Lindsay Graham

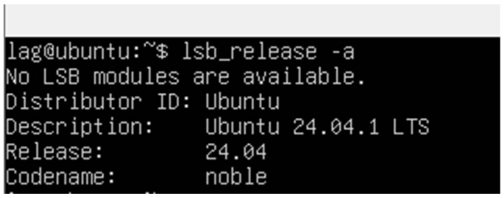
Networking

The first part of this document will outline some of the networking capabilities for the Ubuntu and CentOS servers. It will detail popular commands and provide examples of the output. The second section will include a basic networking script that can be used with each system, and the reasoning for the various commands within the script.

Ubuntu Server

To identify what version of Ubuntu is installed, use the **lsb_release -a** command (Kinsta, 2024). The version is listed on the Description line.

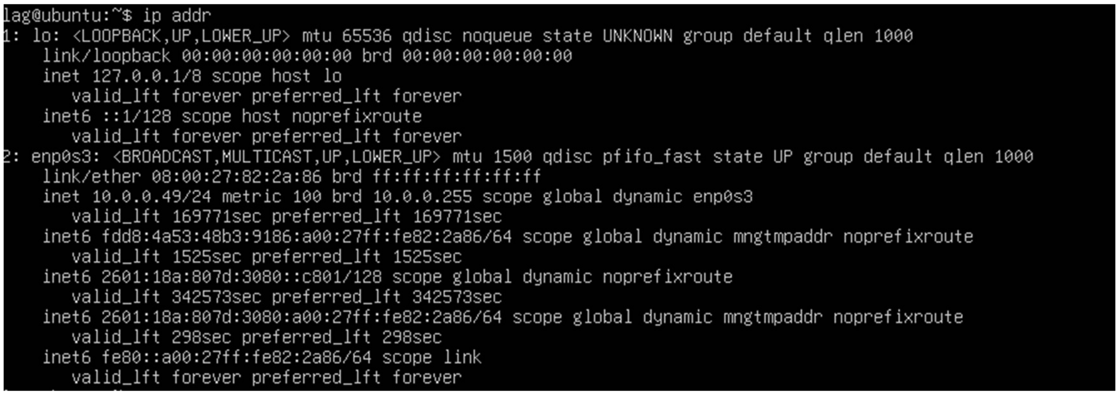
Figure 1.1 Results of **lsb_release -a**



```
lag@ubuntu:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.1 LTS
Release:        24.04
Codename:       noble
```

The **ip addr**, or **ip a**, command displays information about the network interfaces (van Vugt, 2016). The first section indicated by “lo” represents the loopback interface. This interface is used for internal communication within the system. For this server, the IPv4 address is 127.0.0.1. The second section indicated by “enp0s3” represents the broadcast and multicast interface (McKay, 2020). According to McKay, “the “en” stands for ethernet or the wireless adapter, “p0” is the bus number of the ethernet card, and “s3” is the slot number.” The inet, or IPv4 address for this system is 10.0.0.49.

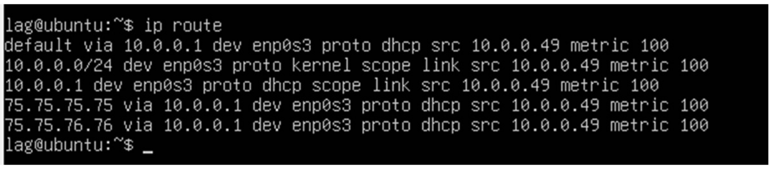
Figure 1.2 Results of **ip addr**



```
lag@ubuntu:~$ ip addr
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:82:2a:86 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.49/24 metric 100 brd 10.0.0.255 scope global dynamic enp0s3
       valid_lft 169771sec preferred_lft 169771sec
    inet6 fdd8:4a53:48b3:9186:a00:27ff:fe82:2a86/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 1525sec preferred_lft 1525sec
    inet6 2601:18a:807d:3080::c801/128 scope global dynamic noprefixroute
       valid_lft 342573sec preferred_lft 342573sec
    inet6 2601:18a:807d:3080:a00:27ff:fe82:2a86/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 298sec preferred_lft 298sec
    inet6 fe80::a00:27ff:fe82:2a86/64 scope link
       valid_lft forever preferred_lft forever
```

The **ip route** command shows the interfaces that data packets are sent through. To view the routes that are defined in the Ubuntu server, use the command **ip route**. The first line specifies the default route. In the default route, the “dev enp0s3” specifies that the 10.0.0.1 interface should be used to send packets to the system’s router. The “proto dhcp” signifies that the routing protocol indicated that routes will be selected dynamically. The “metric 100” refers to the preference of one route over another.

