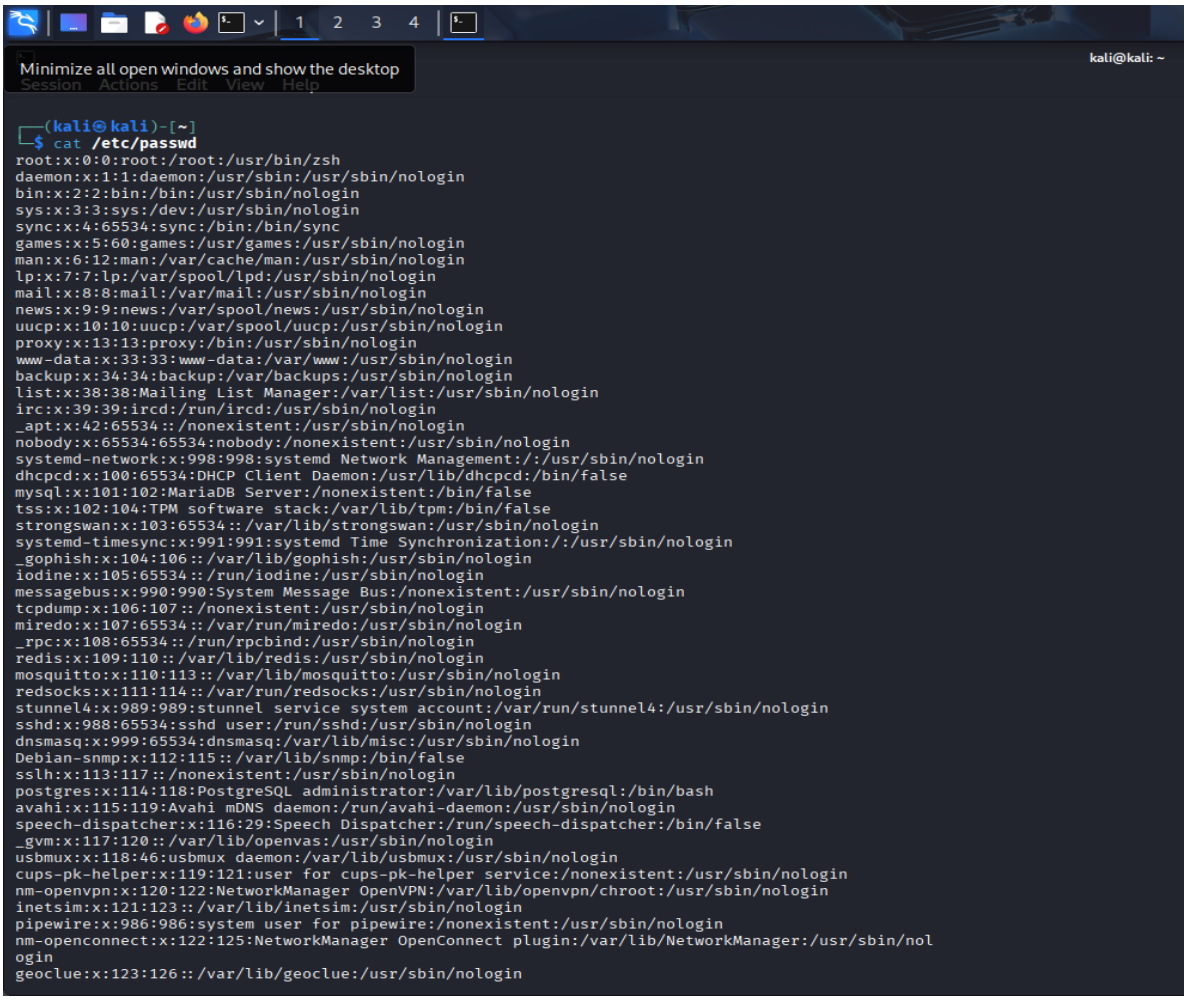


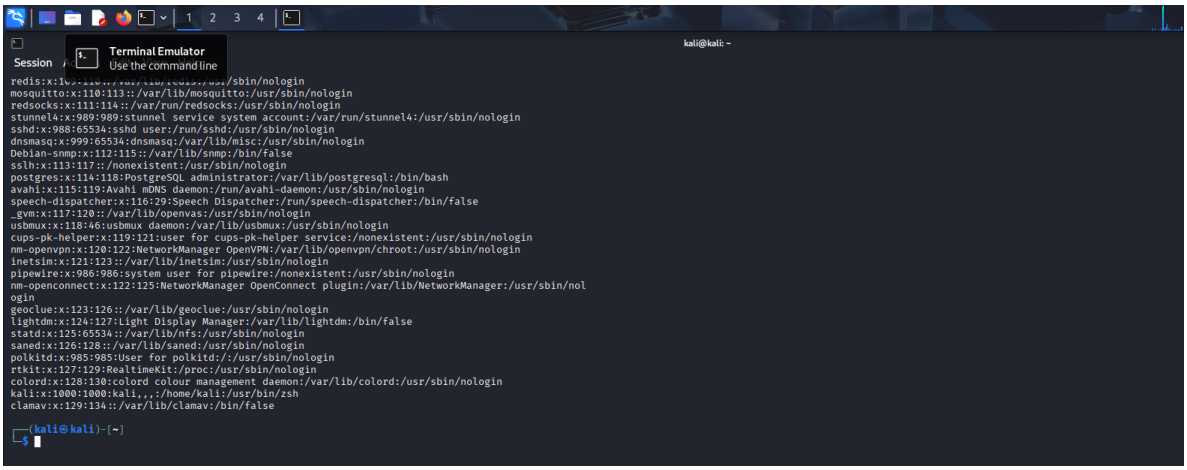
Change user password



Audit user accounts

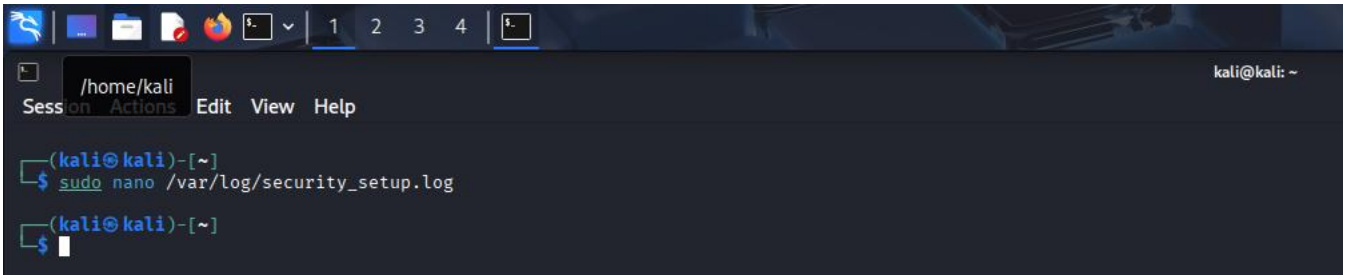
As a group, we audited user accounts on our system by reviewing the contents of /etc/passwd using the command **cat /etc/passwd**. This file lists all user accounts and system services, providing details such as usernames, user IDs (UIDs), group IDs (GIDs), home directories, and default shells. By examining 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.



