SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

YEAR 2, SEMESTER 1

INDIVIDUAL ASSIGNMENT

IE2012 – SYSTEMS AND NETWORK PROGRAMMING

IT23440418

D.O.J. Silva

TERMS OF REFERENCE

This report is presented to meet the criteria of the Sri Lanka Institute of Information Technology's IE2012 – Systems and Network Programming.

Table of Content

1. Basics of Linux Environment

1.1 Installing Kali Linux

Kali Linux can be installed as normal operating system or a virtual operating system, which can be run virtually on windows or any other operating system. To install it virtually the following prerequisites should be met:

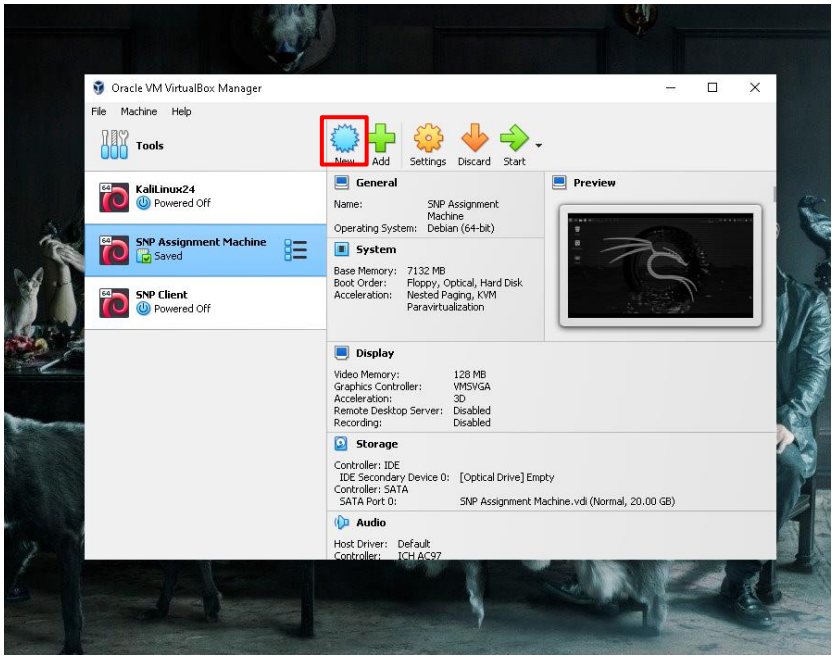
- A computer having at least 8GB of RAM, and a 4-6 Core CPU.
- A software to run virtual operating systems.
- Kali Linux operating system.

Oracle virtual box is a good choice for virtual operating system, because its free and open source, while offering cross-compatibility and many other features. Oracle virtual box can be downloaded from the official website which will be linked here.

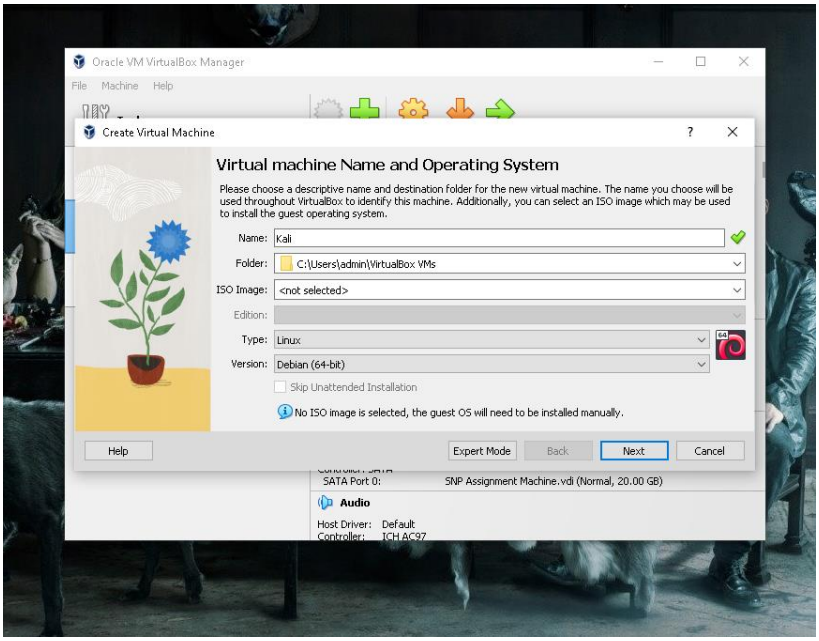
Kali Linux OS can be downloaded either as an ISO file for custom installation, or as a pre-installed version to be run on virtual box software. The difference lies between the installation, while the ISO file offers the user to customize the installation, the pre-installed version is ready to use. Both can be downloaded from official website, which will be linked here.

After the above prerequisites are met, Kali Linux can be installed with the following steps.

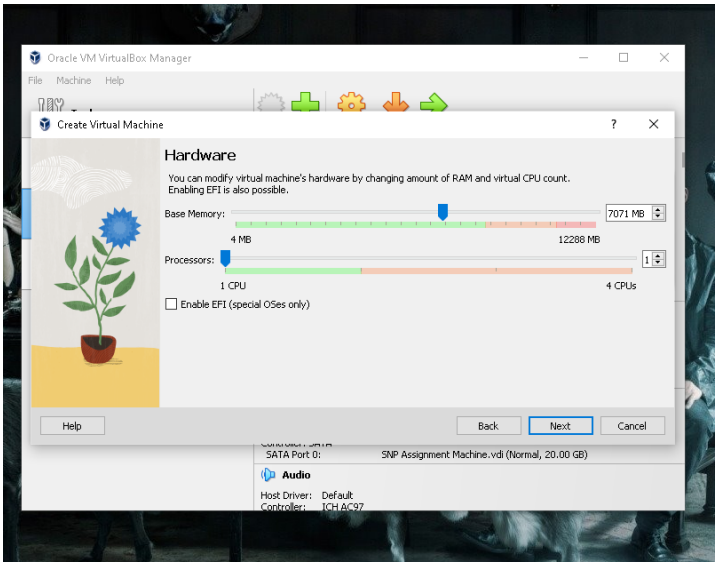
1. Open Oracle virtual box.
2. Select New option to create a new virtual machine on VirtualBox.



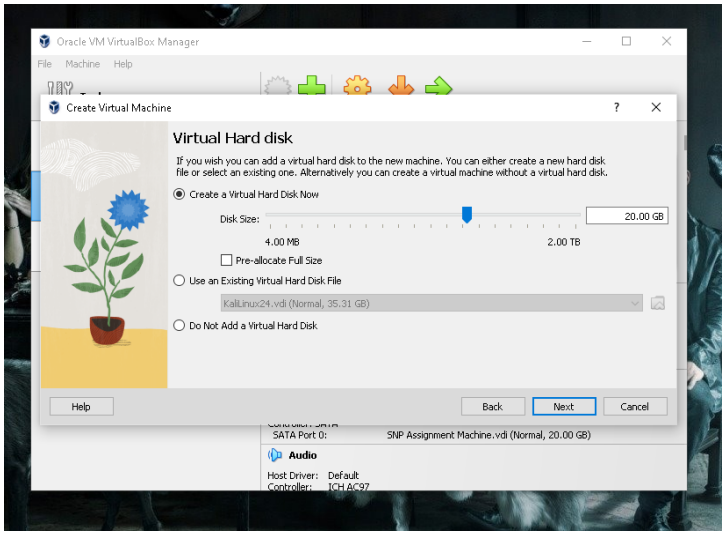
3. Name the machine and select location, and select type as Linux and Version debain 64 (as I use Kali Linux distribution). Then click next.



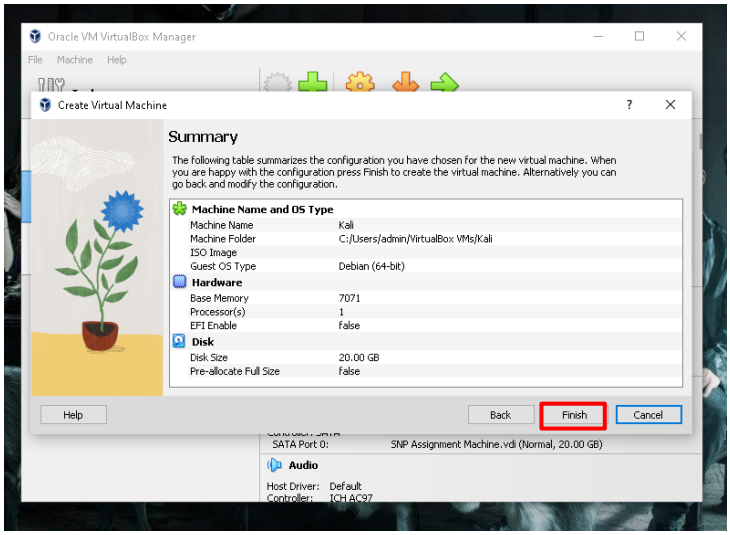
4. Allocate memory and CPU cores. It is good to select a number within green color range.



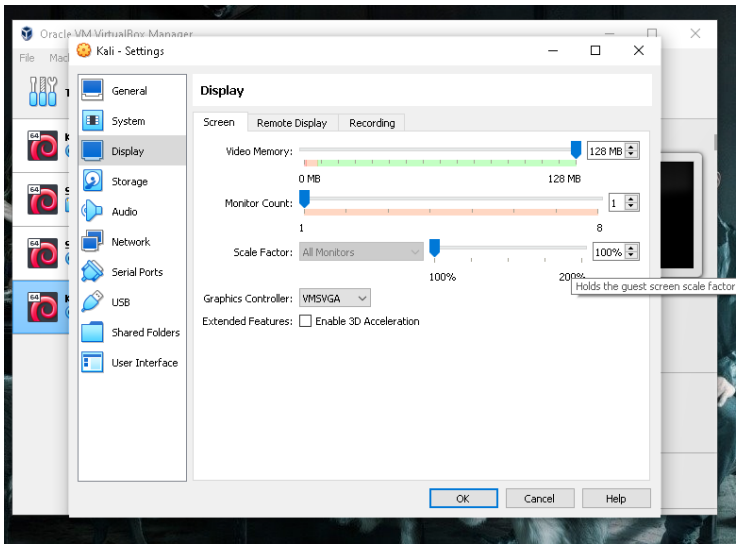
5. Then allocated disk size to virtual machine. It is good to allocate about 20GB.



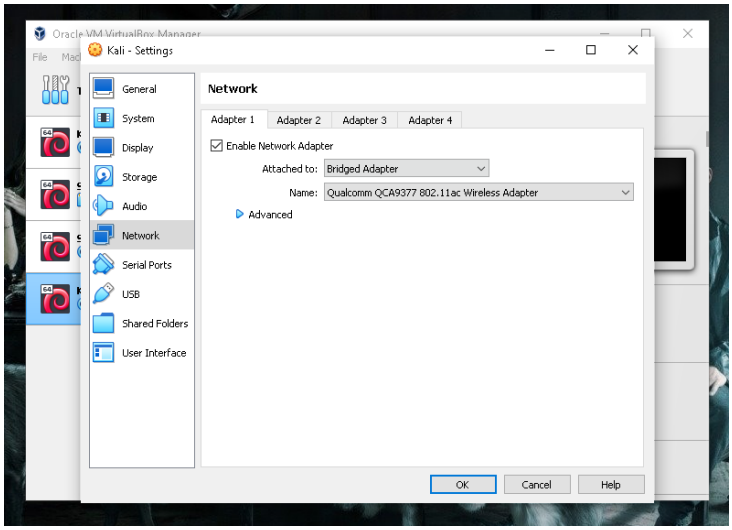
6. Then click finish.



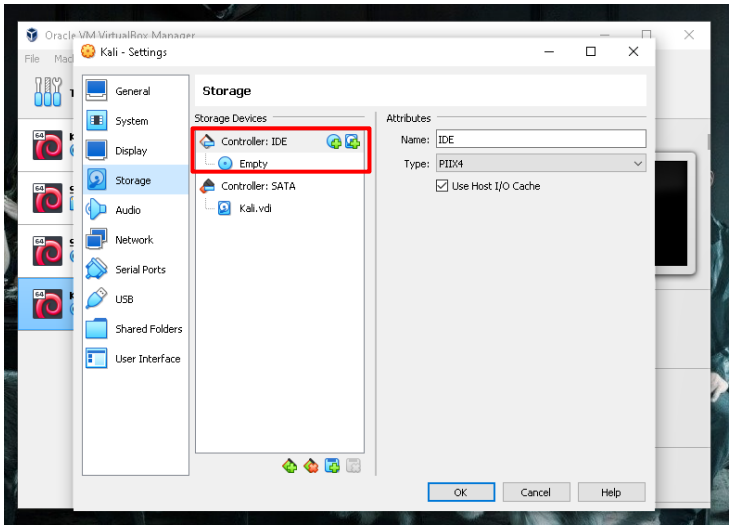
7. Then click settings and go to display option and allocate video memory.

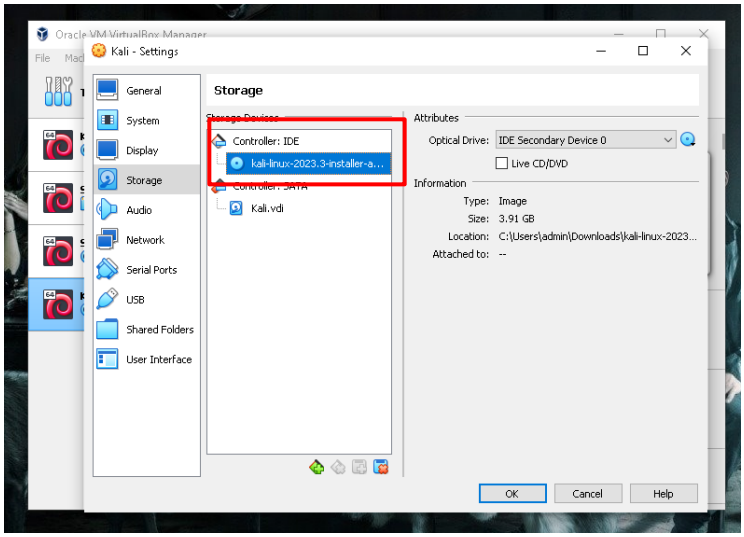


8. Configure the network settings.



9. Mount an OS installation ISO file.





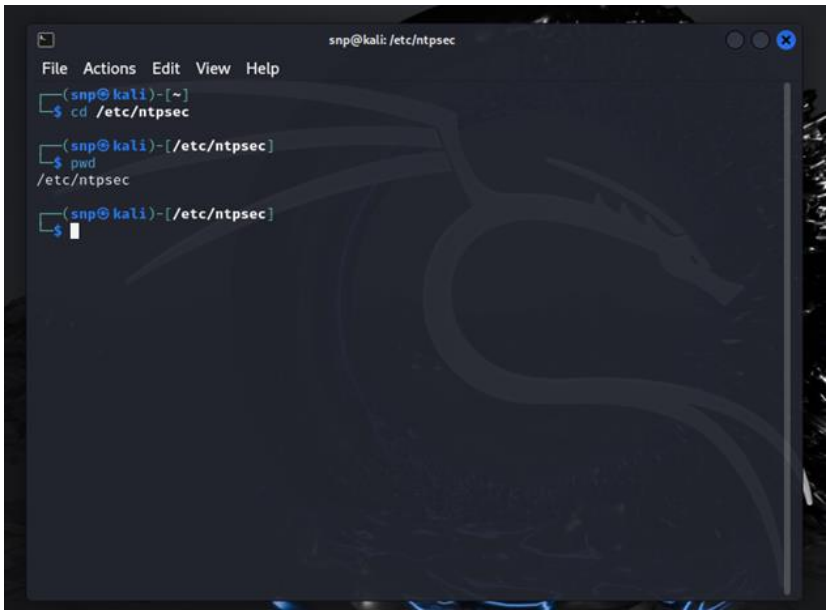
10. Start the VM and follow the OS installation steps.
11. Log in with the username and password given in OS installation steps.

1.2 Command Line Introduction

Navigation Commands

cd - The cd command in Linux is used to change the current directory in the terminal. It allows you to navigate between directories in the file system.

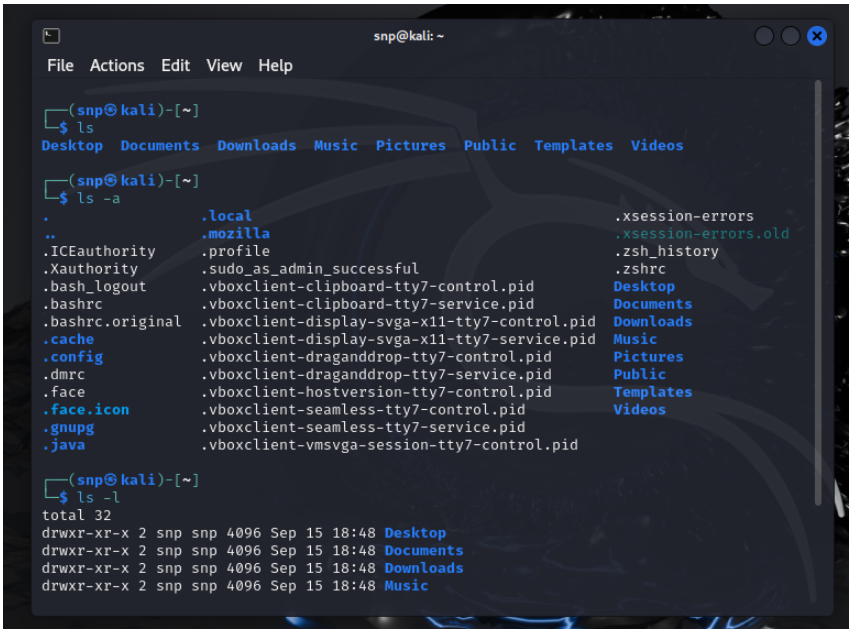
pwd - The pwd command stands for Print Working Directory. It displays the full path of the current directory you are in.



ls - used to list files and directories in the current directory or a specified directory.

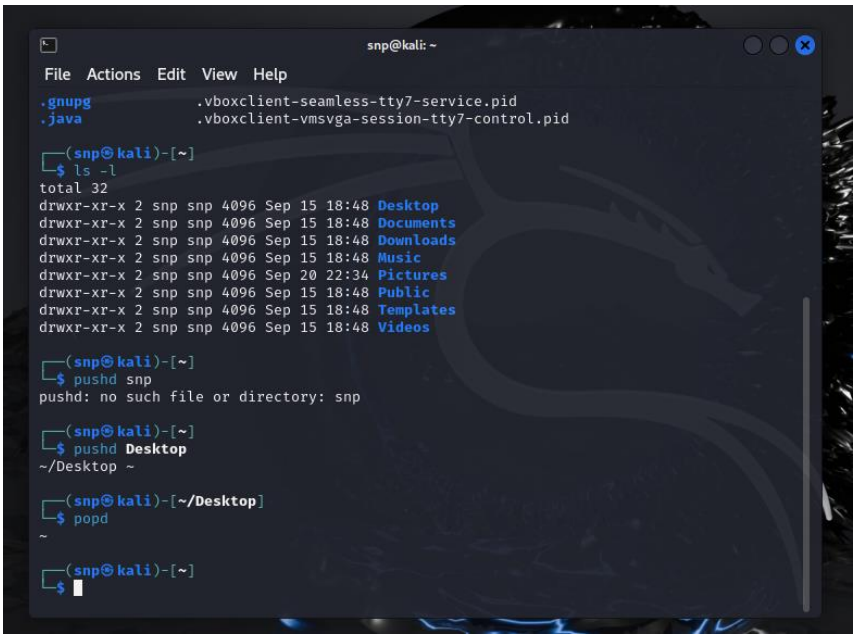
ls -a - lists all files and directories, including hidden ones, in the current directory. Hidden files and directories in Linux are those that start with a dot (.) .

ls -l - lists files and directories in a detailed format, showing additional information such as permissions, number of links, owner, group, file size, and last modification date.



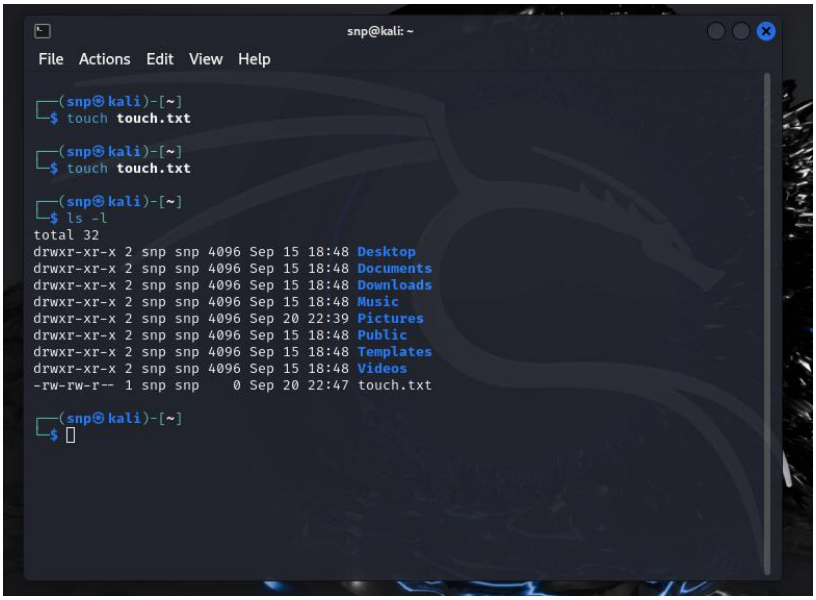
pushd - used to change the current directory and save the previous directory on a stack.

popd - used to change the current directory back to the directory that was most recently saved on the directory stack by the pushd command.