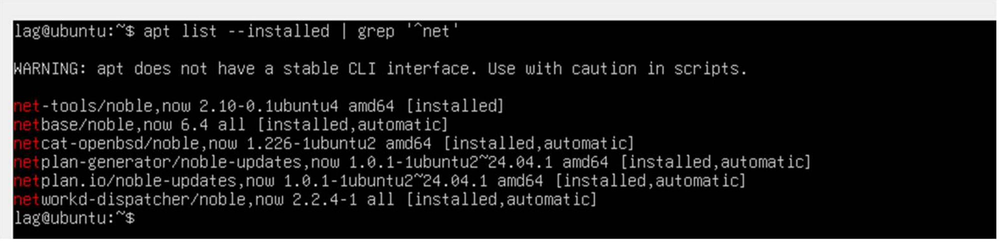
Figure 1.3 Results of **ip route**



```
lag@ubuntu:~$ ip route
default via 10.0.0.1 dev enp0s3 proto dhcp src 10.0.0.49 metric 100
10.0.0.0/24 dev enp0s3 proto kernel scope link src 10.0.0.49 metric 100
10.0.0.1 dev enp0s3 proto dhcp scope link src 10.0.0.49 metric 100
75.75.75.75 via 10.0.0.1 dev enp0s3 proto dhcp src 10.0.0.49 metric 100
75.75.76.76 via 10.0.0.1 dev enp0s3 proto dhcp src 10.0.0.49 metric 100
lag@ubuntu:~$ _
```

The command to list all installed packages on the Ubuntu server is **apt list –installed** (Gite, 2024). To specifically search for network related packages use the command with grep to search for all packages that start with “net.” The **apt list -–installed** command, used with grep, can search for all network related packages. Below is a list of network related packages that were found with the command **apt list –installed | grep ‘^net.’**

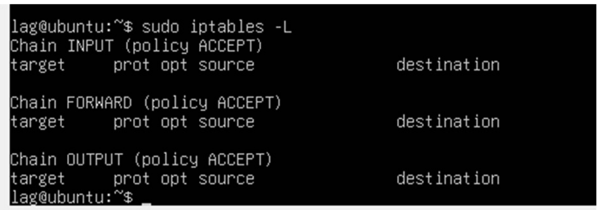
Figure 1.4 Results of **apt list –installed | grep ‘^net’**



```
lag@ubuntu:~$ apt list --installed | grep '^net'

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

net-tools/noble,now 2.10-0.1ubuntu4 amd64 [installed]
netbase/noble,now 6.4 all [installed,automatic]
netcat-openbsd/noble,now 1.226-1ubuntu2 amd64 [installed,automatic]
netplan-generator/noble-updates,now 1.0.1-1ubuntu2~24.04.1 amd64 [installed,automatic]
netplan.io/noble-updates,now 1.0.1-1ubuntu2~24.04.1 amd64 [installed,automatic]
networkd-dispatcher/noble,now 2.2.4-1 all [installed,automatic]
lag@ubuntu:~$
```

This Ubuntu server uses iptables for the firewall. To view the current iptables rules use the command **sudo iptables -L** (Ubuntu, 2020). Figure 1.5 shows that there are no rules in place.

Figure 1.5 Results of **sudo iptables -L**

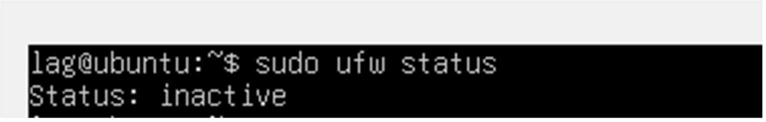


```
lag@ubuntu:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
lag@ubuntu:~$ _
```

The Uncomplicated Firewall (UFW) is an interface that can simply the firewall configuration on the Ubuntu server (Boucheron et al., 2024). The UFW is disabled by default (Ubuntu , 2024).

To check the UFWs status use the command **sudo ufw status**.