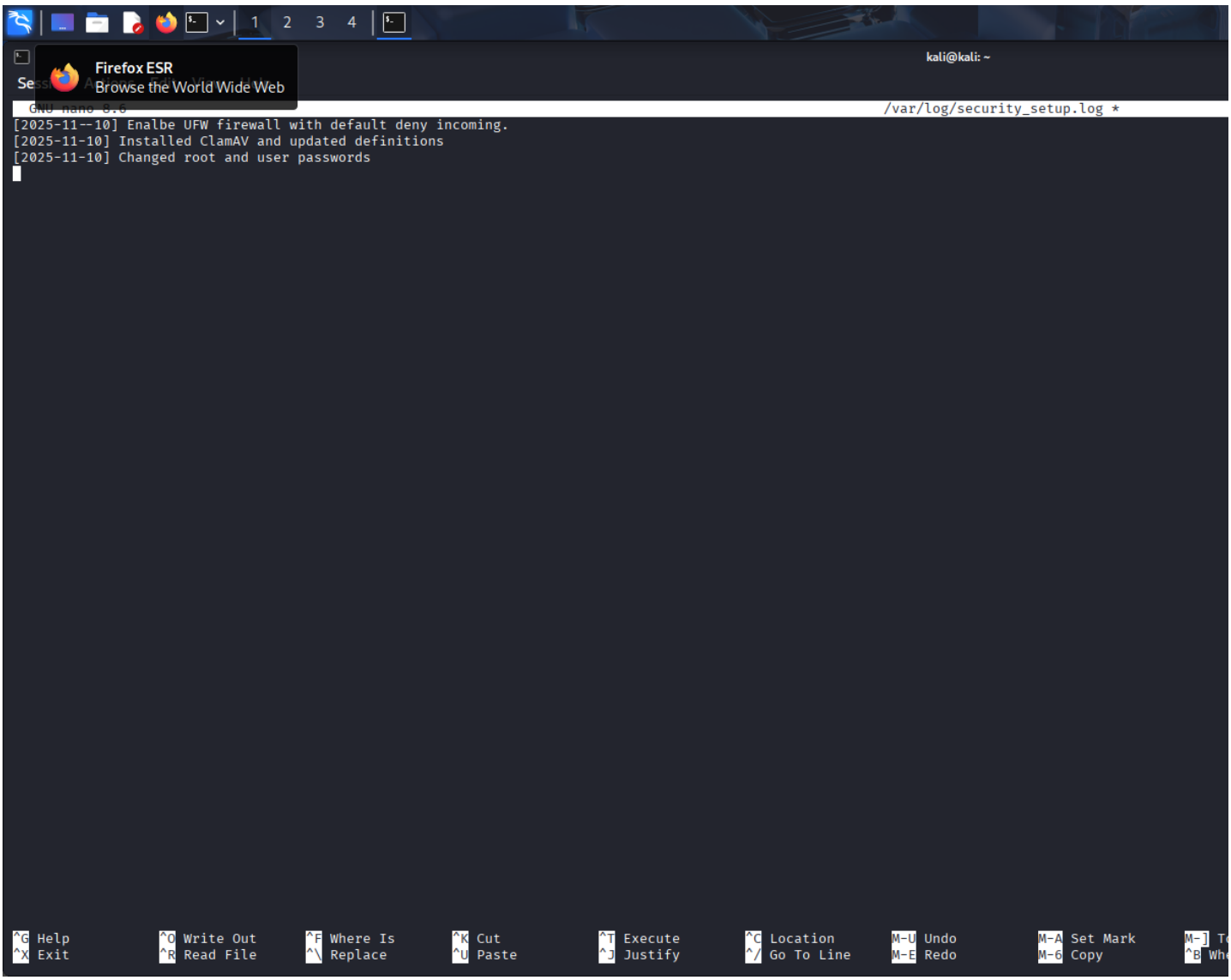


Document Security Settings

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.



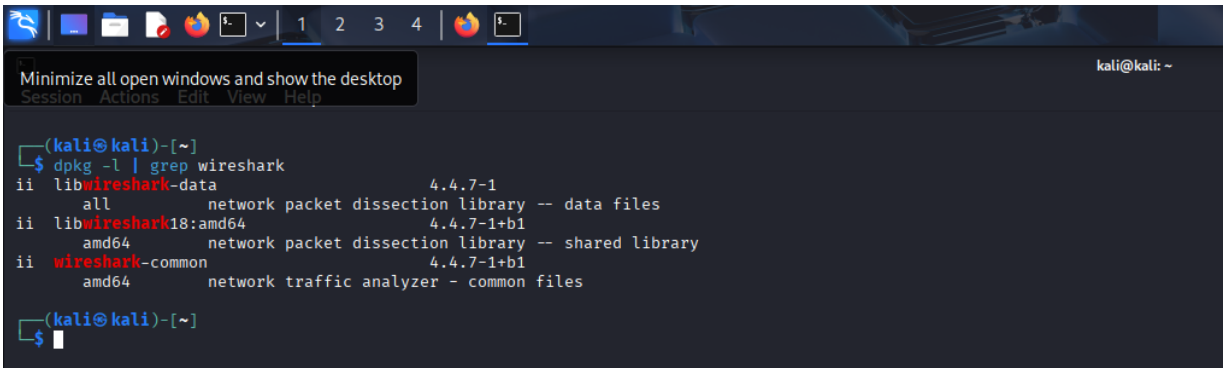
This command opened the GNU nano text editor on the command line which allowed us to log system changes.