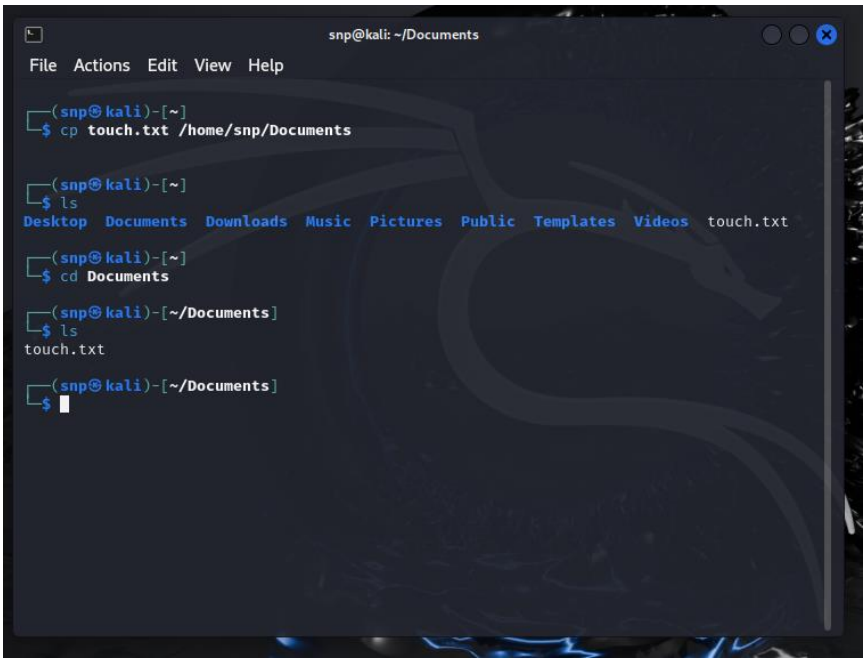


File Manipulation Commands

touch - used in Linux to create empty files or to update the timestamps of existing files. It's a simple way to create a new file without needing to open a text editor.

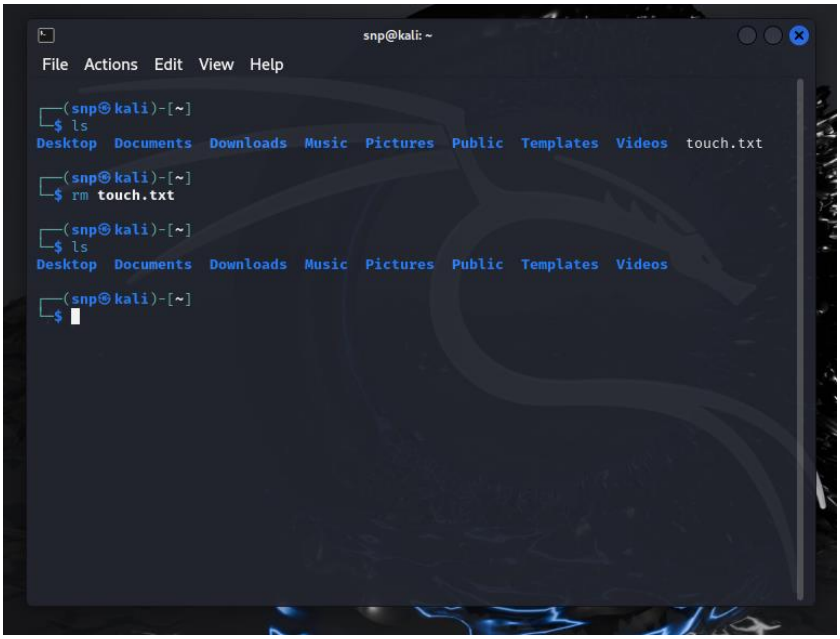


cp - used to copy files and directories in Linux. It allows you to create a duplicate of a file or directory at a specified location.

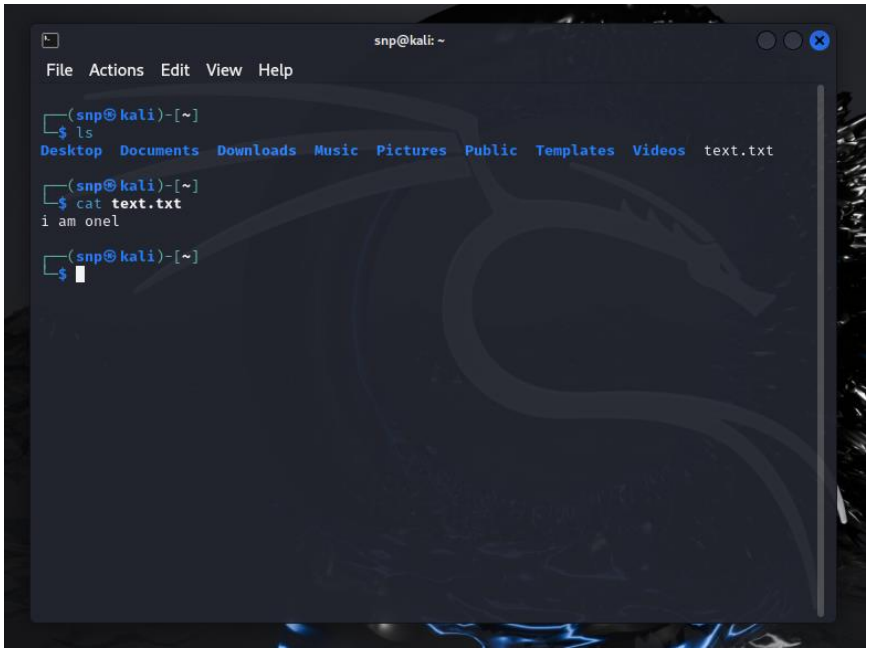


mv - used to move or rename files and directories in Linux. It allows you to change the location of a file or directory or give it a new name.

rm - used to remove files and directories in Linux. It permanently deletes the specified files or directories.

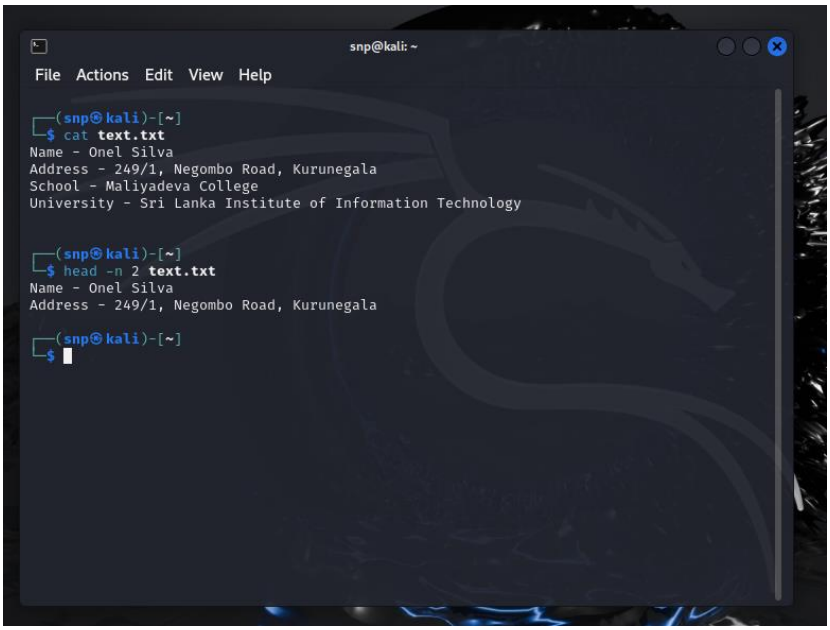


cat - used to display the contents of files, combine multiple files, and create new files in Linux.



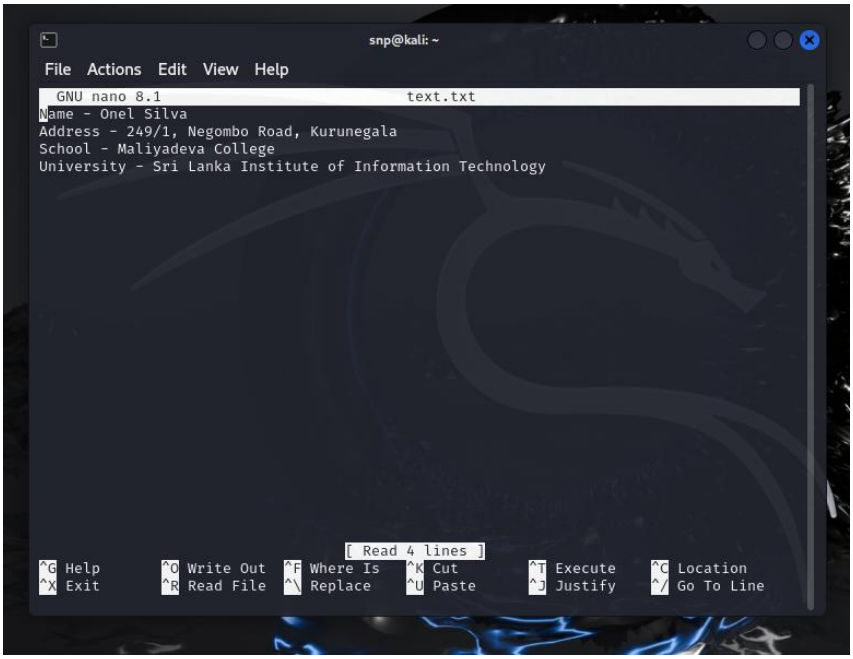
basic Linux commands

head - used to display the first part of files. It shows the first 10 lines of a file by default, but you can specify how many lines or bytes to display.

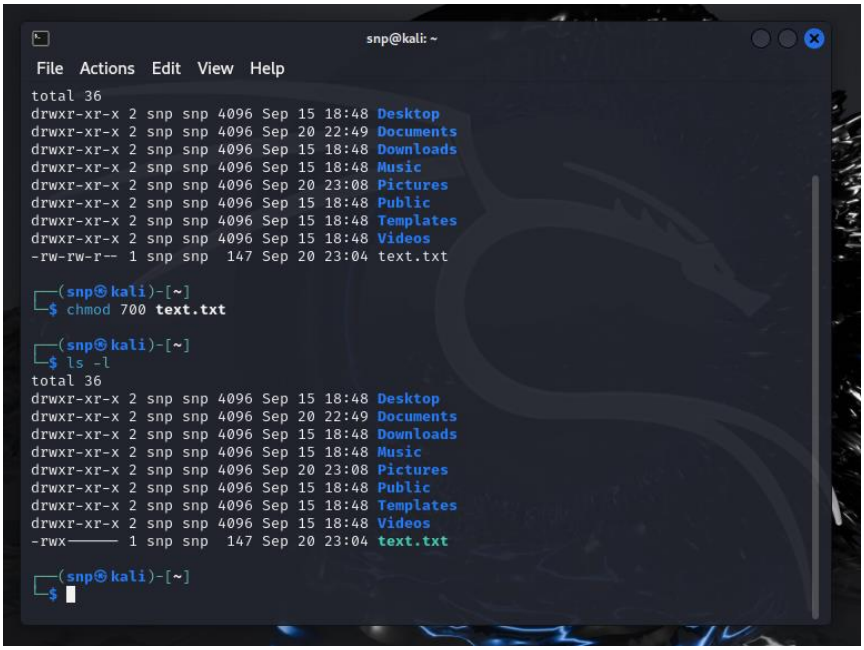


tail - used to display the last part of files. It shows the last 10 lines of a file by default, but you can specify how many lines or bytes to display.

nano - command-line text editor and opens the editor.



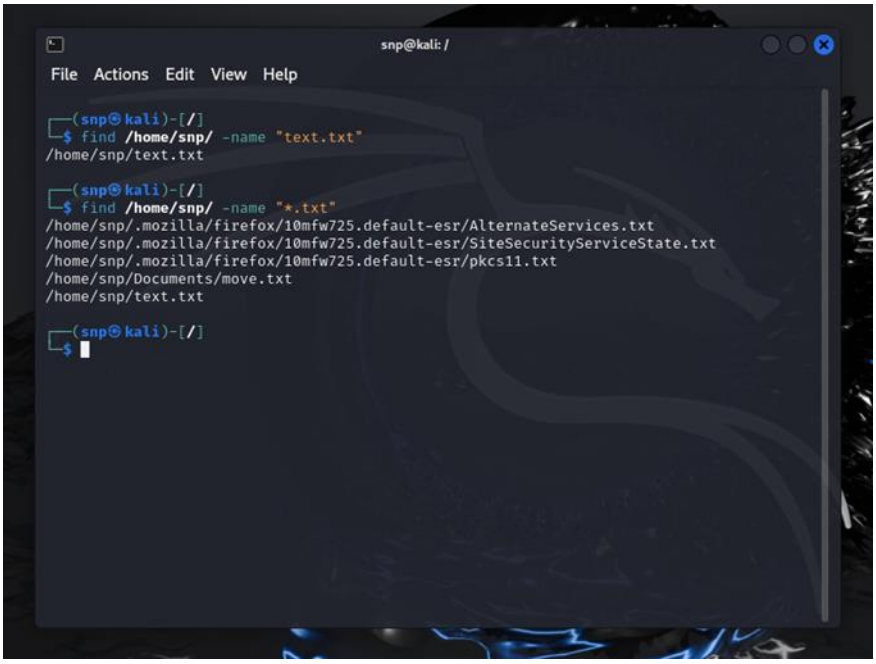
chmod - Change the permission of a file or directory. It controls who can read, write, or execute a file by setting or modifying permissions for the file owner, group, and others.



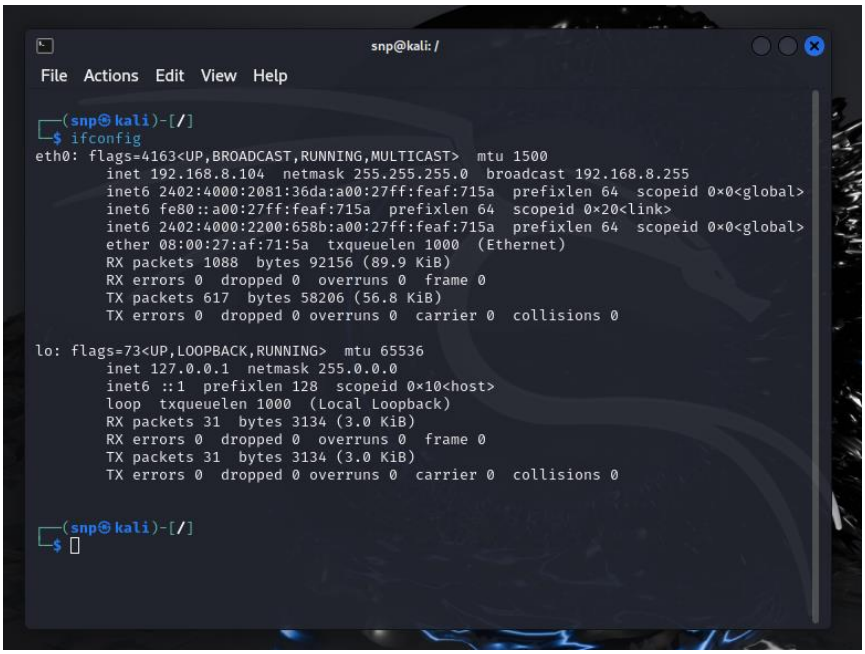
grep - used to search for specific patterns or strings within files. It scans the input and outputs lines that match the pattern you specify.

find - used to search for files and directories in a directory hierarchy.

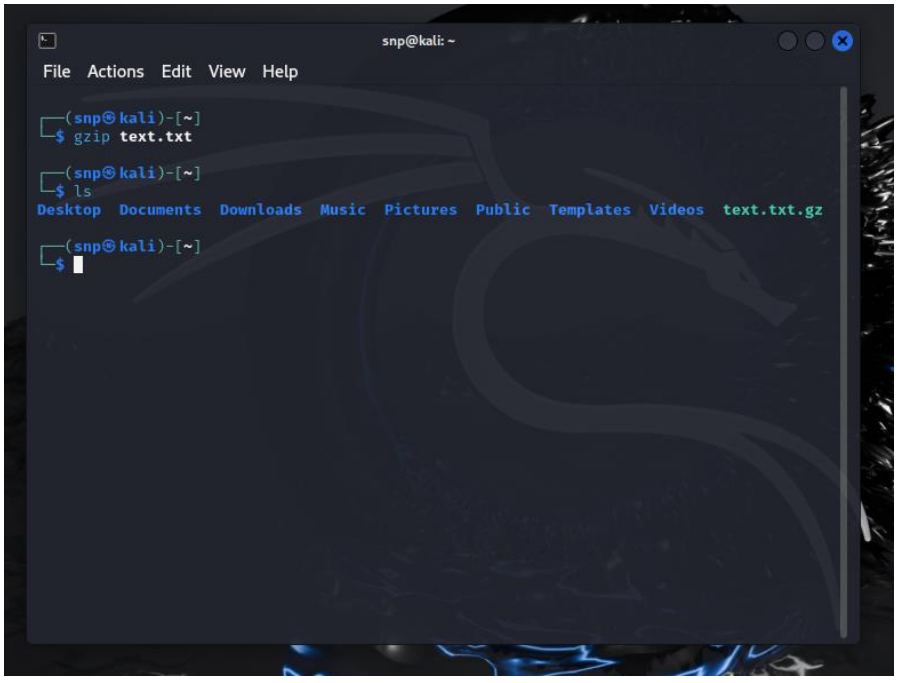
find /path/to/search -name "*. extension" – used to search for files and directories using extensions.



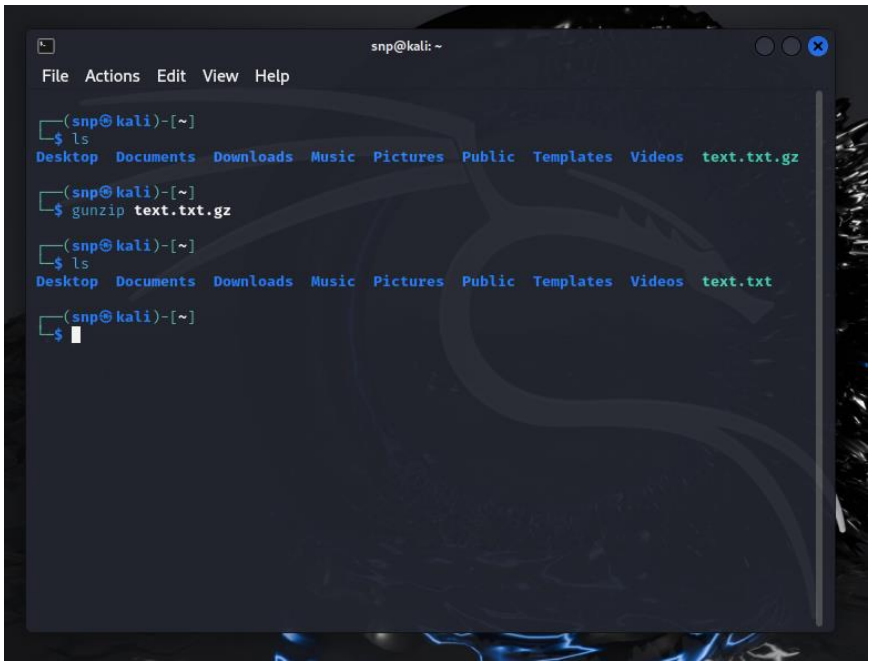
ifconfig - used to display network interface parameters. It provides information about a system's network interfaces, such as their IP addresses, MAC addresses, subnet masks, and more.