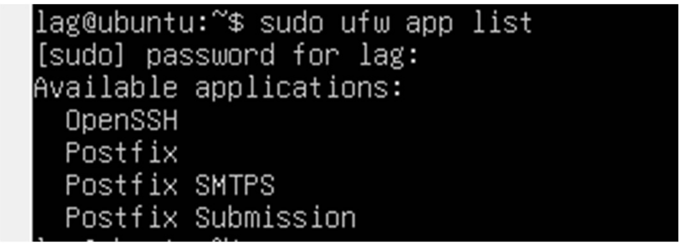
Figure 1.6 Results of **sudo ufw status**



```
lag@ubuntu:~$ sudo ufw status
Status: inactive
```

By default, the UFW's rules deny incoming traffic and allow outgoing connections. To view which profiles are registered with the UFW, use the command **ufw app list**. The OpenSSH was installed with the server setup, but the Postfix programs were installed with emacs.

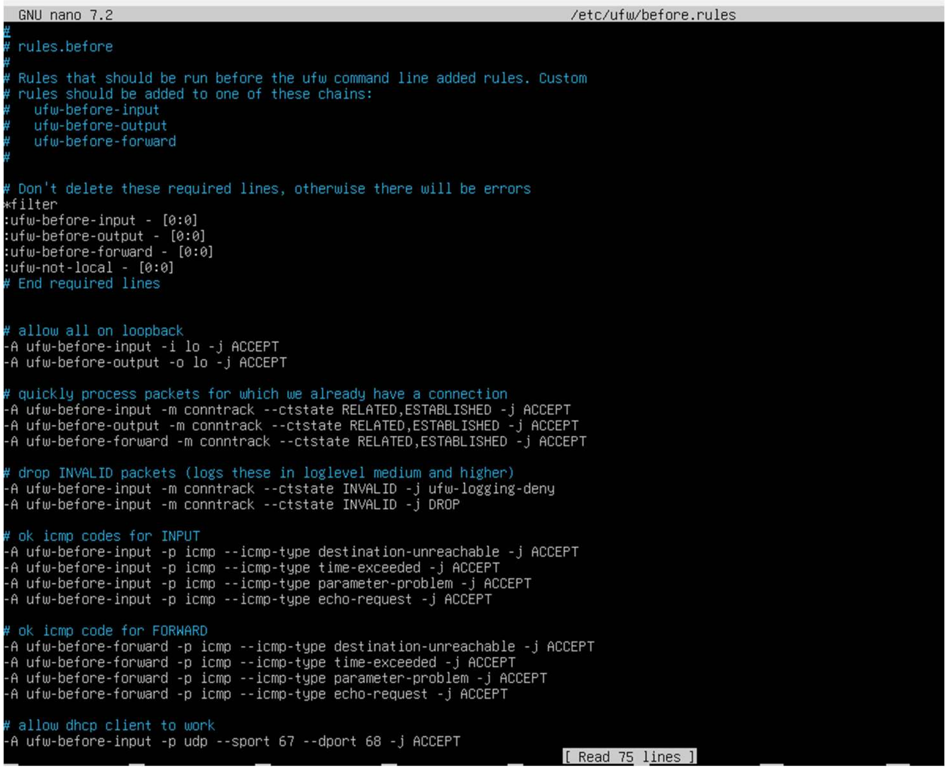
Figure 1.7 Results of **sudo ufw app list**



```
lag@ubuntu:~$ sudo ufw app list
[sudo] password for lag:
Available applications:
  OpenSSH
  Postfix
  Postfix SMTPS
  Postfix Submission
```

To view the rules that should be run before the command line added rules view the rules.before file in the etc/ufw/ directory.

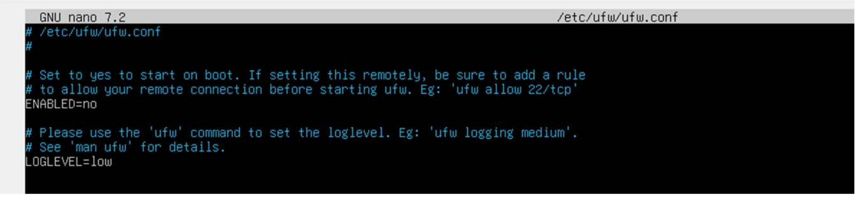
Figure 1.8 Image of /etc/ufw/before.rules



```
  GNU nano 7.2                                               /etc/ufw/before.rules
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#

# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines


# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT

# quickly process packets for which we already have a connection
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

# drop INVALID packets (logs these in loglevel medium and higher)
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP

# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

# allow dhcp client to work
-A ufw-before-input -p udp --sport 67 --dport 68 -j ACCEPT
                                                                  [ Read 75 lines ]
```

To view custom firewall rules use the command **sudo nano /etc/ufw/ufw.conf**.