Troubleshooting & Documentation

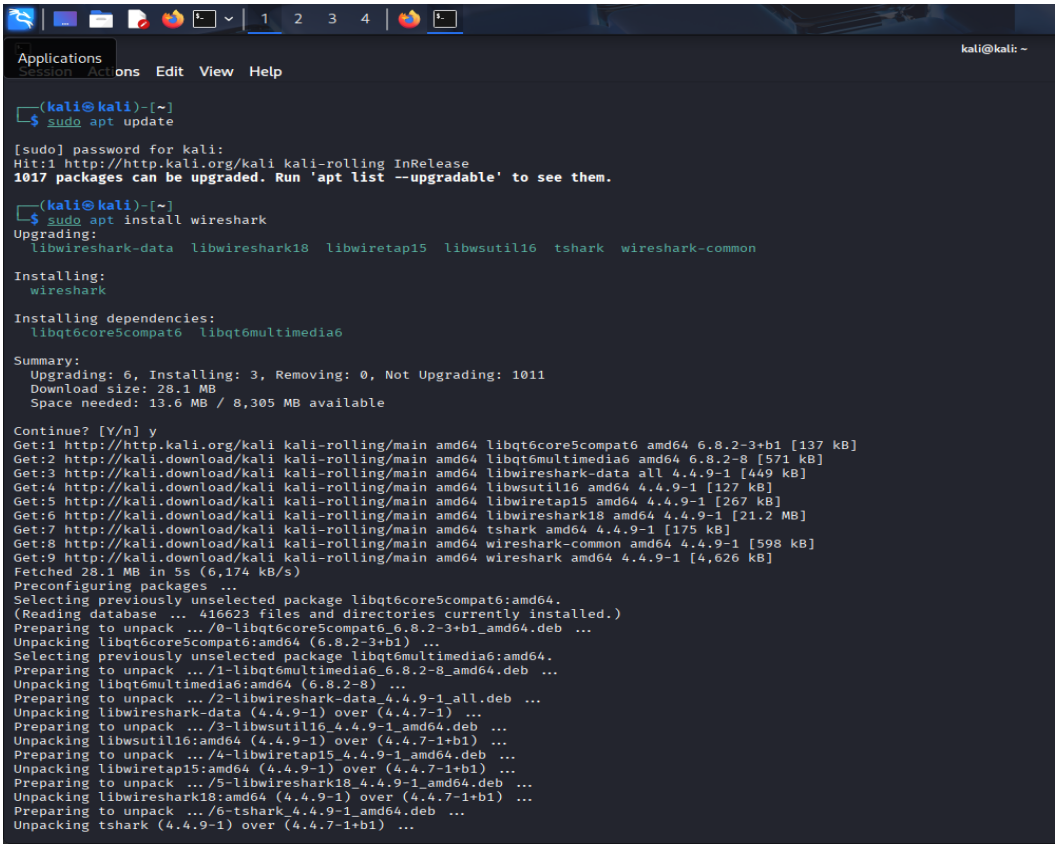
Problem 1 Wireshark won’t launch

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.