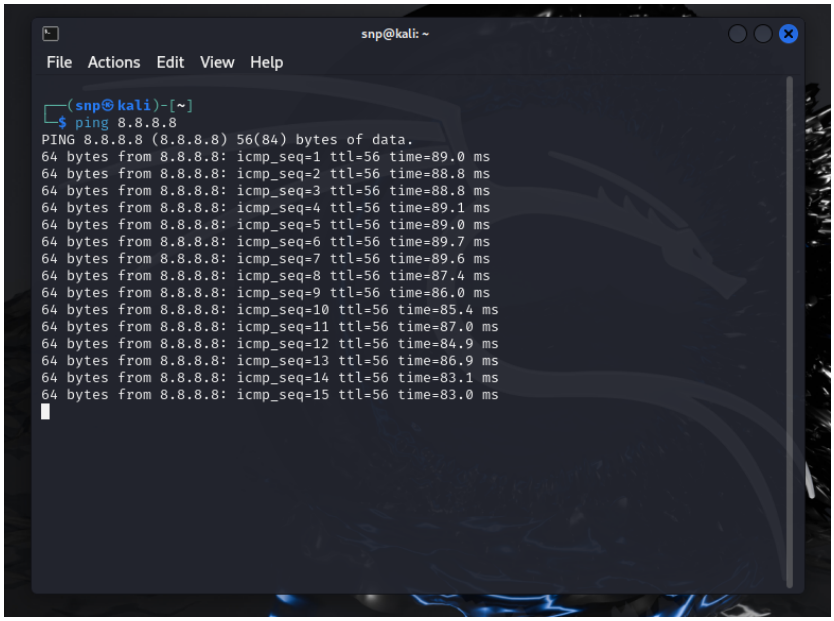
gzip - used to compress files, reducing their size using the GNU zip algorithm. One of the most common tools for file compression in Linux.



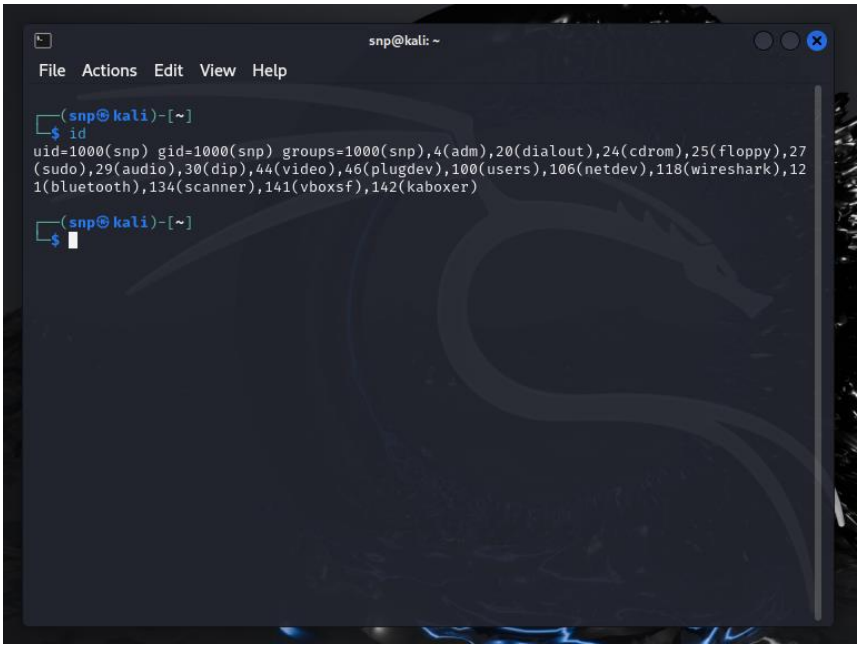
gunzip - used to decompress files that were compressed using the gzip. It restores the original file from its compressed gzip version.



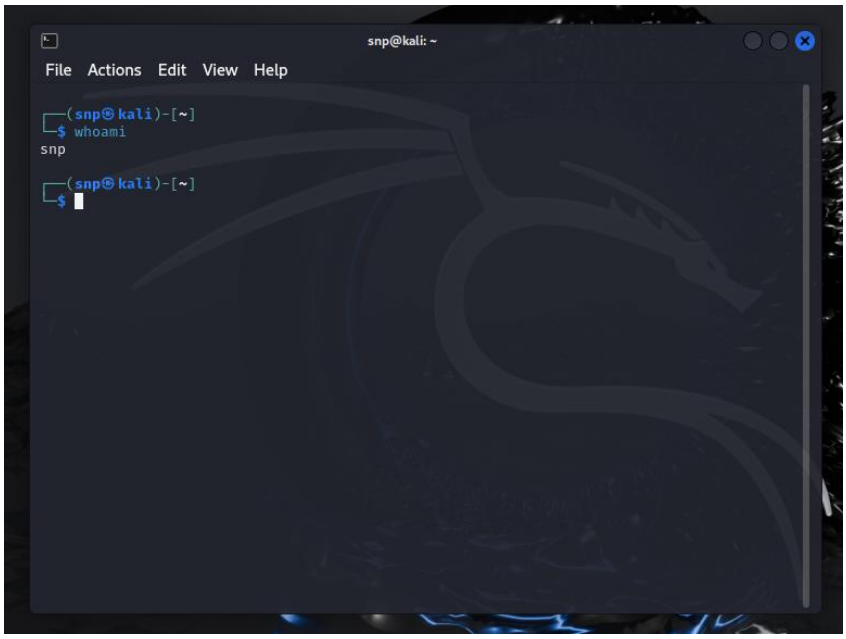
ping - Network diagnostic tool that tests the reachability of a host on a network. It works by sending ICMP Echo Request packets to the specified host and waits for an Echo Reply.



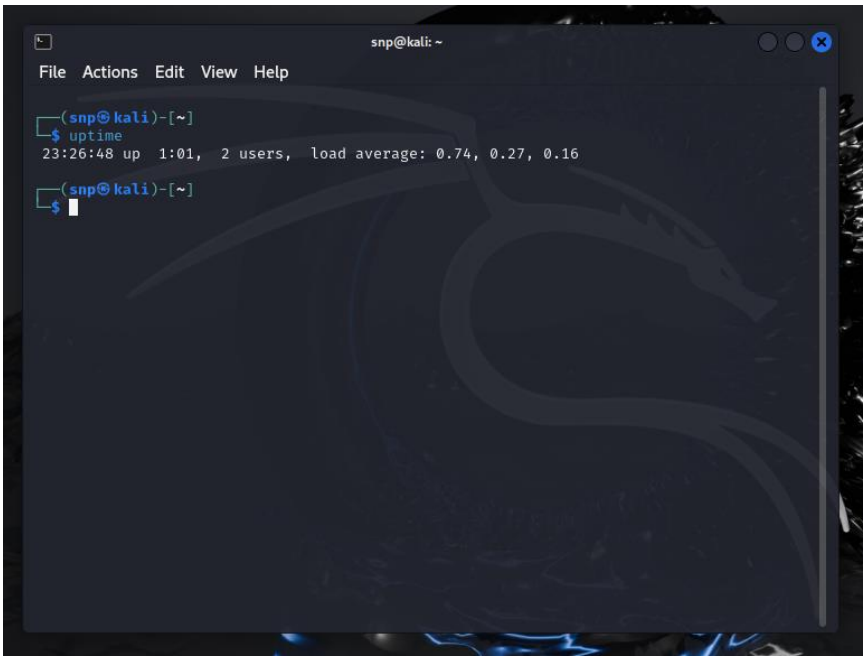
id - used to display user and group information for the current user or a specified user.



whoami - used to display the username of the currently logged in user.

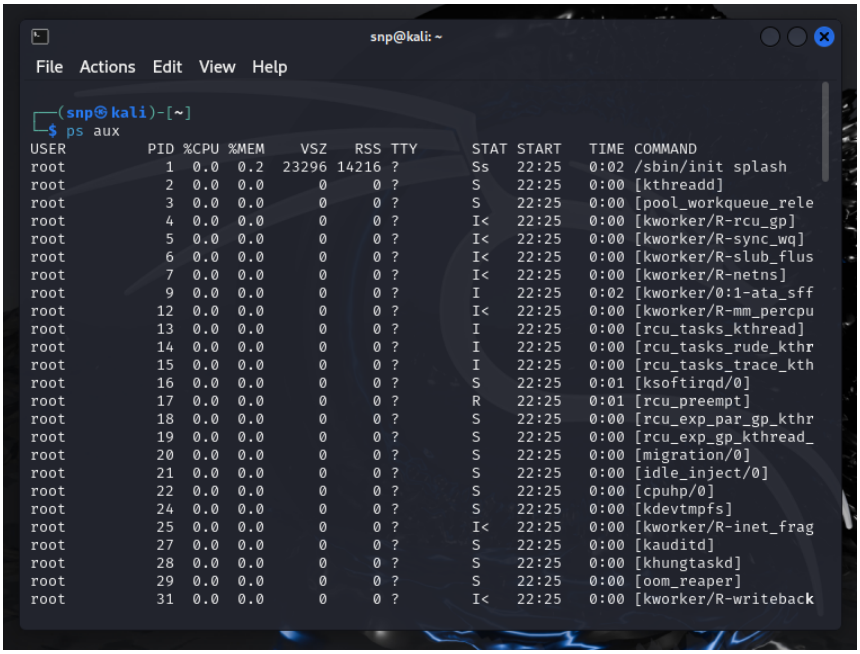


uptime - used to display how long the system has been running since its last reboot, along with information about system load averages.



free -h - used to display information about the system's memory usage, amount of free and used memory. "-h" option stands for "human-readable" and formats the output in a more human readable way using units like MB or GB.

ps aux - used to display detailed information about currently running processes.



2. DHCP, DNS, NTP Services

2.1 Understanding and configuring DHCP

Role of DHCP in Network Configuration

DHCP is a network protocol that is used for the distribution of configuration parameters-such as IP addresses, subnet masks, default date ways, and so on-in an automated manner to clients on the network. The main purpose for which DHCP was designed is to connect devices to the network through automatic means, rather than having to enter the settings of the network manually[2]. Every computer on a network has an IP address that is unique to the computer.

There are 2 ways to assign an IP address to a computer connected to a network.

- Static IP
- Dynamic IP

Static IP

This mode of allocation requires user to manually configure the network settings for the devices.

This allocation mode consumes more time and is prone to errors since the configuration is done by the user.

Dynamic IP

This mode involves the client sending a request message to the server. After, the server

shares the require network configuration and resources to the client and configure the device automatically. Then the client can access the internet and network services.

DHCP Dynamic Allocation Process

DHCP Discover

When the client, or device, connects onto the network, it does not yet have an IP address. So the client broadcasts a packet, which is called the DHCP Discover message, to discover which DHCP servers are available on that network. Within that message, it includes its MAC address and a request for an IP address [2].

DHCP Offer

Each DHCP server that receives the DHCP Discover message from the client replied with a DHCP Offer message. And this message includes an available IP address, subnet mask, lease time and other configurations. The offer message is sent to client's MAC address as a broadcast packet [2].

DHCP Request

In return, the client gets one or more DHCP offers from the servers. The client then selects one of the offers and responds with a DHCP Request, informing the server of which offer it has accepted. A DHCP request message contains the IP address that the client device is requesting [2].

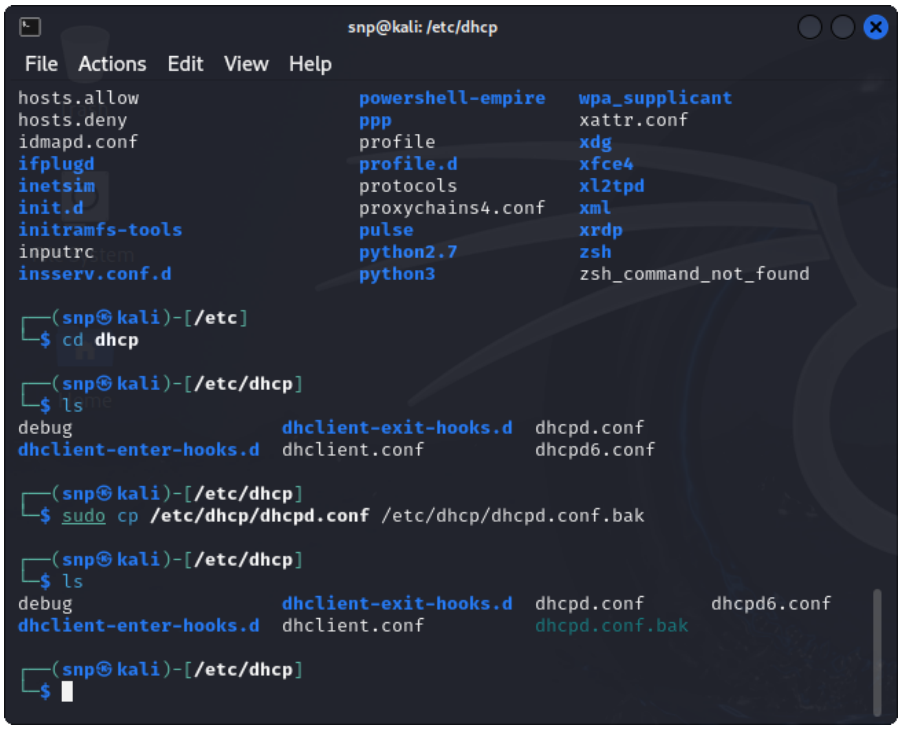
DHCP Acknowledgment

The DHCP request is received by the DHCP server, which then sends a DHCP Acknowledgment message in return for acknowledging that the IP address has been assigned to the client. On reception of the DHCP Acknowledgment message, the client configures its network interface with the assigned IP address and other settings [2].

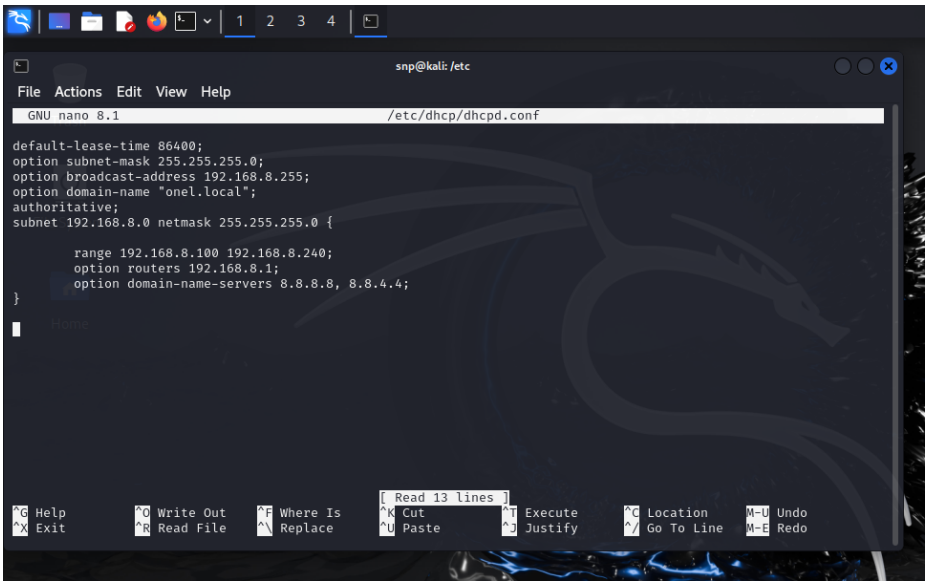
Installing and Configuring a DHCP Server

1. Install the DHCP server package.

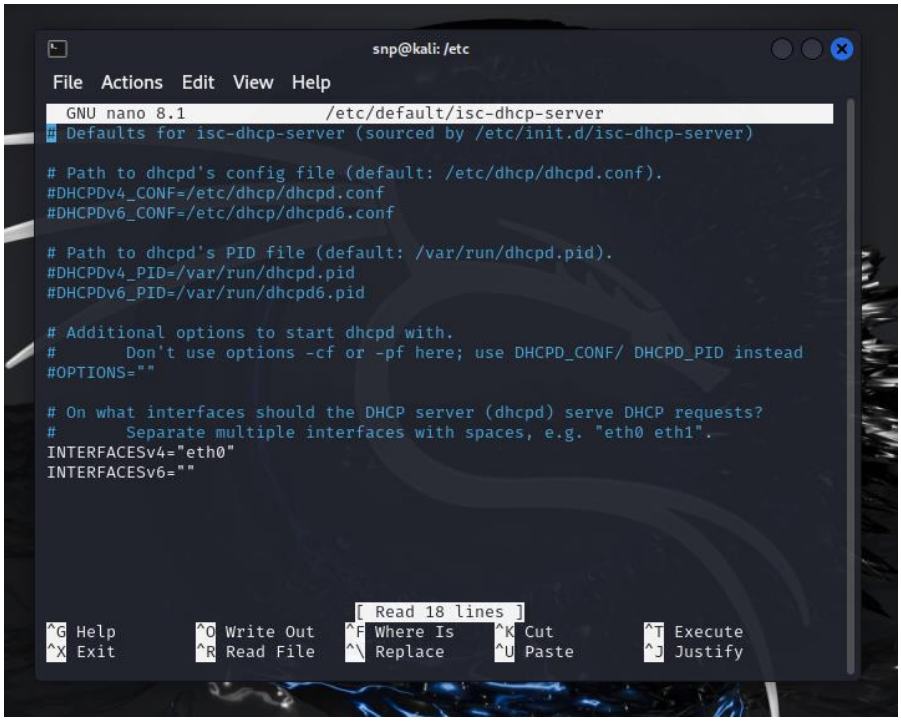
2. Create a backup of DHCP configuration file



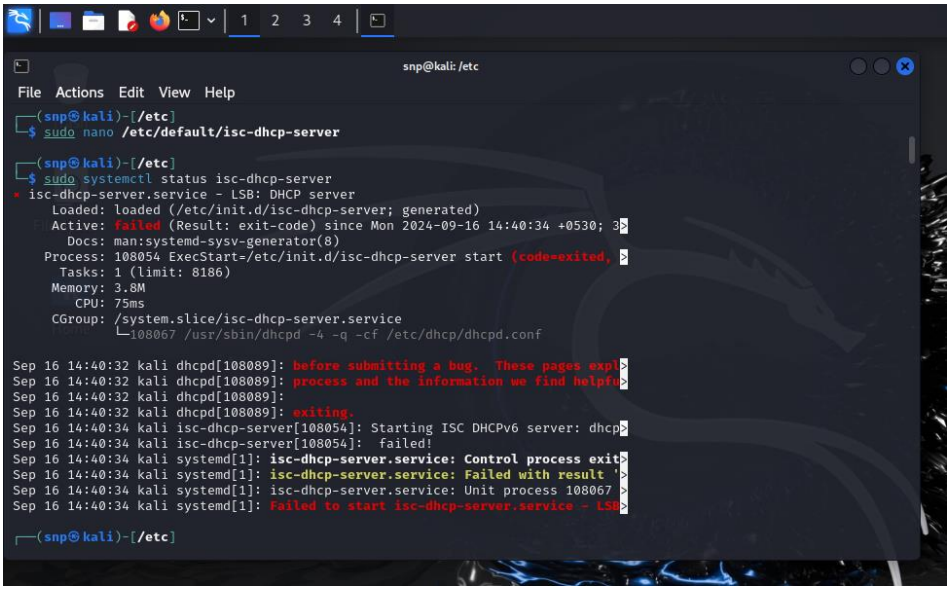
3. Open the DHCP configuration file and modify the configuration according to your requirement.



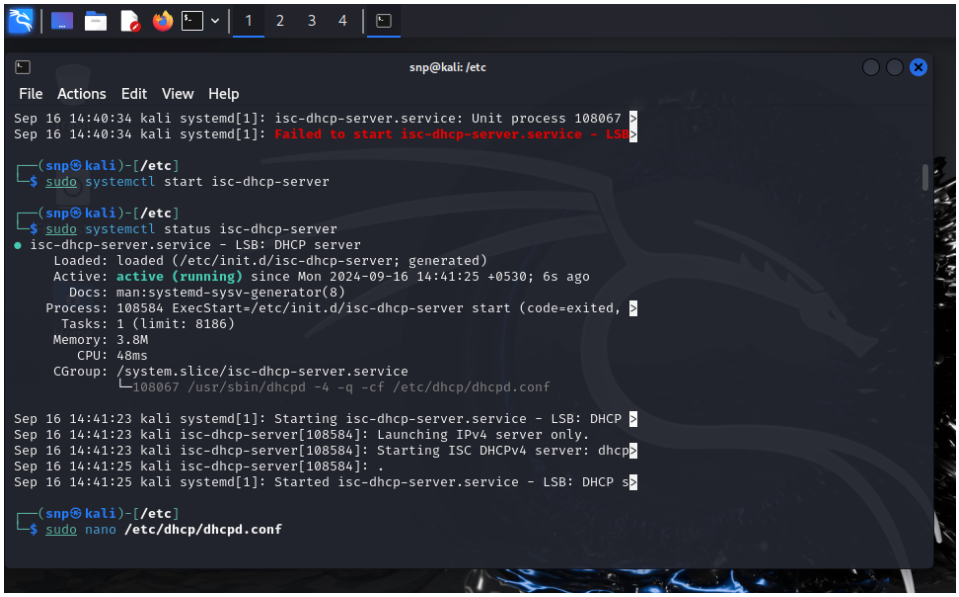
4. Assign network Interfaces for the DHCP server.