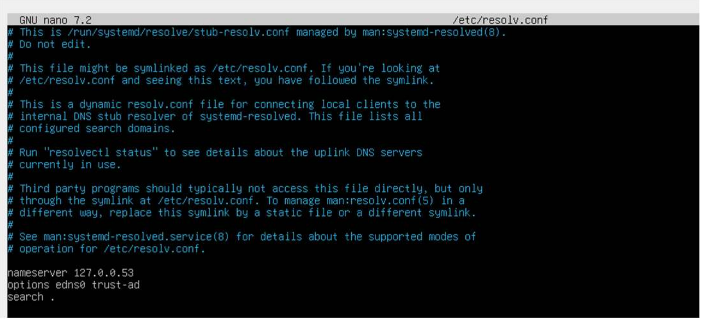
Figure 1.9 Image of /etc/ufw/ufw.conf



```
  GNU nano 7.2                                    /etc/ufw/ufw.conf
# /etc/ufw/ufw.conf
#

# Set to yes to start on boot. If setting this remotely, be sure to add a rule
# to allow your remote connection before starting ufw. Eg: 'ufw allow 22/tcp'
ENABLED=no

# Please use the 'ufw' command to set the loglevel. Eg: 'ufw logging medium'.
# See 'man ufw' for details.
LOGLEVEL=low
```

The resolv.conf file outlines how the system will resolve alpha-numeric names into numerical network addresses (Debian, 2024). This file will hold the IP addresses of the domain name system (DVS) resolvers, or nameservers, that can do this translation. Below, the IP address 127.0.0.53 is the address used for this purpose.

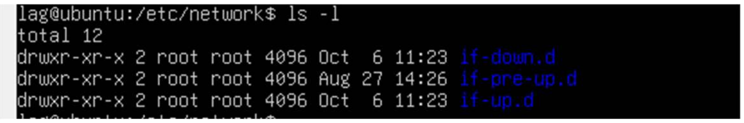
Figure 1.10 Image of /etc/resolv.conf



```
  GNU nano 7.2                                    /etc/resolv.conf
# This is /run/systemd/resolve/stub-resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search .
```

This server does not have the /etc/network/interfaces configuration file. This is the content of the /etc/network/ directory.

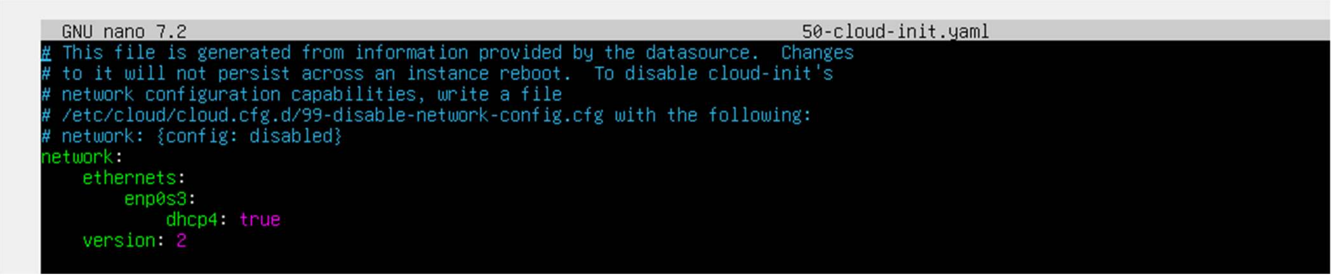
Figure 1.11 Contents of /etc/networks directory



```
lag@ubuntu:/etc/network$ ls -l
total 12
drwxr-xr-x 2 root root 4096 Oct  6 11:23 if-down.d
drwxr-xr-x 2 root root 4096 Aug 27 14:26 if-pre-up.d
drwxr-xr-x 2 root root 4096 Oct  6 11:23 if-up.d
```

Modern versions of Ubuntu use netplan (Ubuntu Core, 2024). In the /etc/netplan directory, there was a file 50-cloud-init.yaml. This shows the default network configuration.