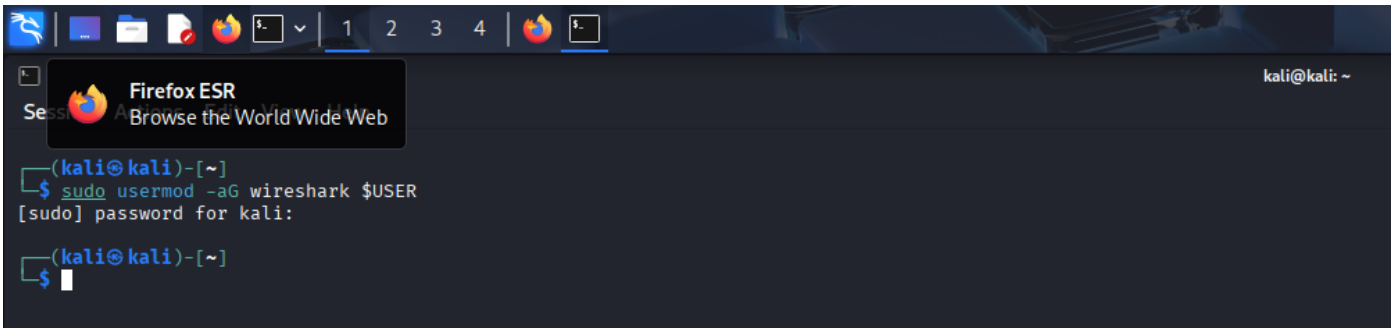


Installing Wireshark

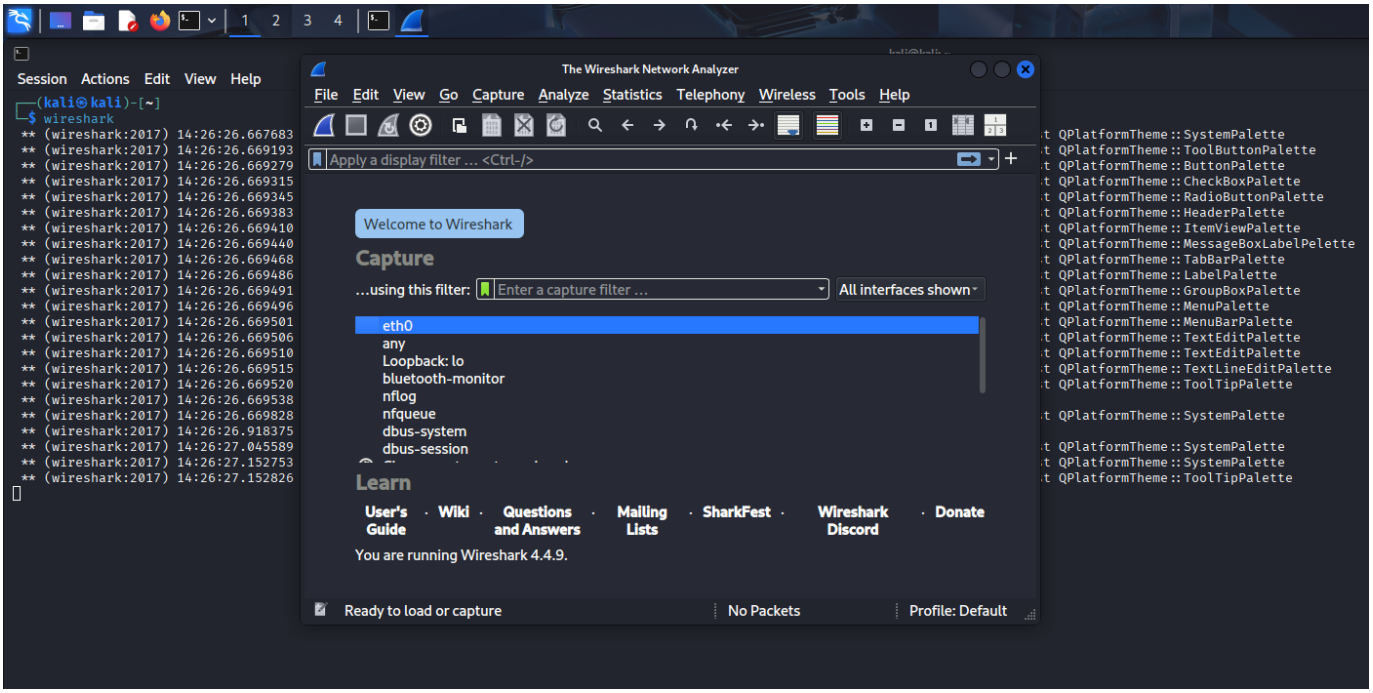
We then updated the packages using the sudo apt update command and then went on to install wire shark using the sudo apt install I wireshark command.



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.



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

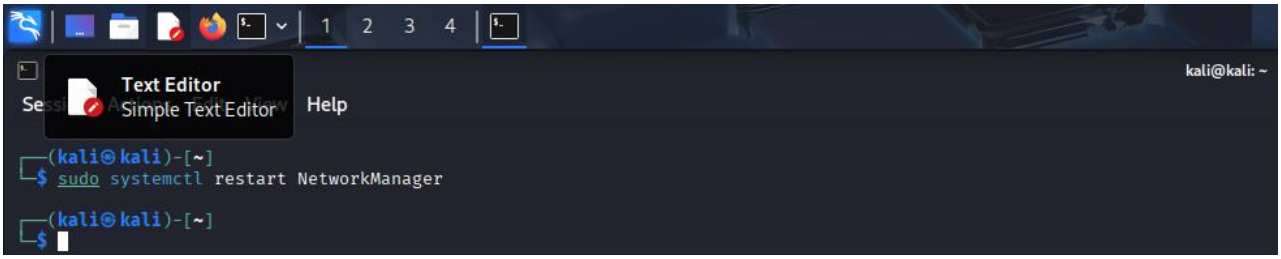


wireshark GUI launched without issue.