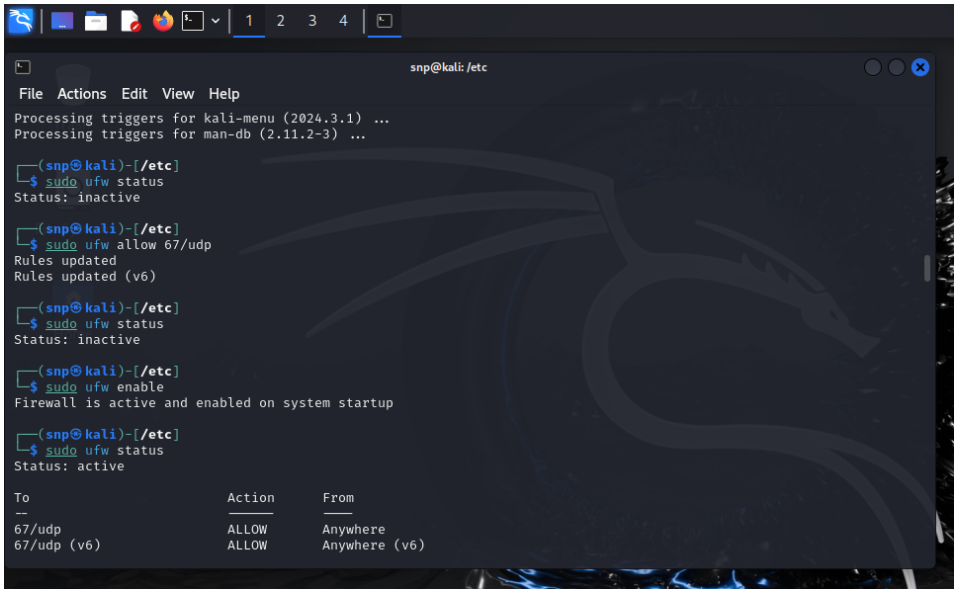
5. Check status of DHCP server.

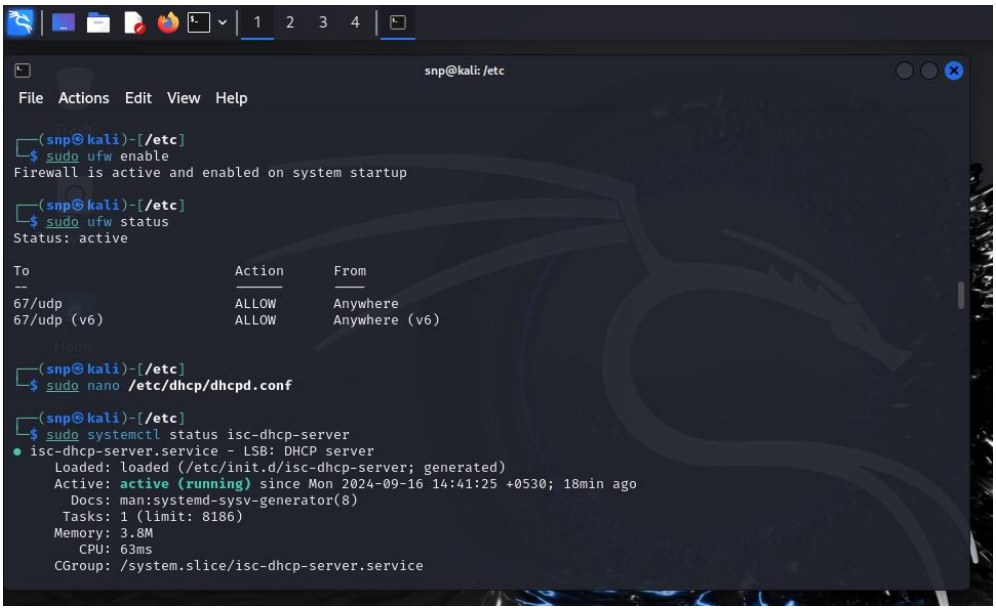


6. Restart and Enable the DHCP Server.



7. Configure Firewall

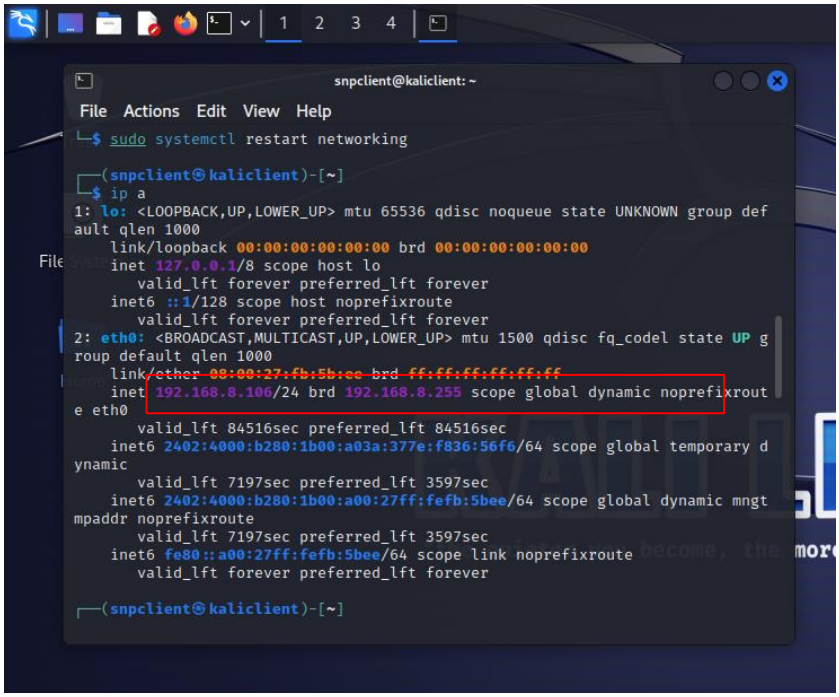




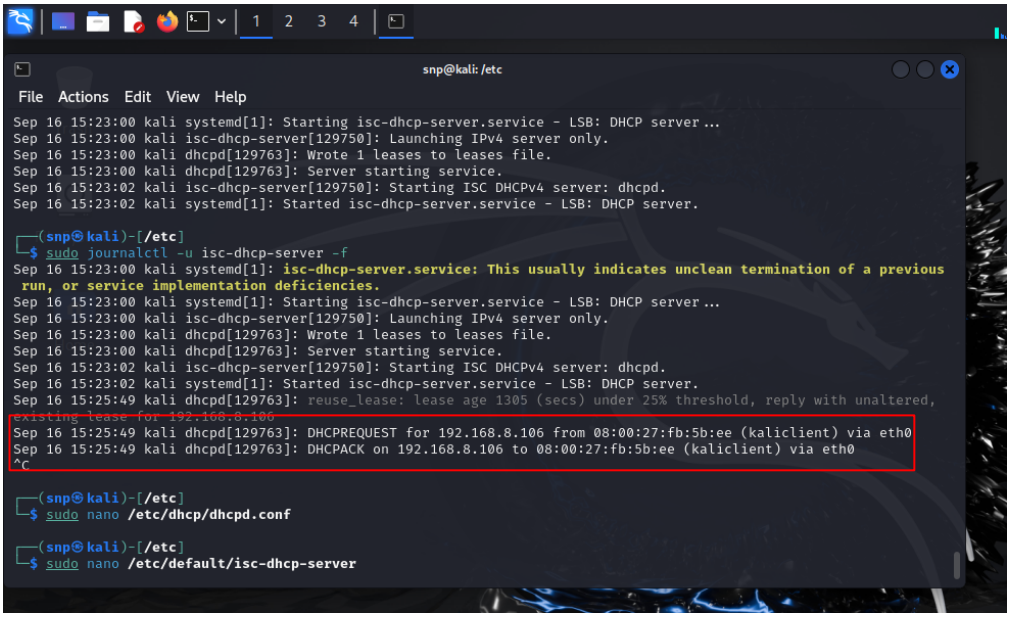
Testing DHCP in virtual environment

A separate virtual machine is created in Oracle VirtualBox specifically for the client system. This VM was configured to use DHCP for network settings, ensuring it could request an IP address from the DHCP server running on another VM. Both the client and server VMs were connected to the same virtual network within VirtualBox to allow communication between them.

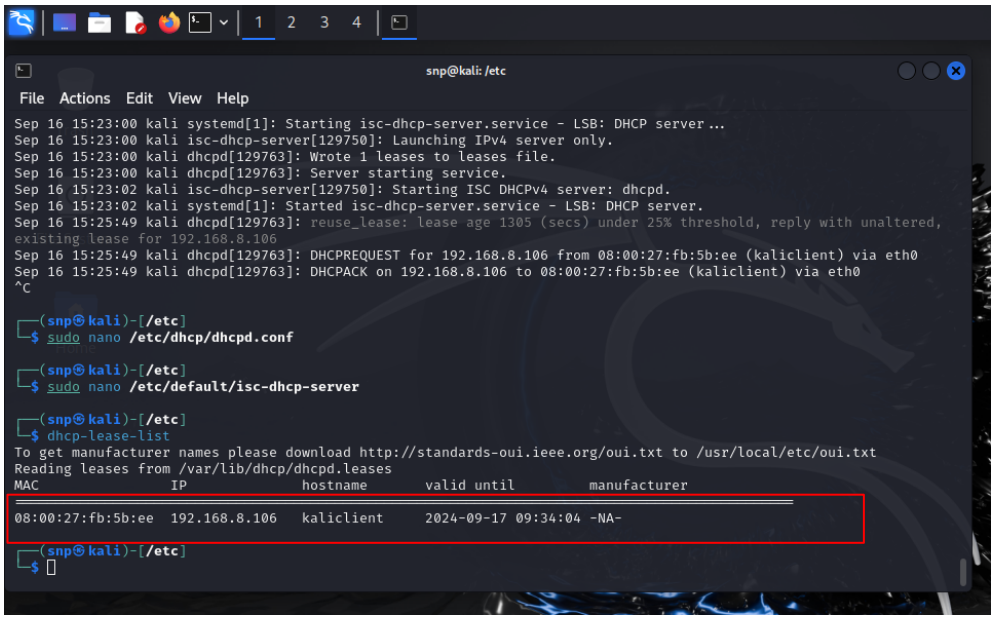
1. Log to the client machine and check network configuration.



2. Confirm DHCP Server logs



3. Confirm DHCP Lease



2.2 Understanding and configuring DNS

Role of DNS

DNS translates human-readable, readable domain names into their respective IP addresses, which computers use to find each other on any network. At its very core, DNS means a way for ease of access by humans trying to access websites and services on the internet. Instead of having to remember long strings of numbers for each and every website they want to go to, users can simply type in a domain name into their web browsers and DNS will find the proper IP address that is associated with that domain name [1].

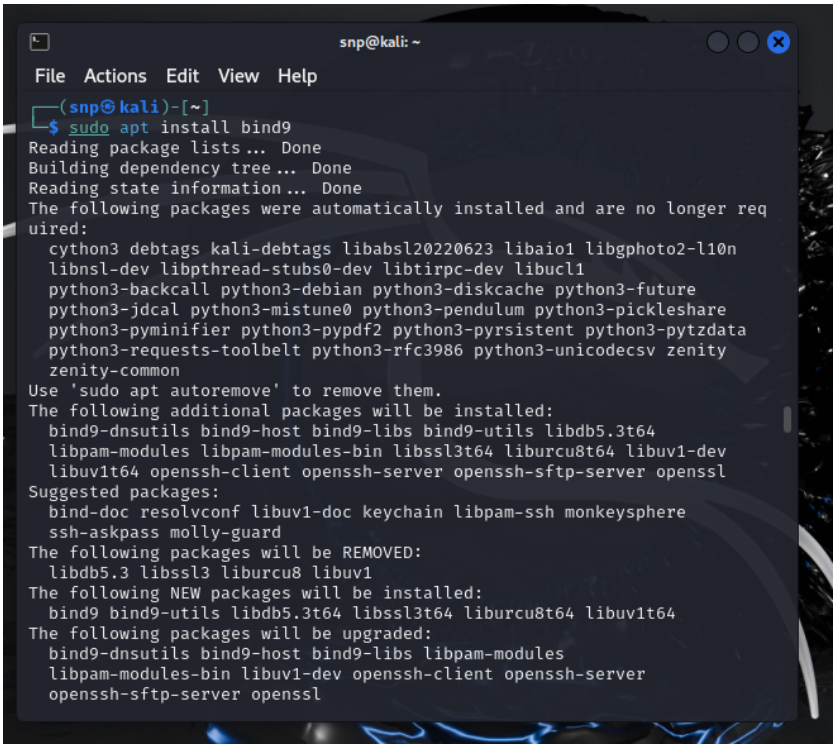
DNS Process

1. A user types a URL in their browser's address bar.
2. The browser checks its cache for IP address relevant to the domain. If not found, the request is sent to the DNS resolver.
3. The DNS resolver checks if it has the IP address cached. If IP address is cached, it is returned to the user. If not, the resolver forwards query to a Root DNS server.
4. The root dns server direct the resolver to the appropriate TLD server.
5. The resolver queries the TLD server, which knows the Authoritative Name Server for the domain given and return the address of the domain's authoritative server to the DNS.
6. Then DNS resolver contacts the Authoritative Name server for the domain given. And authoritative server responds with the IP address associated with the domain.
7. The DNS resolver caches the IP address for future requests and sends the Ip address back to the browser.
8. The browser uses the provided IP address to access the web server associated with the domain.
9. The web server responds by sending the requested content, and the browser displays the website to the user.

Configuring a DNS Server (BIND)

1. Install DNS server software package.

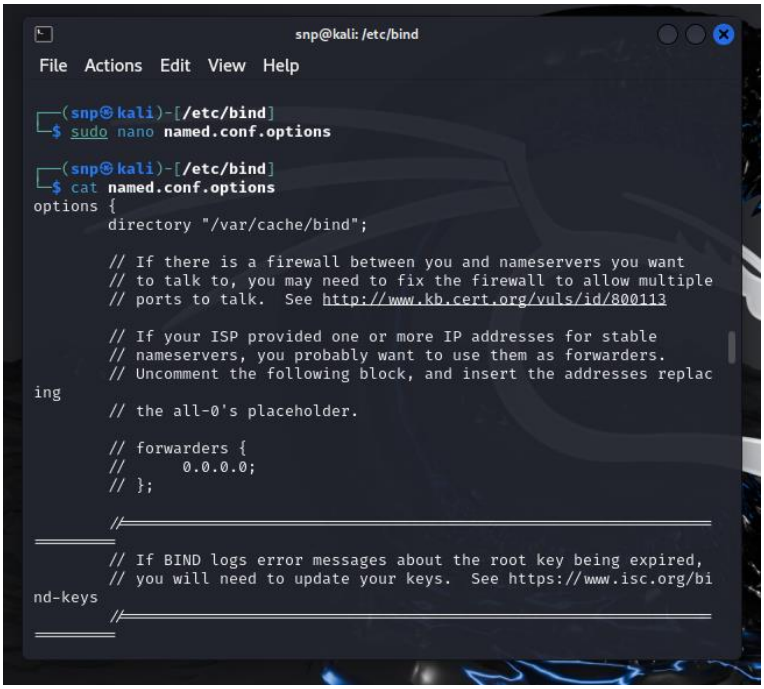
   BIND (Berkeley Internet Name Domain) is the most common DNS server software.

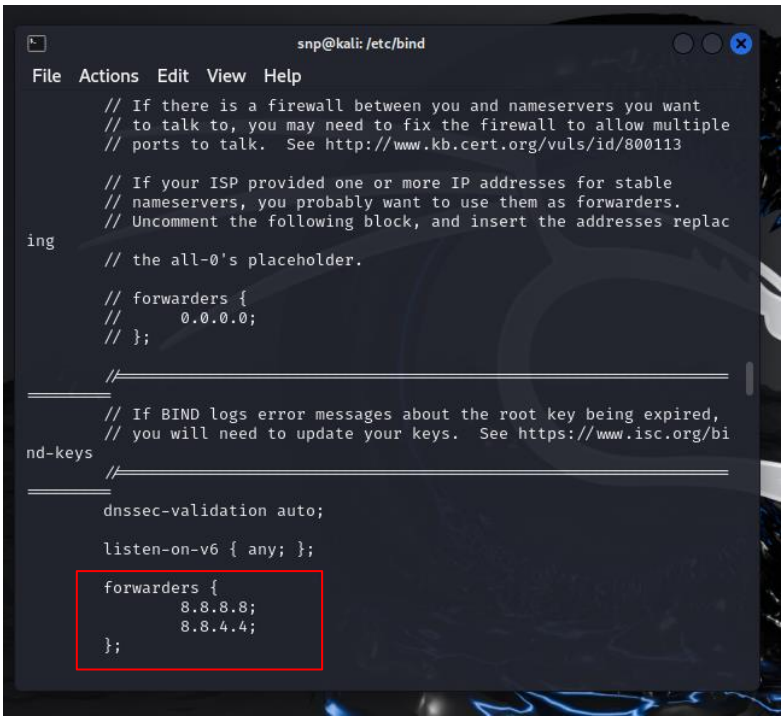


2. Configure BIND

Configure named.conf.options – This file contains global server options, including forwarders.

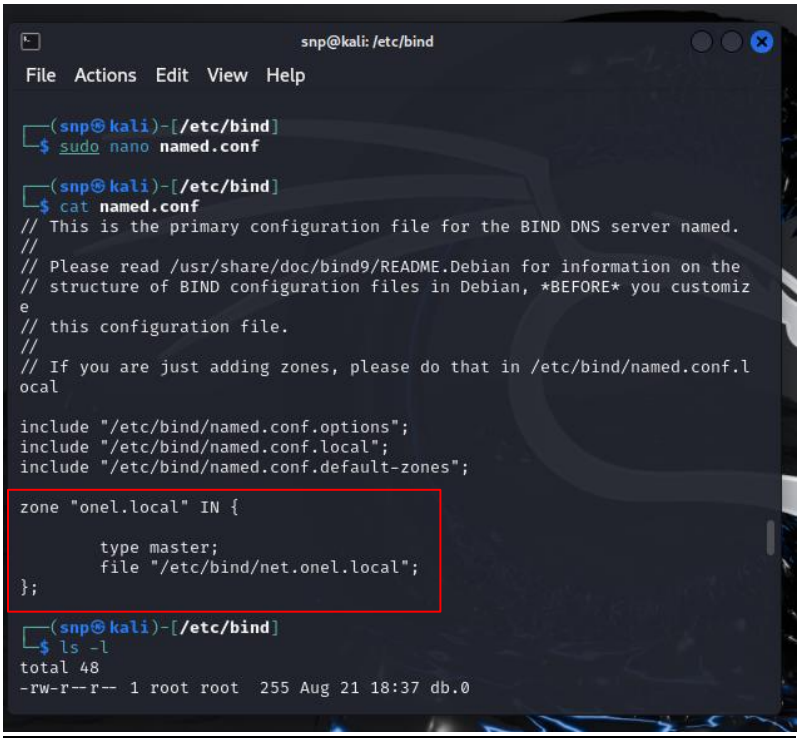
- Forwarders - Specifies upstream DNS servers that BIND should forward queries to if it does not have the answer locally. These are usually our ISP's DNS servers or public DNS servers (Google DNS) [1].





Configure named.conf

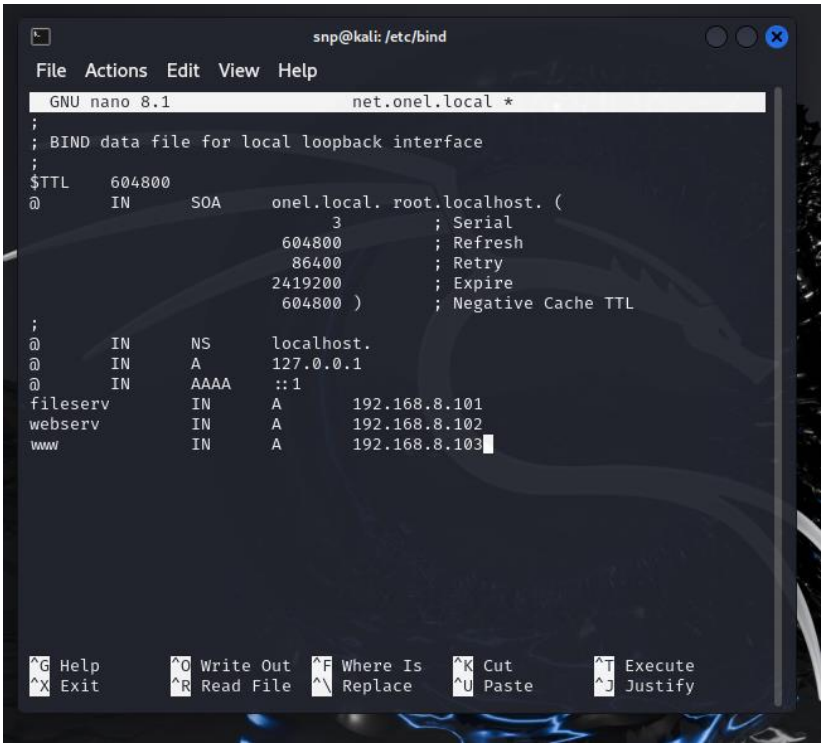
The named.conf file is the main configuration file for BIND (Berkeley Internet Name Domain), a popular DNS server software. This file defines the overall behavior of the DNS server, including the zones it manages, access control settings, logging, and various server options [1].



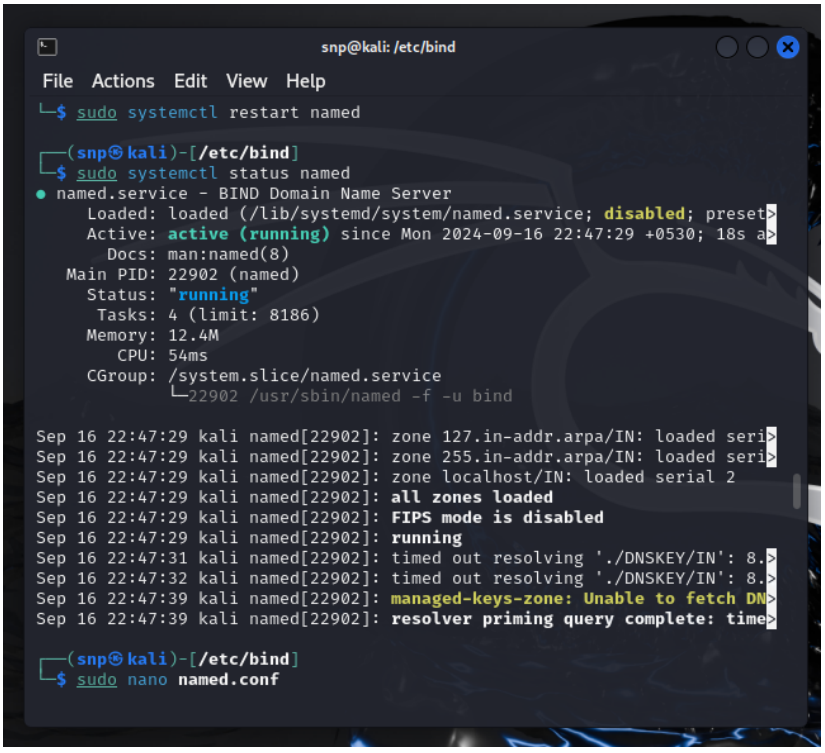
Zone - The zone section is one of the most important parts of named.conf. This is where you define the DNS zones that the server is responsible for.

Create Zone File

Zone files are simple text files that follow a specific format. They contain **resource records** (RRs) that map domain names to various types of information (such as IP addresses, mail servers, etc.). Each zone file begins with **SOA (Start of Authority)**, which defines administrative information about the zone, followed by various resource records [1].

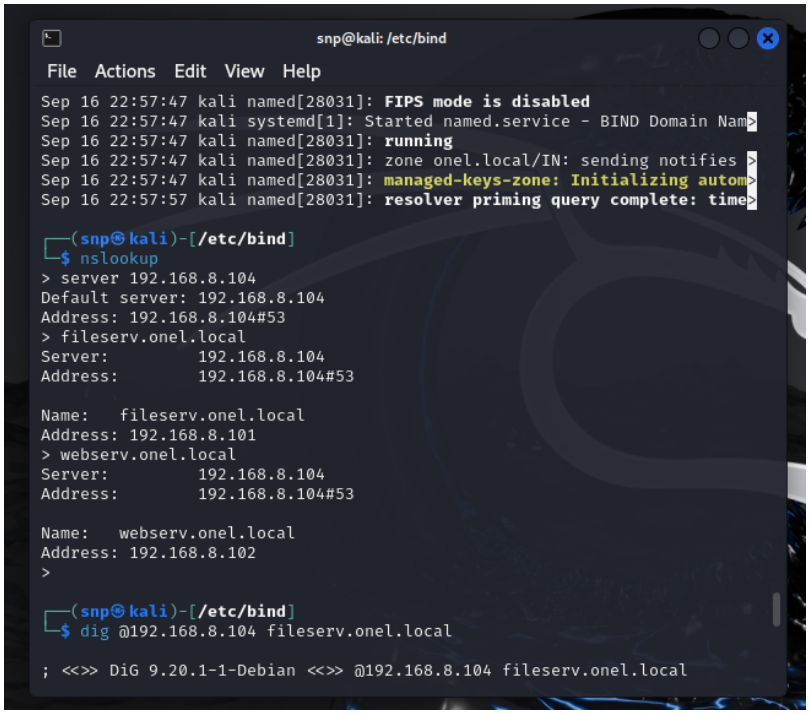


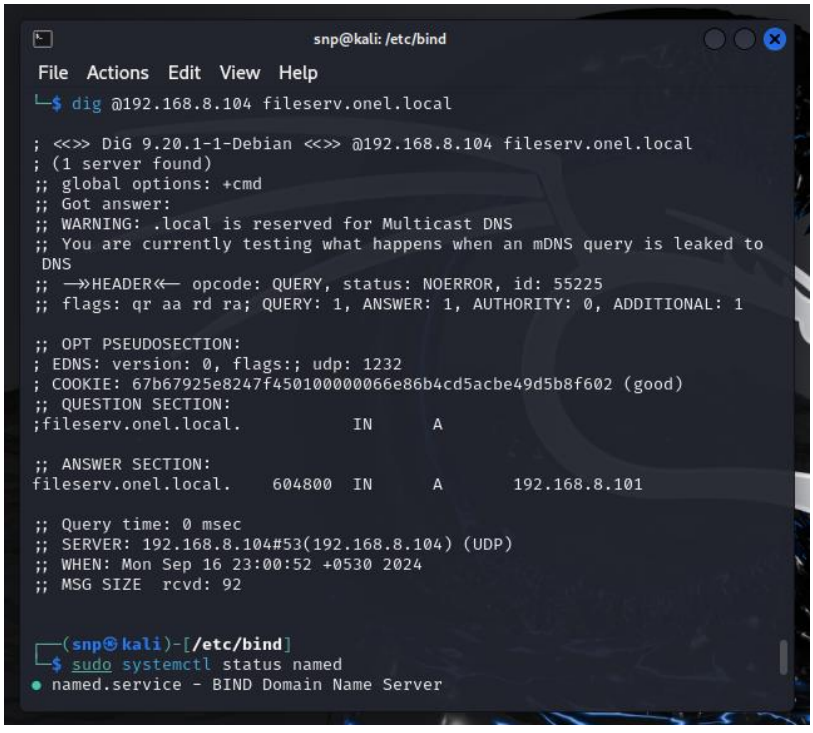
3. Enable and Restart BIND



4. Test the DNS Server

Use dig or nslookup to test if the DNS server is responding correctly.





2.3 Understanding and Configuring NTP

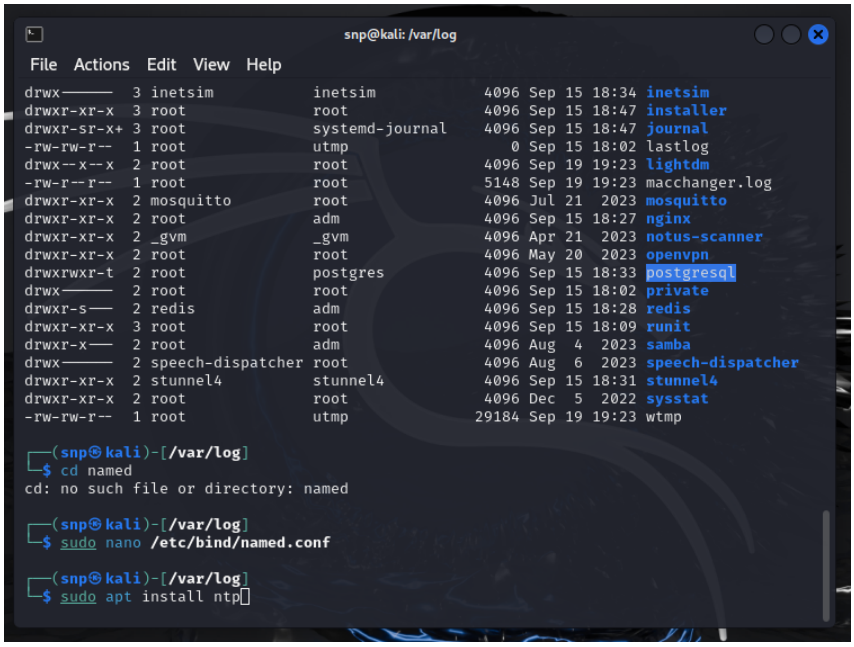
What is NTP

Network Time Protocol NTP is a protocol that synchronizes the time of computers and network devices over a network. By using a hierarchy of time sources, NTP makes it possible for all participating systems to maintain a consistent time for logging, transaction, and system operations applications

Configuring a NTP Server

1. Install the NTP Software

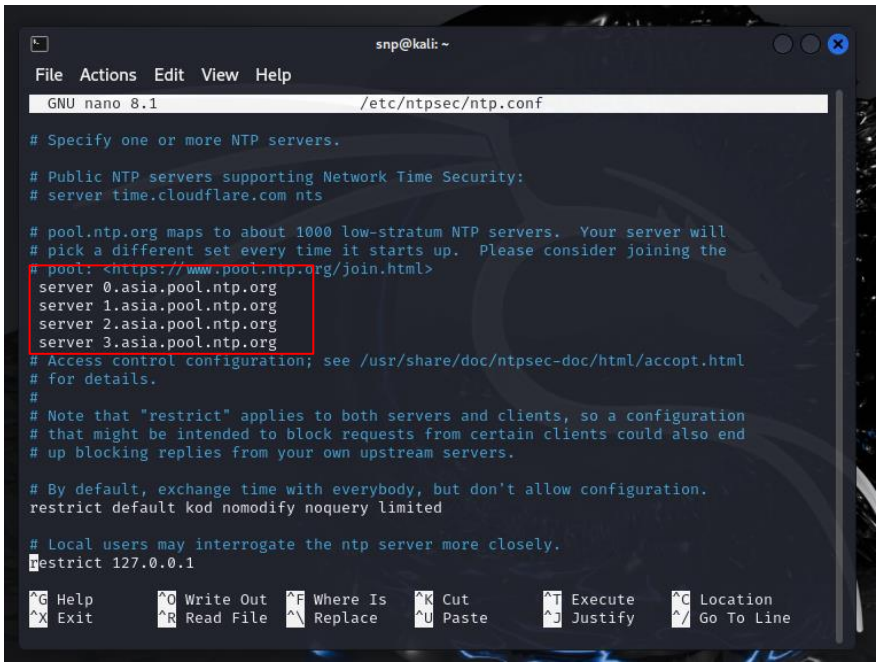
Install the NTP package using your distribution's package manager



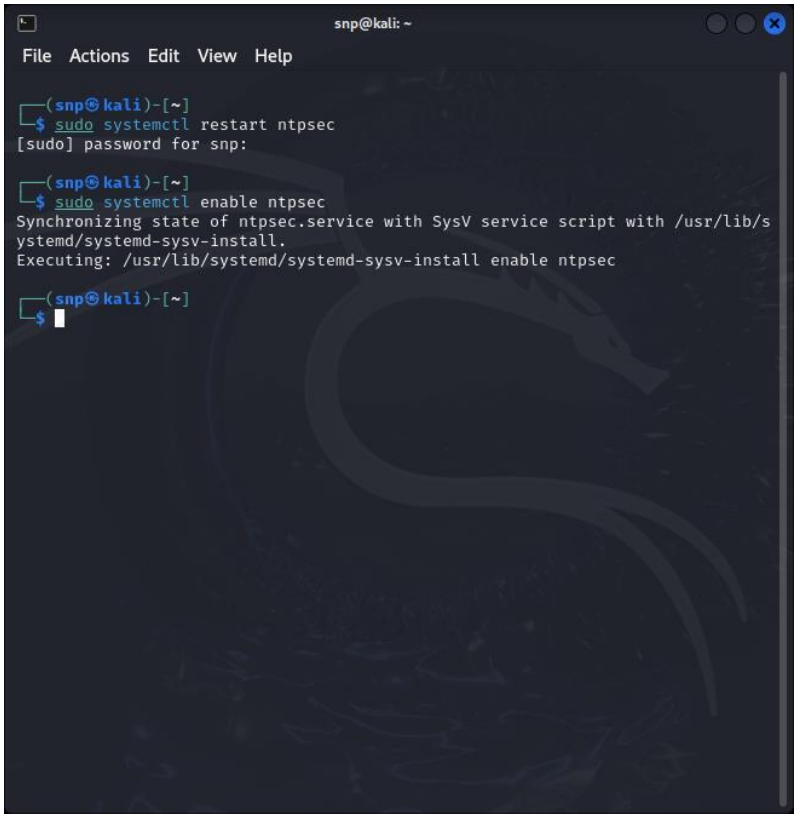
2. Configure NTP Server

The primary configuration file for the NTP server is /etc/ntp.conf .

The default configuration usually contains public NTP servers. We can modify these by replacing the server entries with our preferred NTP servers.

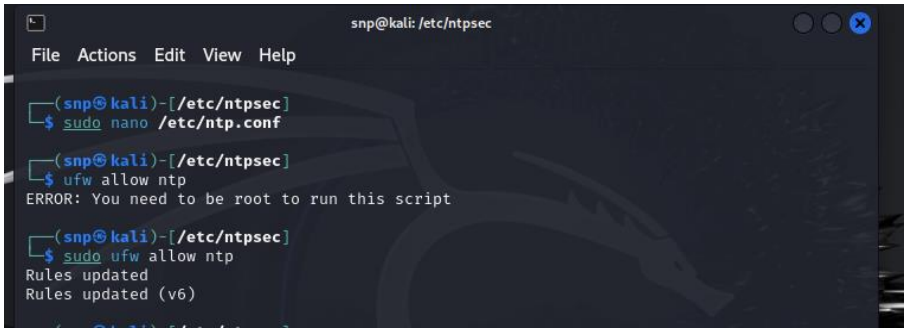


3. Restart and Enable the NTP Service.

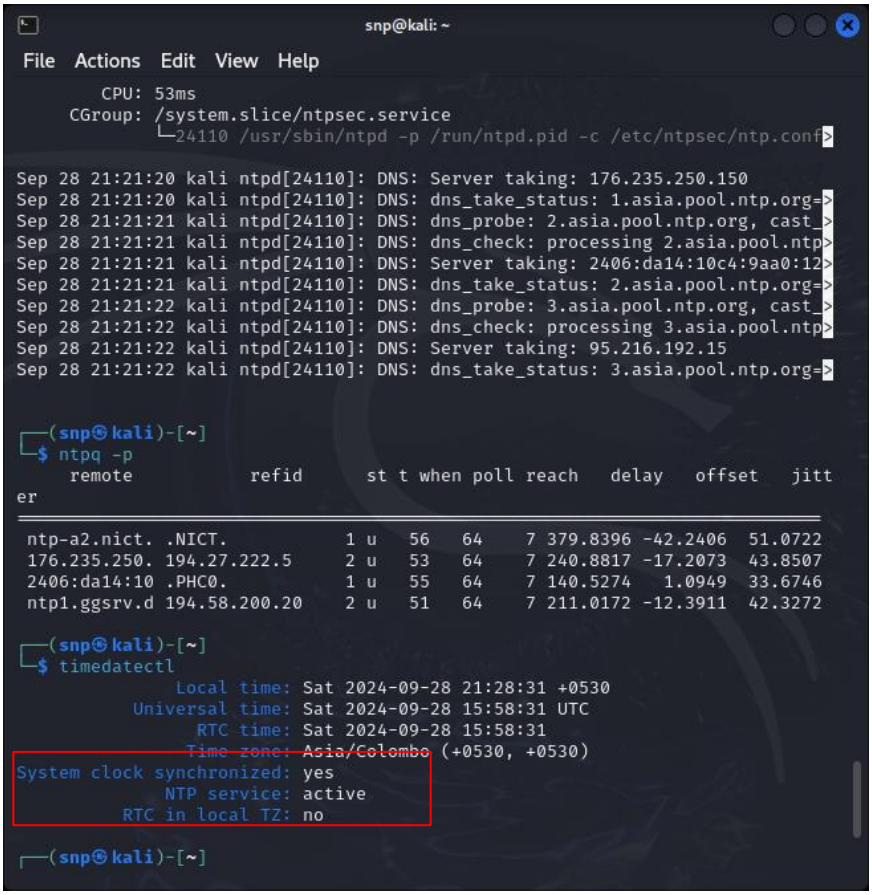


4. Open NTP port in firewall.

NTP uses port 123 (UDP) for communication. Make sure this port is open in firewall.



5. Verify NTP Server Status

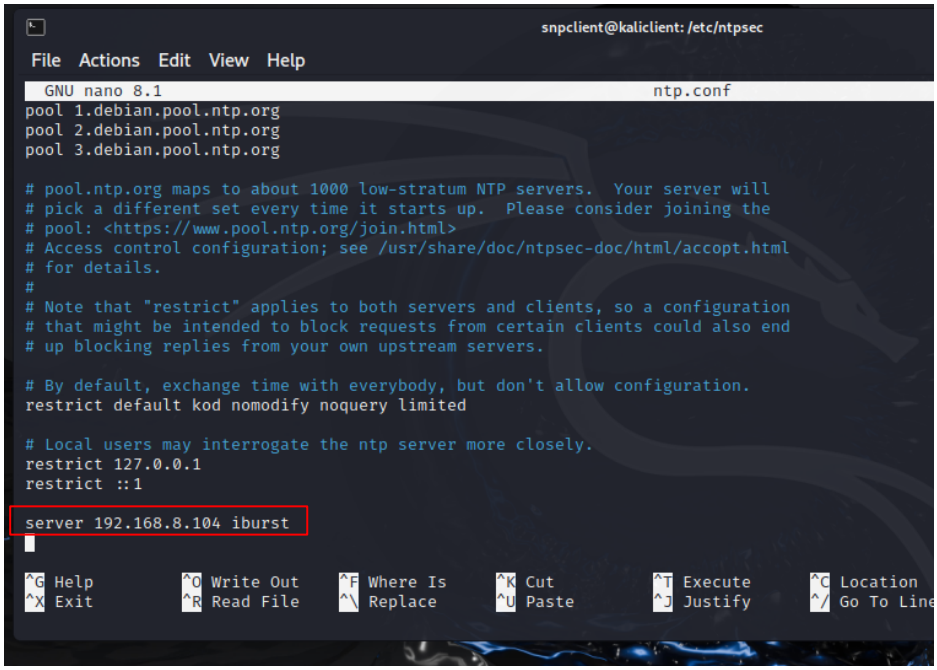


Configure NTP Clients

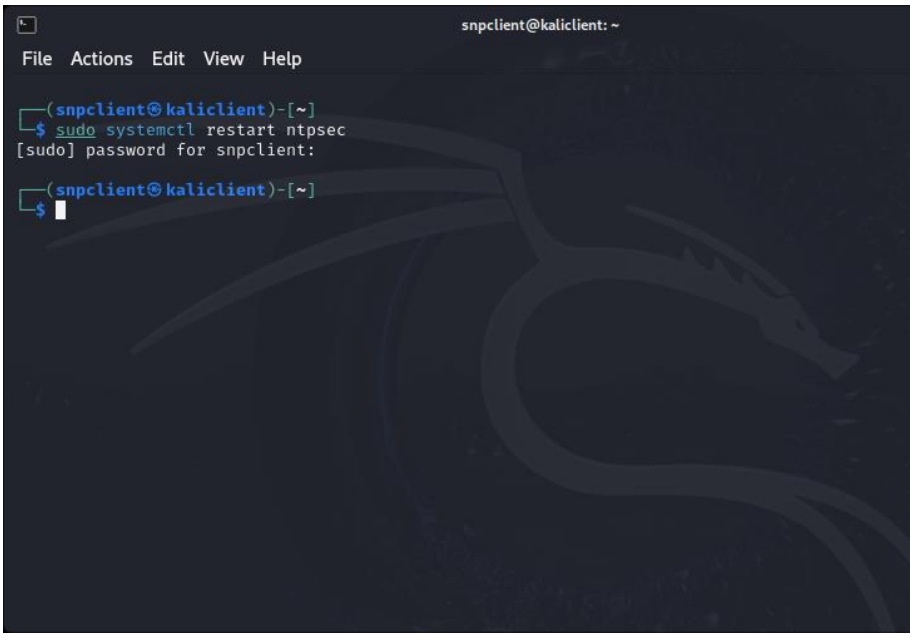
A separate virtual machine is created in Oracle VirtualBox specifically for the client system. This VM was configured to use NTP Server running on another VM. Both the client and server VMs were connected to the same virtual network within VirtualBox to allow communication between them.