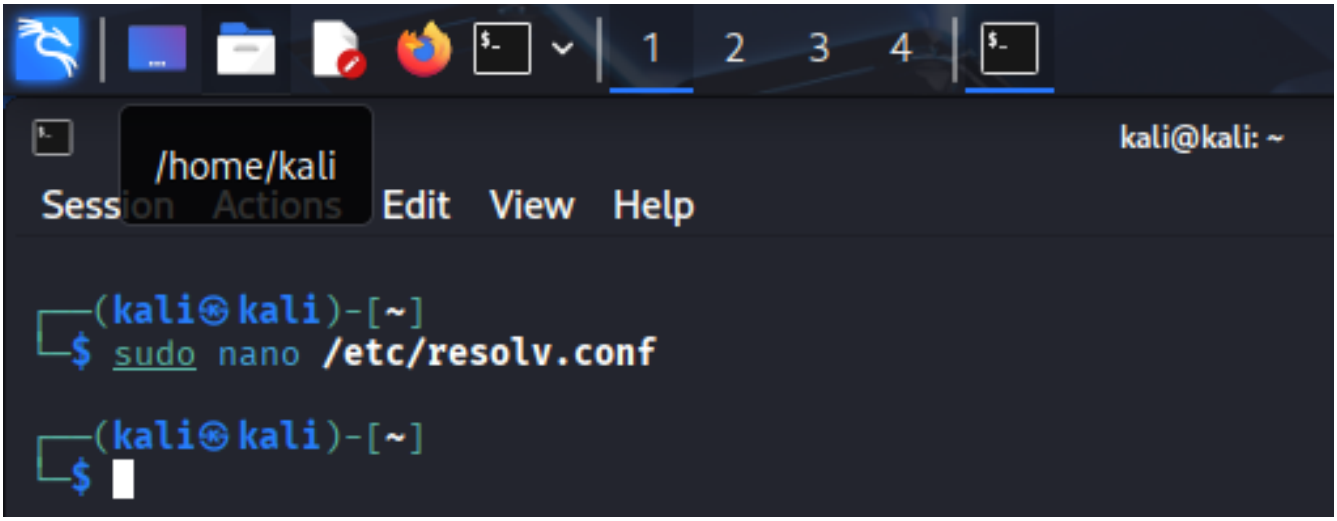
problem 2 slow internet access

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.

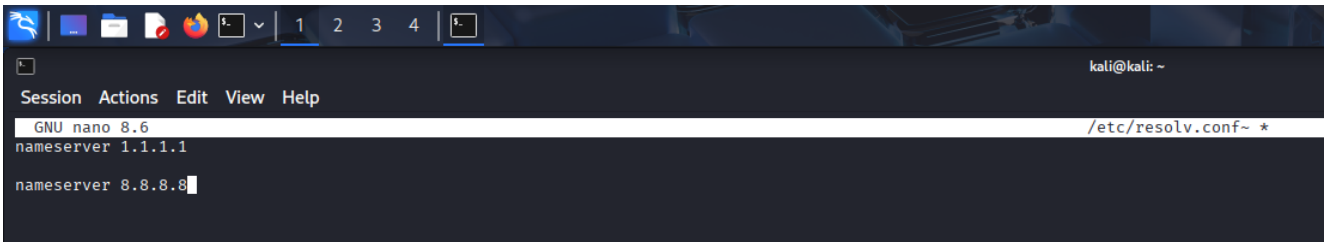
We restarted the network adapter using the **sudo systemctl restart NetworkManager**



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



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



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.


ali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
SPEEDTEST
English
Apps
Learn
Data
SHARE
Result ID 18458162871
RESULTS
SETTINGS
DOWNLOAD Mbps
42.83
UPLOAD Mbps
61.48
Ping ms
7
63
20
GO
Connections
Multi
FirstNet UrbanXConnect
Johannesburg
Change Server
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