1. Configure /etc/ntpsec/ntp.conf

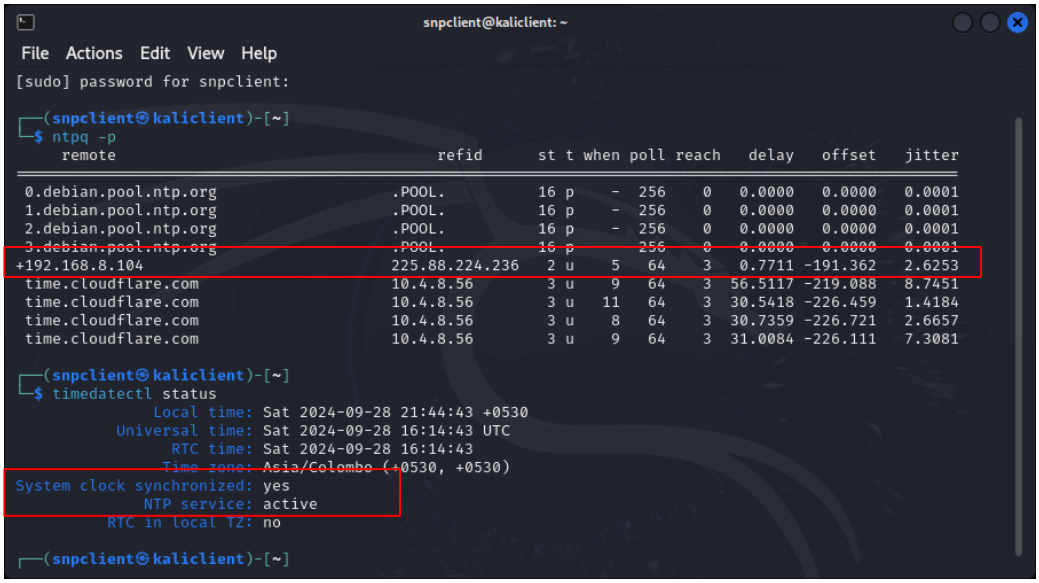
Edit and specify your NTP server running on host machine.



2. Restart NTP on client machine



3. Verify Client Synchronization


snpclient@kaliclient: ~
File Actions Edit View Help
[sudo] password for snpclient:
(snpclient㉿kaliclient)-[~]
$ ntpq -p
remote refid st t when poll reach delay offset jitter
0.debian.pool.ntp.org .POOL. 16 p - 256 0 0.0000 0.0000 0.0001
1.debian.pool.ntp.org .POOL. 16 p - 256 0 0.0000 0.0000 0.0001
2.debian.pool.ntp.org .POOL. 16 p - 256 0 0.0000 0.0000 0.0001
3.debian.pool.ntp.org .POOL. 16 p - 256 0 0.0000 0.0000 0.0001
+192.168.8.104 225.88.224.236 2 u 5 64 3 0.7711 -191.362 2.6253
time.cloudflare.com 10.4.8.56 3 u 9 64 3 56.5117 -219.088 8.7451
time.cloudflare.com 10.4.8.56 3 u 11 64 3 30.5418 -226.459 1.4184
time.cloudflare.com 10.4.8.56 3 u 8 64 3 30.7359 -226.721 2.6657
time.cloudflare.com 10.4.8.56 3 u 9 64 3 31.0084 -226.111 7.3081
(snpclient㉿kaliclient)-[~]
$ timedatectl status
Local time: Sat 2024-09-28 21:44:43 +0530
Universal time: Sat 2024-09-28 16:14:43 UTC
RTC time: Sat 2024-09-28 16:14:43
Time zone: Asia/Colombo (+0530, +0530)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
(snpclient㉿kaliclient)-[~]

3 . Shell Scripting and Security

3.1 Shell Scripting

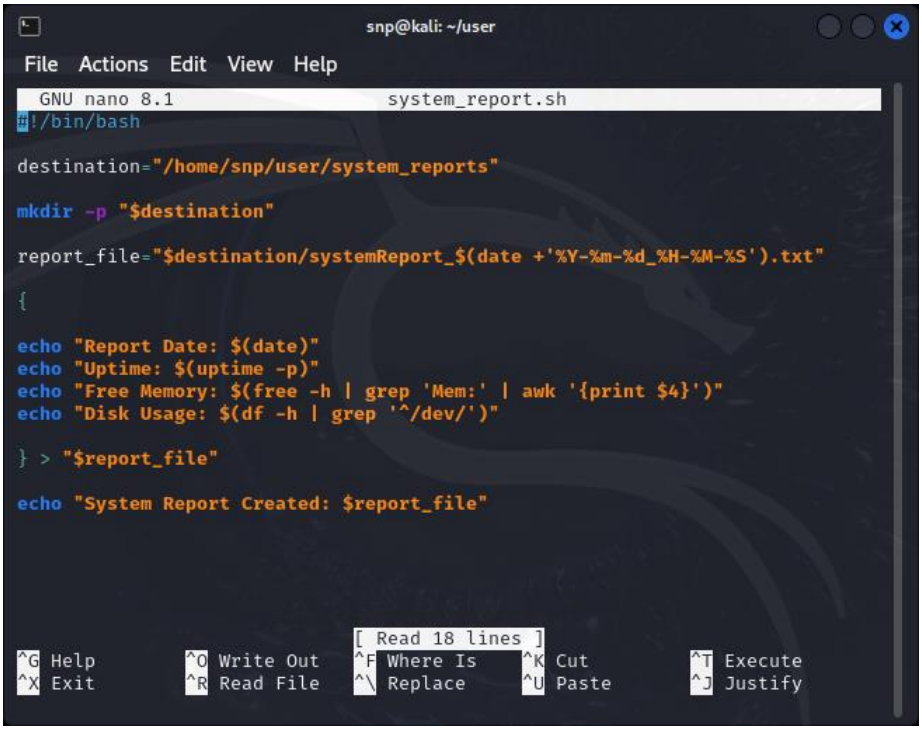
Automating a System Report

The script automates the process of generating a system report that includes:

- The current date and time.
- The system's uptime.
- The amount of free memory.
- The disk usage for mounted file systems.

It saves this information in a text file with a filename that includes the current date, ensuring that each report is unique and easily identifiable.

```
#!/bin/bash

destination="/home/snp/user/system_reports"

mkdir -p "$destination"

report_file="$destination/systemReport_$(date +'%Y-%m-%d_%H-%M-%S').txt"

{

echo "Report Date: $(date)"

echo "Uptime: $(uptime -p)"

echo "Free Memory: $(free -h | grep 'Mem:' | awk '{print $4}')"

echo "Disk Usage: $(df -h | grep '^/dev/')"

} > "$report_file"

echo "System Report Created: $report_file"
```

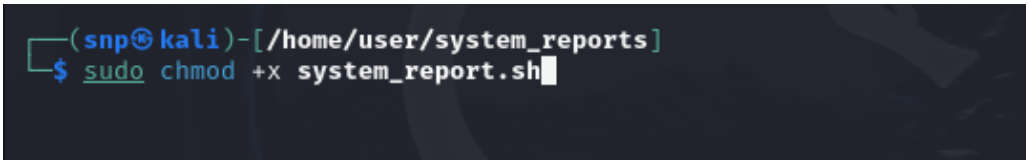


Automating Task with crontab

Crontab is a Unix-based utility that allows users to schedule tasks (known as "cron jobs") to run automatically at specified intervals. This is especially useful for automating repetitive tasks, such as running scripts, generating reports, or performing system maintenance.

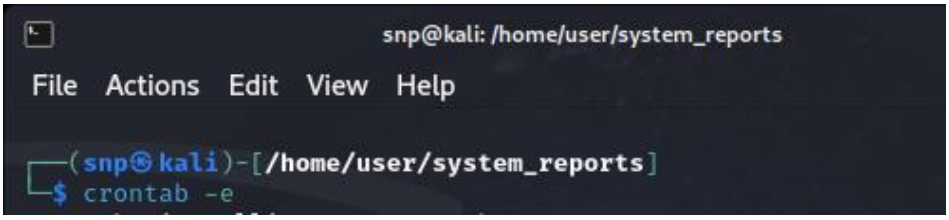
1. First, we need to ensure it is executable by running.

sudo chmod +x system_report.sh



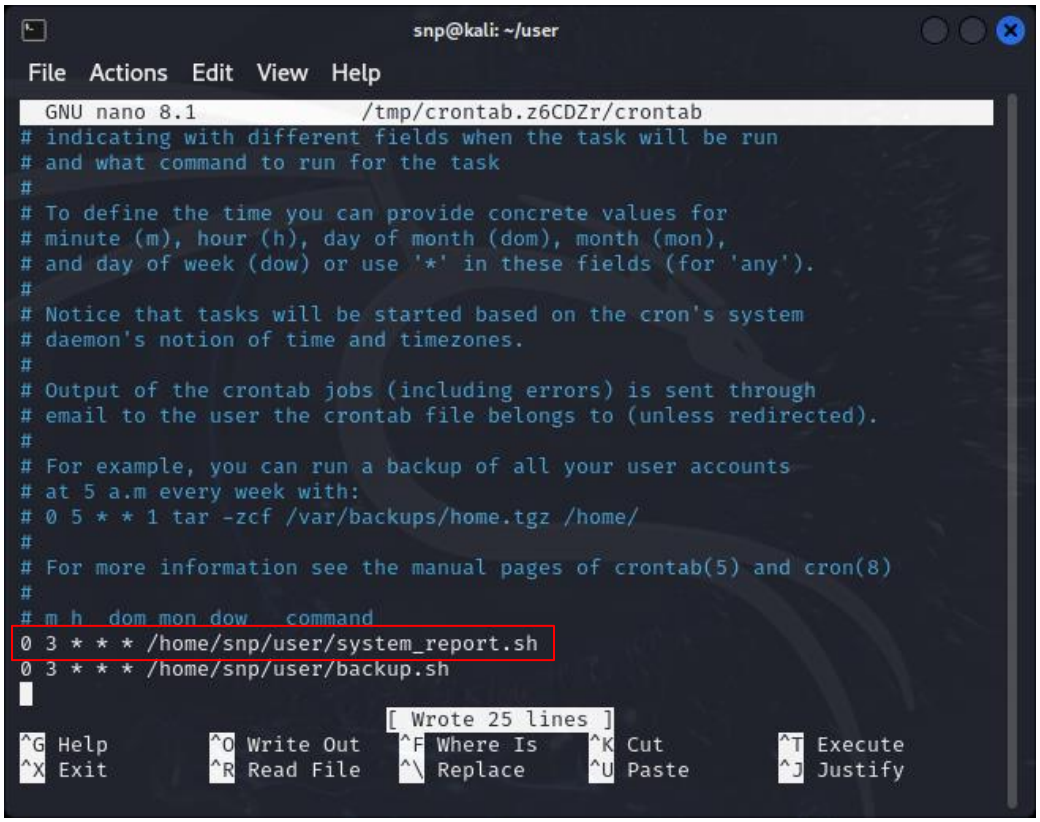
2. Open crontab file.

cron tab -e

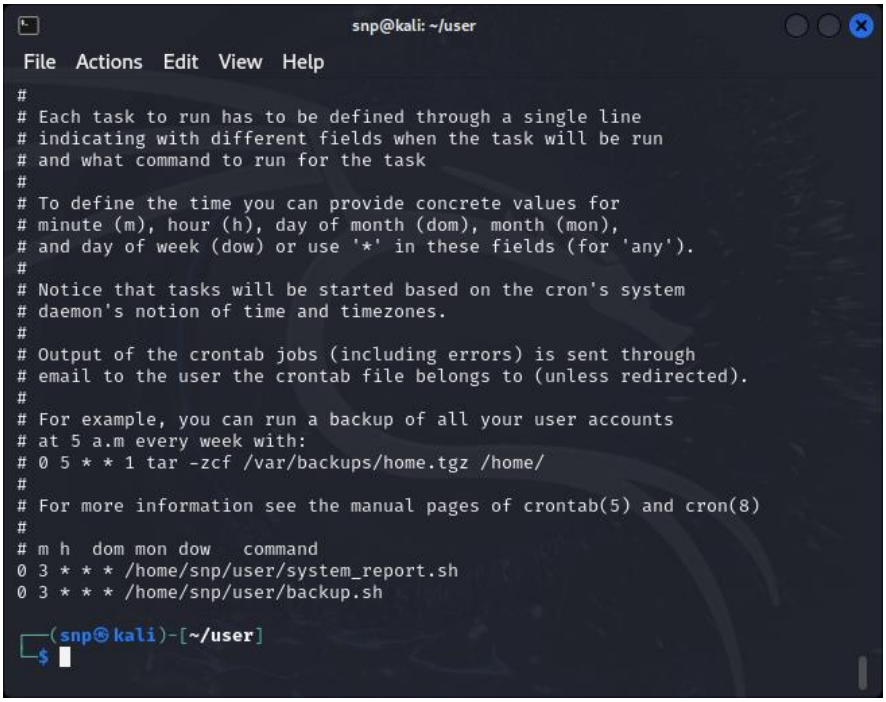


3. Adding a cron job.

0 3 * * * /home/snp/user/system_reports/system_report.sh



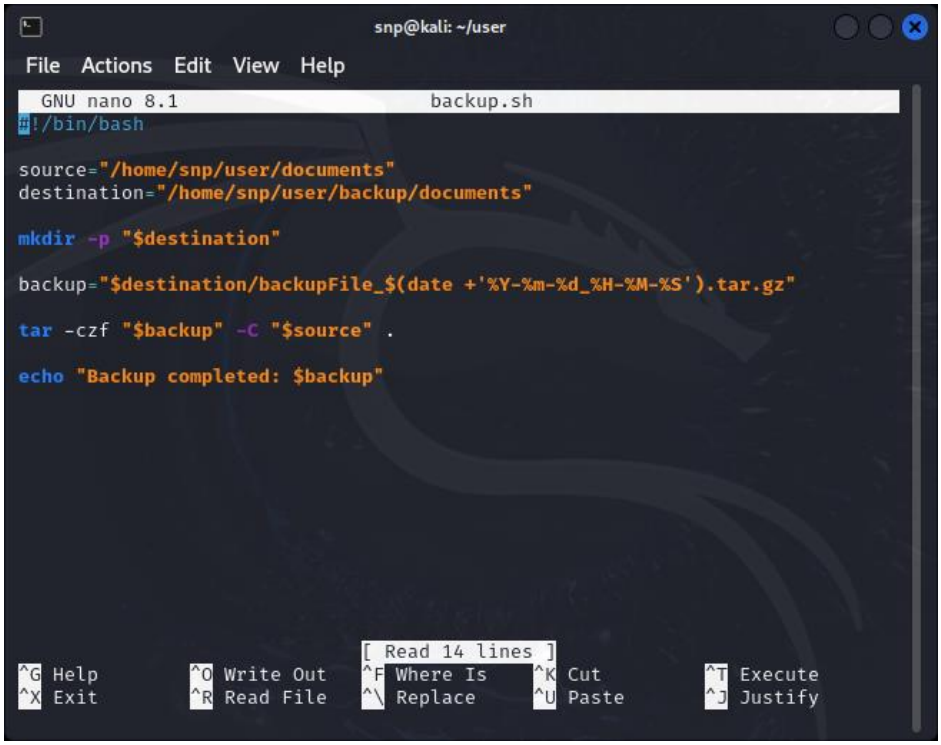
4. Save and exit.
5. Check scheduled jobs.



Automating Backup of a critical Directory

This Bash script automates the process of backing up files from a specified source directory to a destination directory in a compressed format.

```
#!/bin/bash

source="/home/snp/user/documents"

destination="/home/snp/user/backup/documents"

mkdir -p "$destination"

backup="$destination/backupFile_$(date +'%Y-%m-%d_%H-%M-%S').tar.gz"

tar -czf "$backup" -C "$source" .

echo "Backup completed: $backup"
```



To automate the task, We'll setup cron job as previous task.

1. Make script executable.

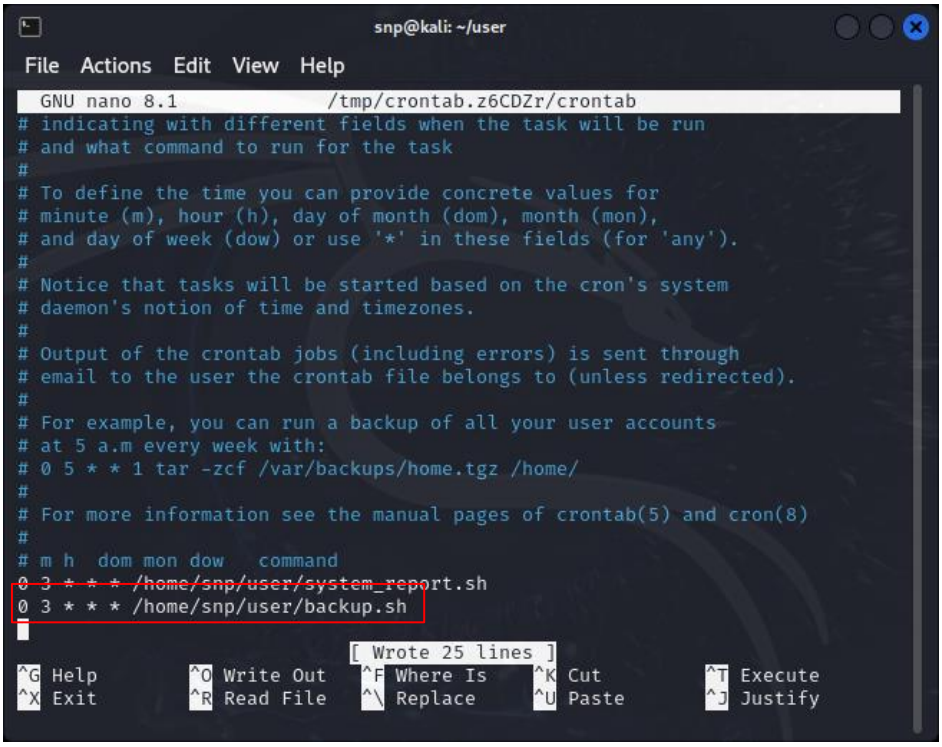
   sudo chmod +x backup.sh

2. Open cron tab file

   crontab -e

3. Adding a cron job.

   0 3 * * * /home/snp/user/backup.sh



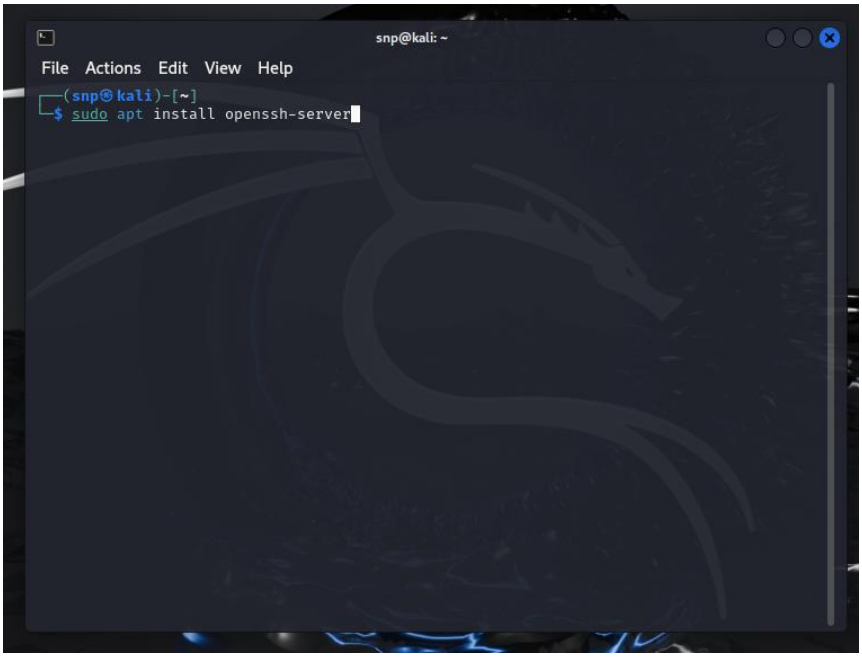
4. Save and exit.

3.2 SSH (Secure Shell) Configuration

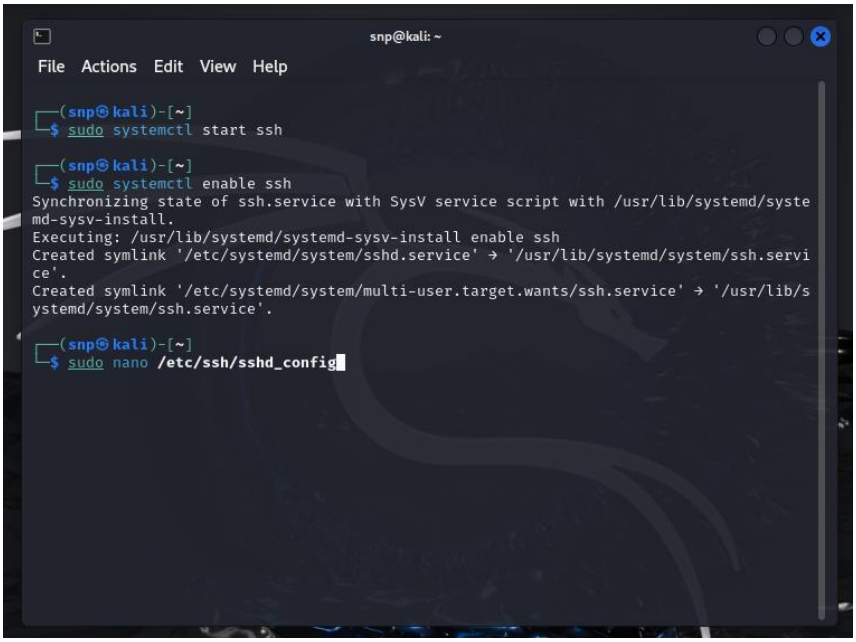
Secure Shell (SSH) is a cryptographic network protocol used to securely access and manage network devices and servers over an unsecured network. It provides a secure channel for communication between a client and a server, allowing users to execute commands, transfer files, and manage remote systems safely.

Configuring SSH Server for Secure Remote Access

1. Install SSH Server



2. Start and Enable SSH.



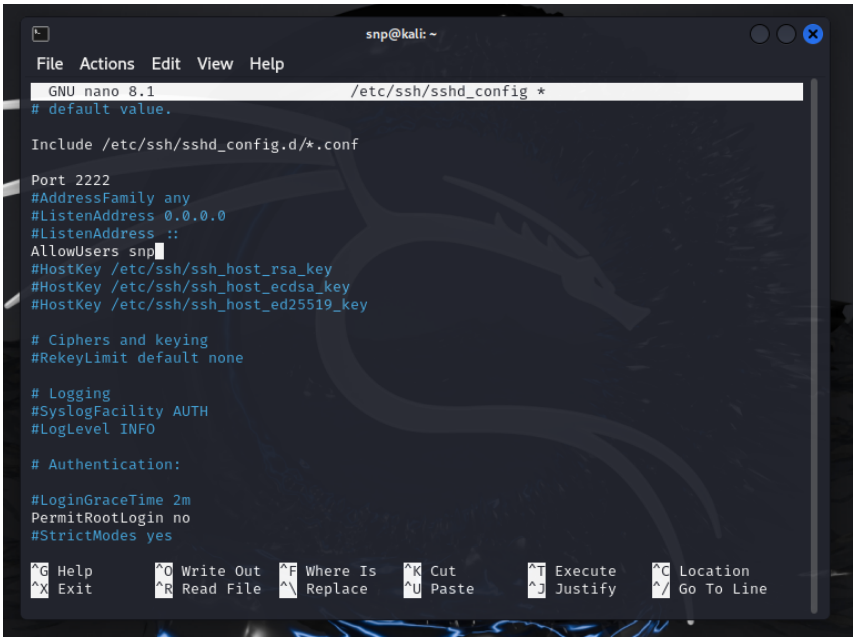
3. Configure SSH Settings

This file controls the behavior of the SSH server.

Port - specifies the port number.

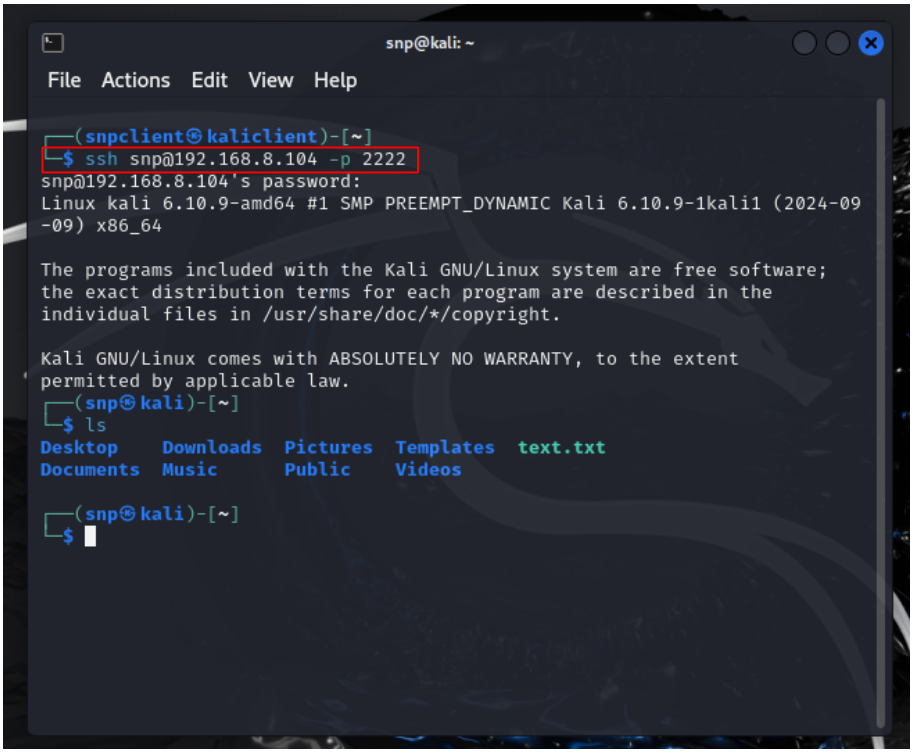
AllowUsers - restricts which users or groups can log in via SSH.

PermitRootLogin – Controls whether root login is allowed.



4. Test SSH Connection

Attempt to connect to the remote machine using SSH.



3.3 Network Traffic Management Using iptables and ACLs

iptables is a command-line utility used to configure the Linux kernel's netfilter framework, which acts as a packet filtering mechanism. It is commonly used to set up, maintain, and inspect the rules of a firewall in Linux-based systems.

An **Access Control List (ACL)** is a set of rules that defines which users or system processes have access to particular resources and what operations they are allowed to perform. ACLs are commonly used in networking, file systems, and system security to regulate permissions.

Web Server Security

1. Set the default policy to drop all incoming traffic.

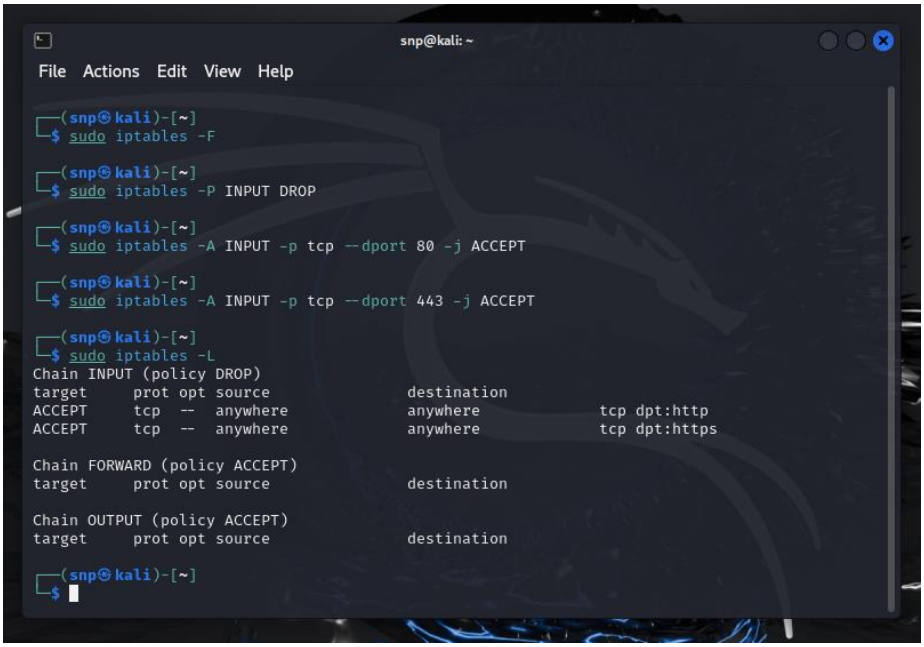
```
sudo iptables -P INPUT DROP
```

2. Allow incoming HTTP (port 80) traffic

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

3. Allow incoming HTTPS (port 443) traffic

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



4. Allow established and related traffic

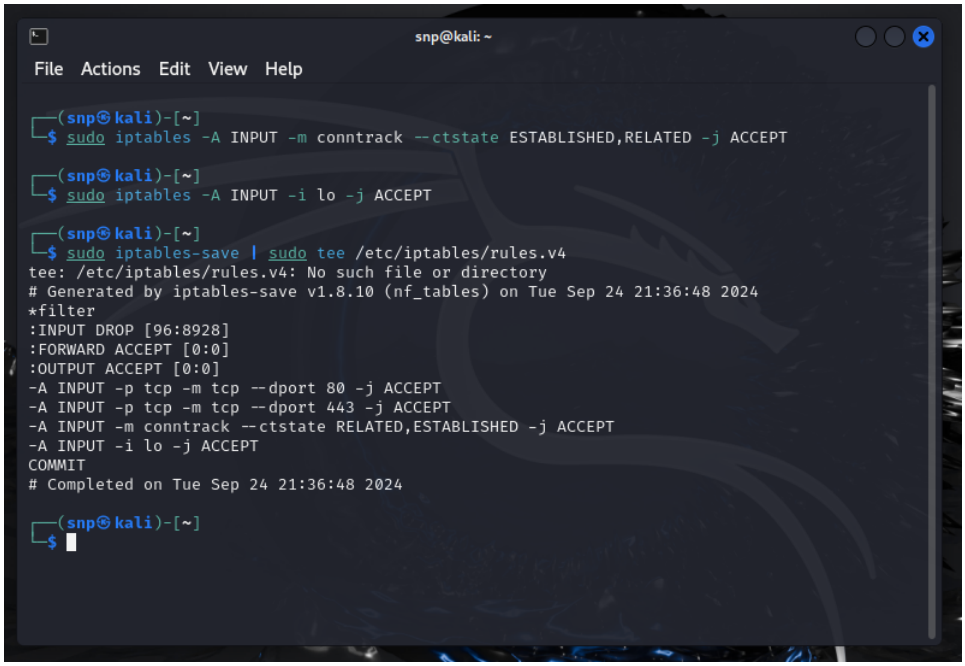
sudo iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

5. Allow incoming traffic on the loopback interface

sudo iptables -A INPUT -i lo -j ACCEPT

6. Save the rules.

sudo iptables-save | sudo tee /etc/iptables/rules.v4



Remote Administration Access

1. Allow incoming SSH (port 22) traffic from specific trusted IP addresses

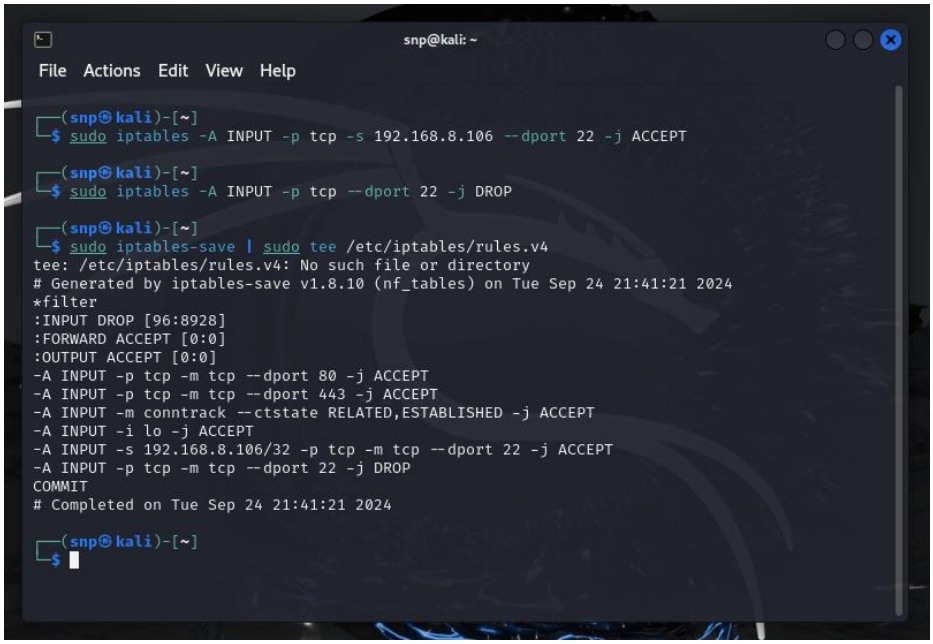
sudo iptables -A INPUT -p tcp -s 192.168.8.106 --dport 22 -j ACCEPT

2. Block all other SSH access attempts

sudo iptables -A INPUT -p tcp --dport 22 -j DROP

3. Save the rule

sudo iptables-save | sudo tee /etc/iptables/rules.v4



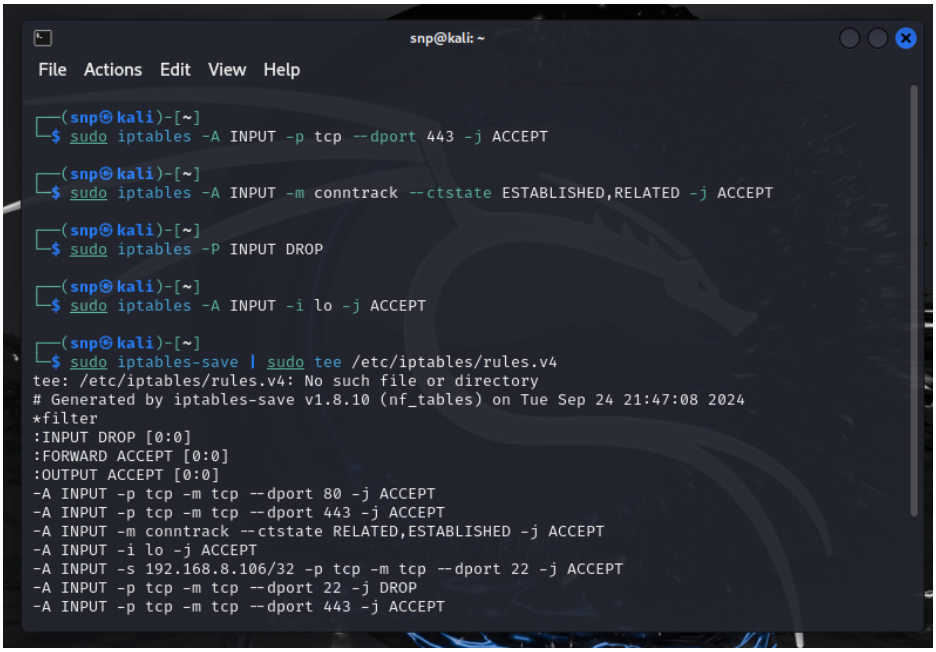
Application Specific Rules

1. Allow HTTPS traffic for video conferencing apps (port 443)

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

2. Save the rules

sudo iptables-save | sudo tee /etc/iptables/rules.v4



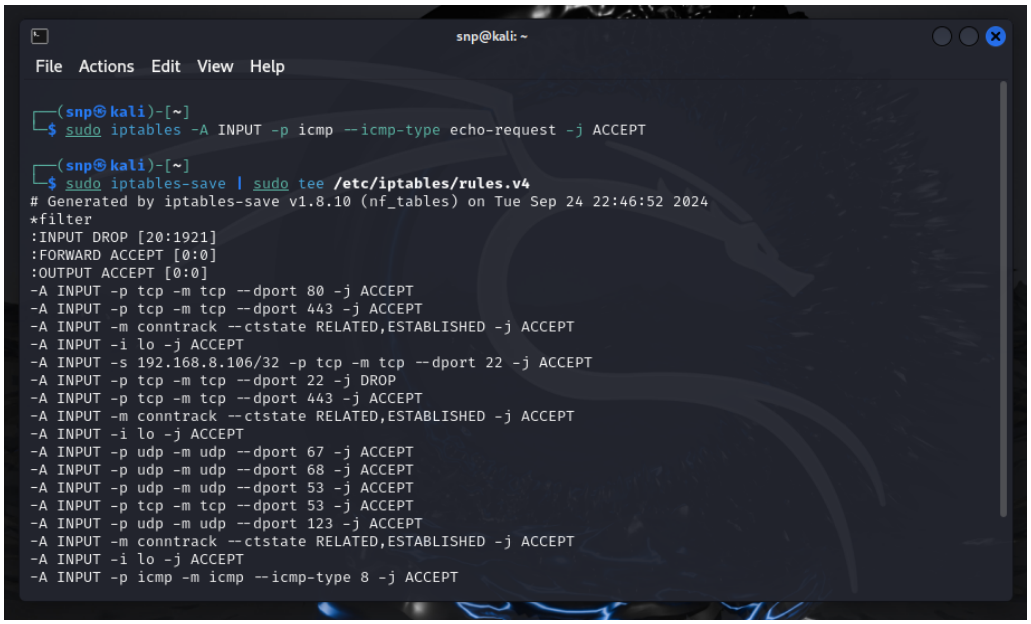
Allowing ICMP Echo Requests (Ping)

1. Allow ICMP Echo requests

sudo iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

2. Save the rule

sudo iptables-save | sudo tee /etc/iptables/rules.v4



Printer Server Security

1. Allow printer traffic from specific ip/ ip range

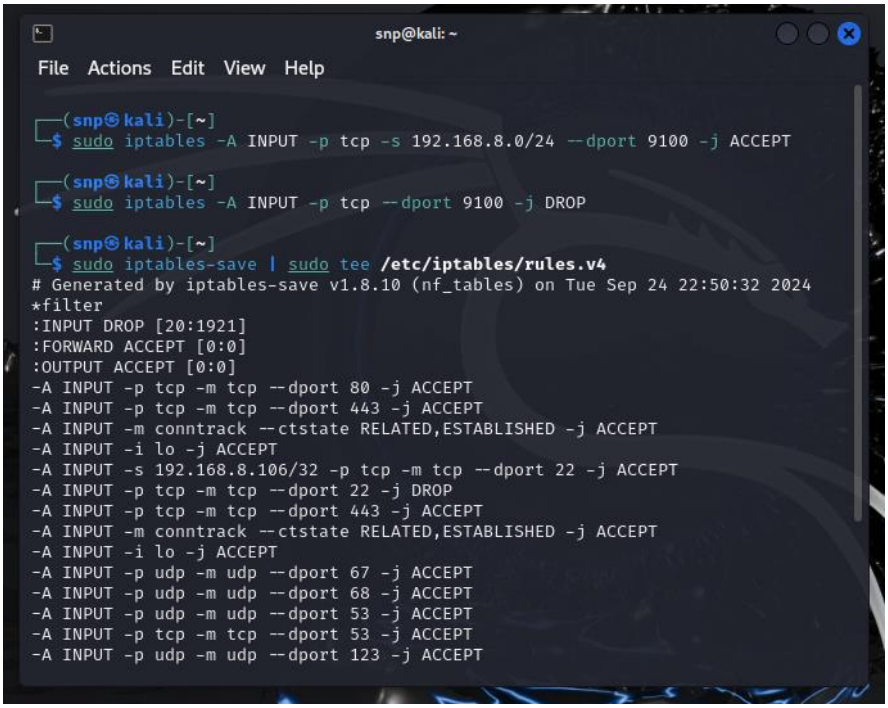
sudo iptables -A INPUT -p tcp -s 192.168.8.0/24 --dport 9100 -j ACCEPT

2. Drop all incoming TCP traffic directed to port 9100

sudo iptables -A INPUT -p tcp --dport 9100 -j DROP

3. Save the rules

sudo iptables-save | sudo tee /etc/iptables/rules.v4

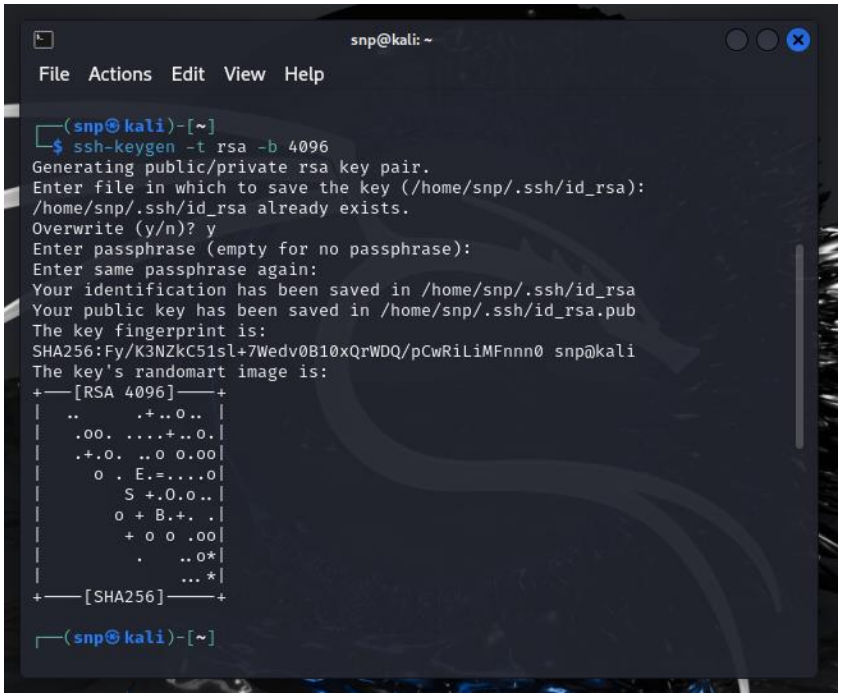

snp@kali: ~
File Actions Edit View Help
┌──(snp㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp -s 192.168.8.0/24 --dport 9100 -j ACCEPT
┌──(snp㉿kali)-[~]
└─$ sudo iptables -A INPUT -p tcp --dport 9100 -j DROP
┌──(snp㉿kali)-[~]
└─$ sudo iptables-save | sudo tee /etc/iptables/rules.v4
Generated by iptables-save v1.8.10 (nf_tables) on Tue Sep 24 22:50:32 2024
*filter
:INPUT DROP [20:1921]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 192.168.8.106/32 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j DROP
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p udp -m udp --dport 67 -j ACCEPT
-A INPUT -p udp -m udp --dport 68 -j ACCEPT
-A INPUT -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m udp --dport 123 -j ACCEPT

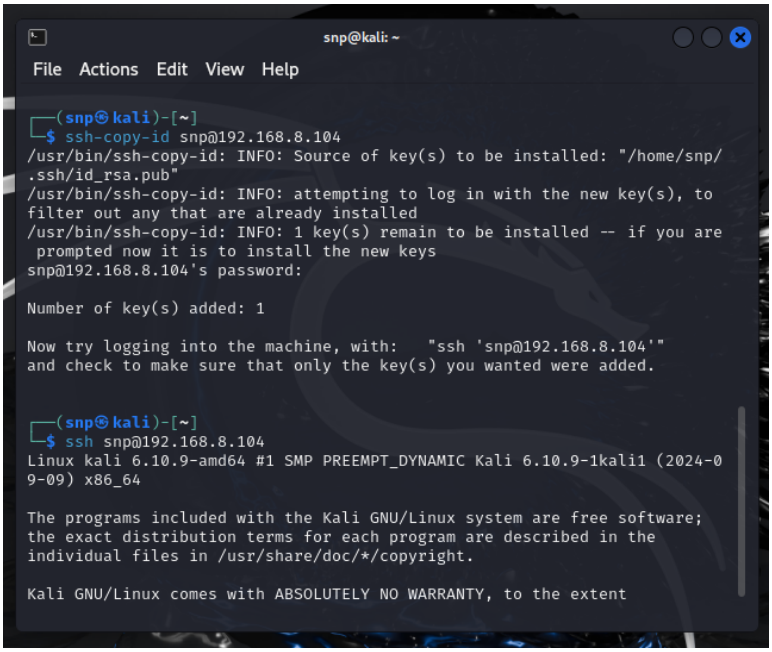
4. Best Practices for Network Interface Configuration Security

1. Ensure OpenSSH Server Security

Use SSH key pairs for authentication

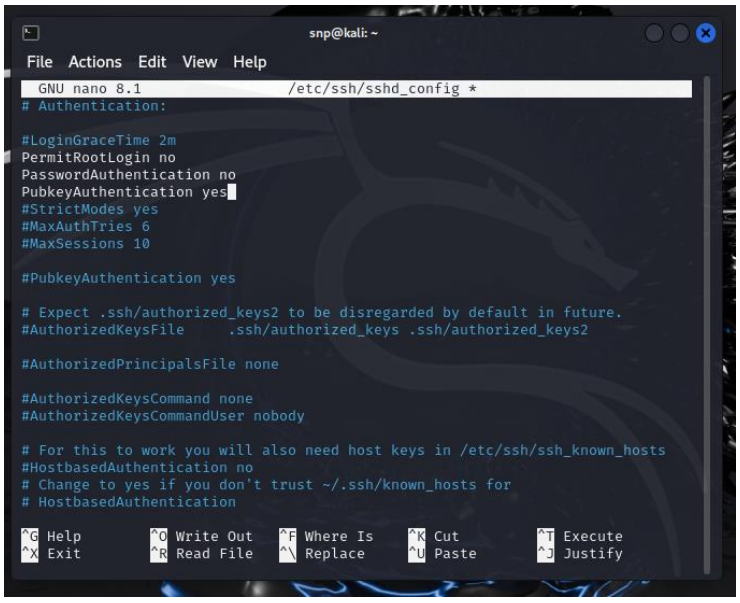
One of the best practices includes securing it using an authentication mechanism through SSH keys. The key pairs enable stronger ways of protecting remote access, as opposed to traditional password-based authentication. This implies the generation of a public-private key pair, where the public key is placed on the server and the private key is kept securely on the client machine. This will prevent brute-force attacks on the password but at the same time ensures that only the correct and authorized user, with his or her private key, gains access.





Disable root login

By default, external root access is turned on for all Linux machines. It leaves an open SSH security vulnerability to brute-force attacks by hackers. By disabling the server SSH root login, unauthorized people cannot take control of the system. An active root account allows an attacker to gain or guess the password of the root with full administrative privileges.

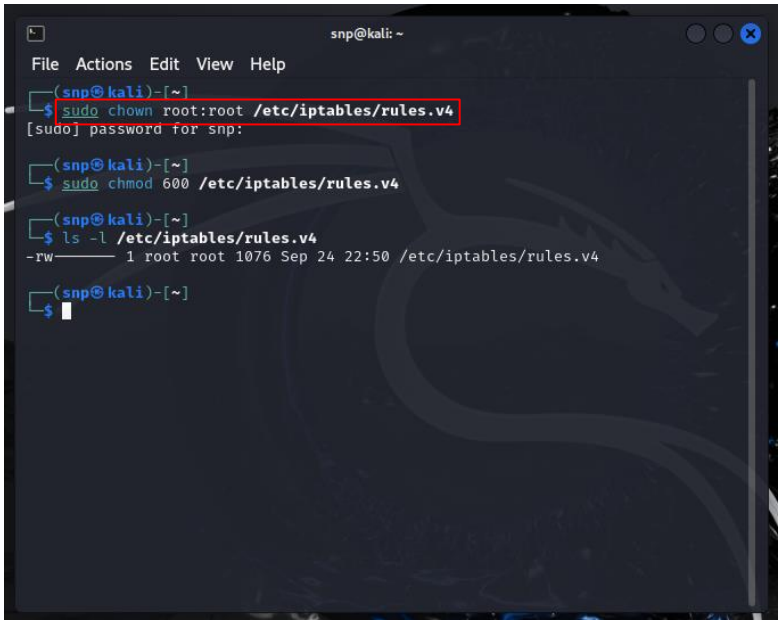


2. Securing the iptables Rules File: Ownership and Permissions

This file should be editable and owned by a root user only, in order to secure the iptables rules file. It is of utmost importance that unauthorized editing, which may open up security vulnerabilities, be blocked along with unexpected disclosure and loss of control over network traffic.

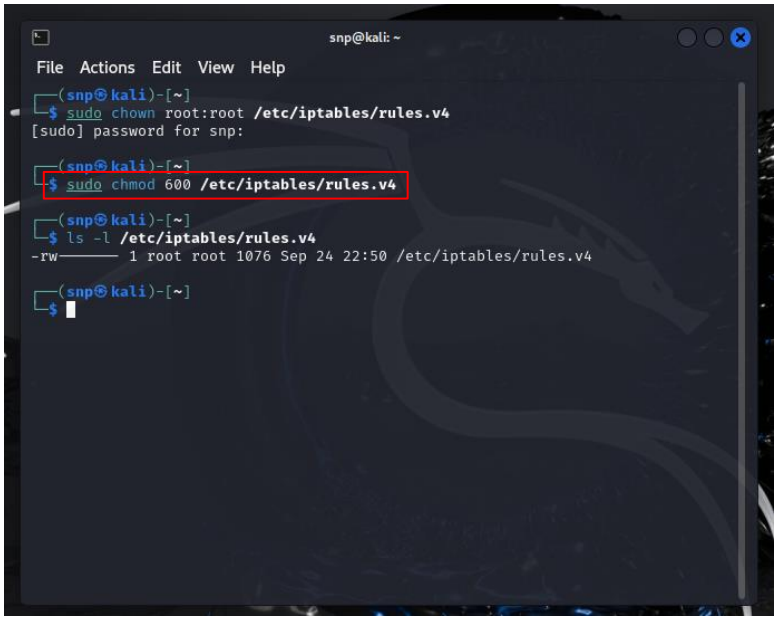
1. Ownership

Ensuring that the iptables rules file is owned by the root user prevents non-privileged users from having the ability to alter firewall settings.



2. Permissions

The rules file should have permissions that allow only the root user to read and write to it while denying all other users access.

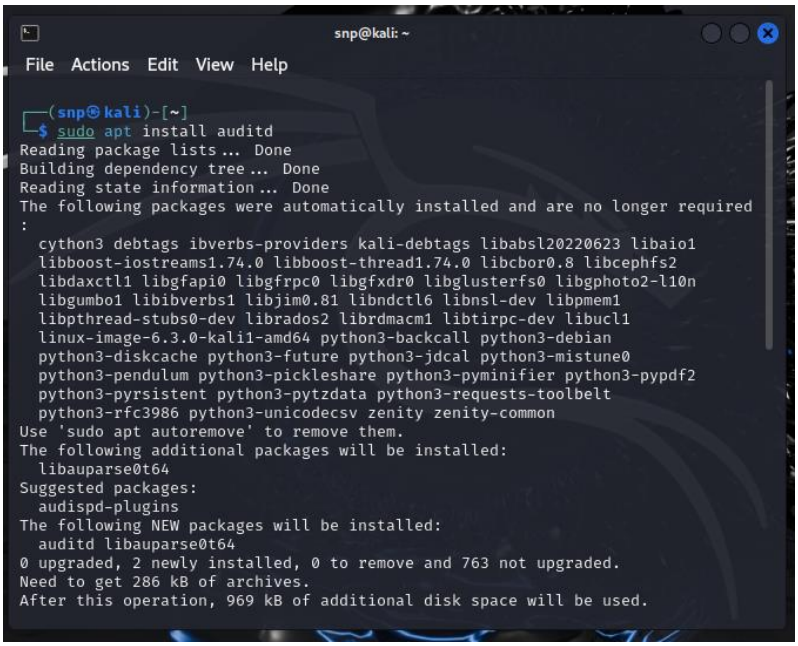


3. Implementing Auditing Software for Security of the iptables Rules File

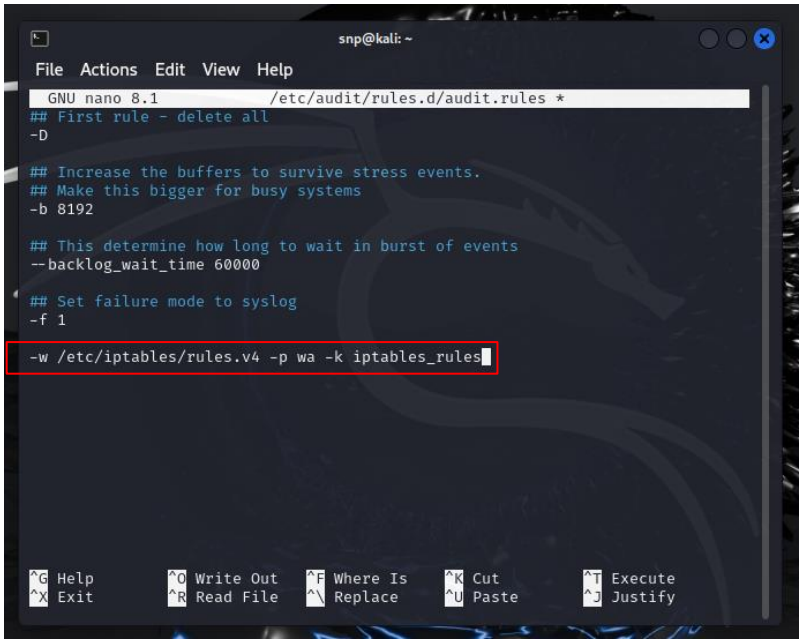
Setting up auditing software is majorly critical in monitoring the integrity of your iptables rules file in your network configurations. Auditing allows you to track changes made to a rules file. This gives good visibility into who made those changes, at what time they were made, and what exactly was changed. Such proactive detection helps in unauthorized alterations and further fortifies the general security posture of your Linux system.

Setting up Auditing with auditd

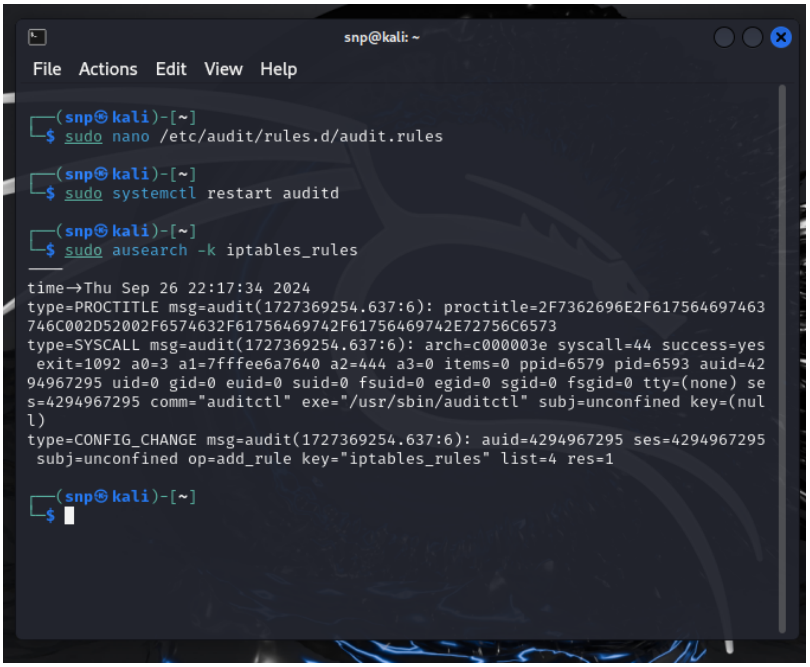
1. Install auditd



2. Add a monitoring rule

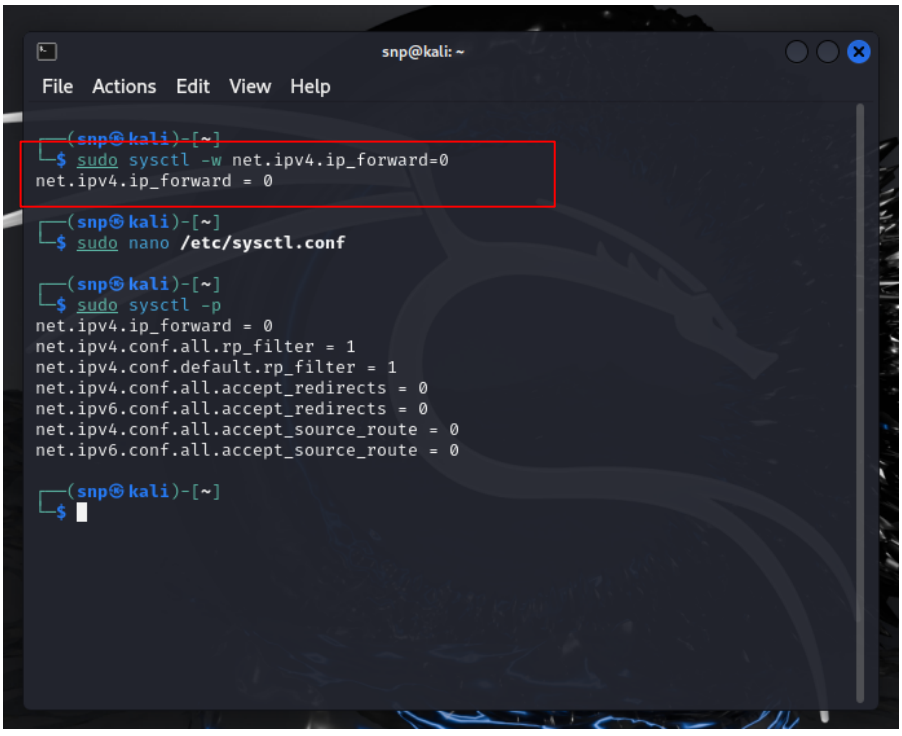


3. Restart auditd
4. View audit logs



4. Enhancing Security by Disabling IP Forwarding on Linux

Disabling IP forwarding is a critical security measure for Linux systems that do not require routing capabilities. By setting net.ipv4.ip_forward=0, you effectively prevent the system from forwarding packets between different network interfaces.



Positive implications for security by disable IP forwarding

- Reduced attack surface
- Prevention unauthorized Access
- Improved firewall efficiency
- Simplified incident response and monitoring

5. Transitioning from Dynamic to Static IP Addresses

Dynamic IP addresses are in use for convenience and ease of management on any network. However, there are a few major security risks associated with them that might even lead to revealing systems and networks.

Dynamic IP addressing can make an attacker's potential for an already assigned IP address impersonate users and devices. This impersonation may facilitate unauthorized access to sensitive resources, especially in systems using only verification of IP addresses as a means of authentication, thus raising a serious risk to security.

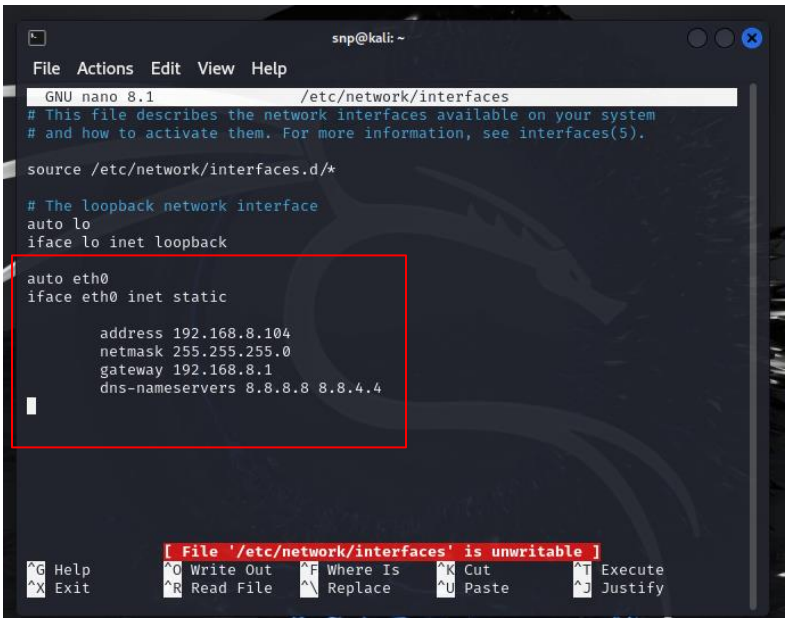
Setting a static IP address means it would assign a certain fixed IP address of a device on a network, so upon reboots or disconnections, it retains the same address. It is quite essential for devices like servers, printers, and networked storage devices when continuous access and identification over a network are needed.

Setting a static IP address

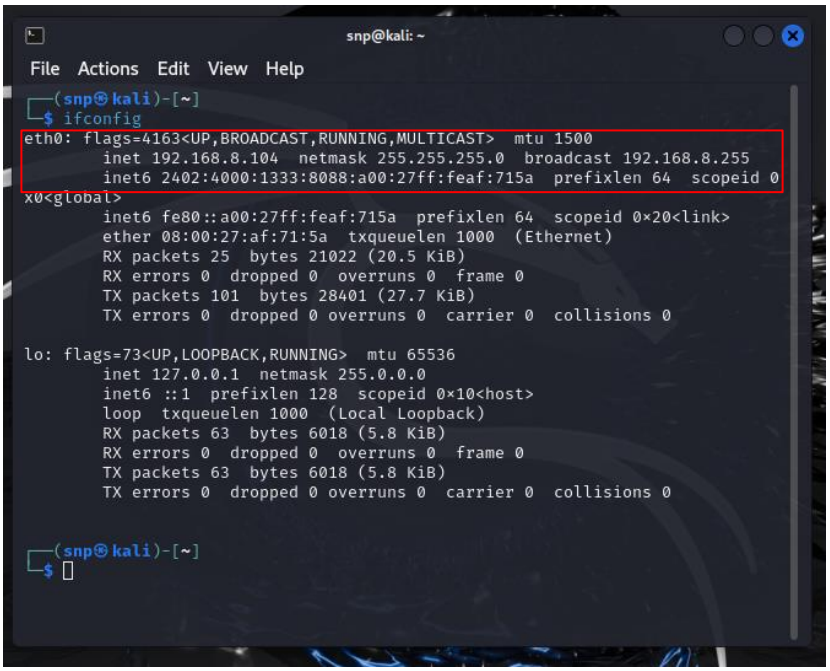
1. Select an IP address within the subnet of your network that is not already in use by other devices.

   192.168.8.104 /24

2. Edit network configuration



3. Restart networking service
4. Verify configuration



References

[1] 07 A. 2017 D. B. 500up 5 comments, “Build your own DNS server on Linux,” *Opensource.com*. https://opensource.com/article/17/4/build-your-own-name-server

[2] A. Shibani, “Baeldung,” *Baeldung on Linux*, Jan. 24, 2024. https://www.baeldung.com/linux/install-configure-dhcp-server.