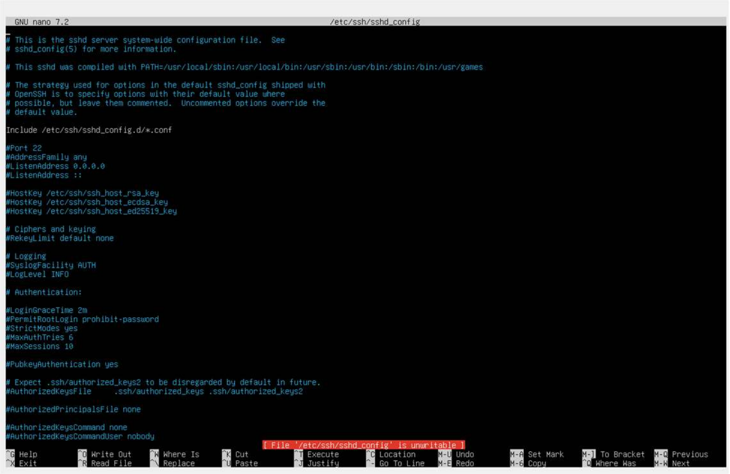
Figure 1.12 Contents of 50-cloud-init.yaml



```
  GNU nano 7.2                                    50-cloud-init.yaml
# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        enp0s3:
            dhcp4: true
    version: 2
```

To view the ssh server configuration, open the sshd_config file. This file does not have any rules, only comments (SSH Academy, 2024).

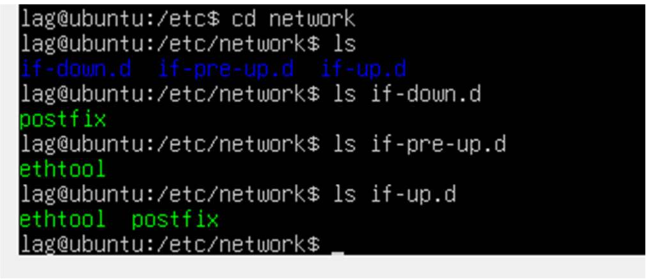
Figure 1.13 results of /etc/ssh/sshd_config



```
  GNU nano 7.2                                    /etc/ssh/sshd_config

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

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

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
                              [ File '/etc/ssh/sshd_config' is unwritable ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut     ^T Execute  ^C Location    M-U Undo  M-A Set Mark  M-] To Bracket  M-Q Previous
^X Exit      ^R Read File  ^\ Replace   ^U Paste   ^J Justify  ^- Go To Line  M-E Redo  M-6 Copy      ^Q Where Was    M-W Next
```

To explore what types of networking files exist, you can **ls** different network related directories. Here is information about what is in the /etc/network directory.

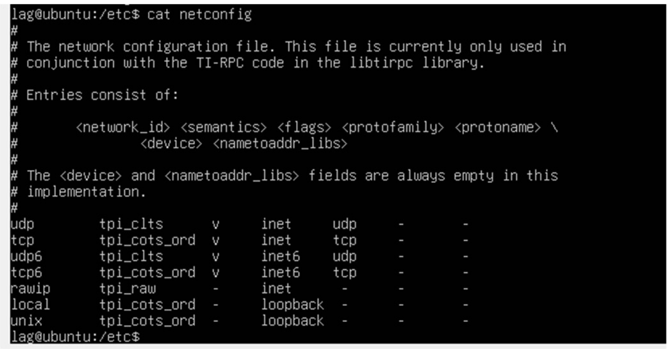
Figure 1.14 Contents of /etc/network directory



```
lag@ubuntu:/etc$ cd network
lag@ubuntu:/etc/network$ ls
if-down.d  if-pre-up.d  if-up.d
lag@ubuntu:/etc/network$ ls if-down.d
postfix
lag@ubuntu:/etc/network$ ls if-pre-up.d
ethtool
lag@ubuntu:/etc/network$ ls if-up.d
ethtool  postfix
lag@ubuntu:/etc/network$ _
```

If you find different files, you can use cat to see their contents. Here is the netconfig file.

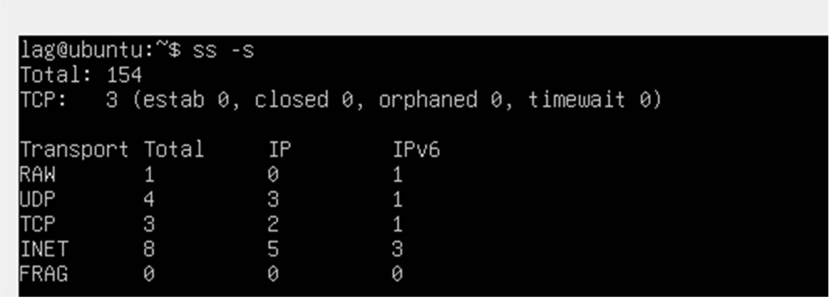
Figure 1.15 The contents of netconfig



```
lag@ubuntu:/etc$ cat netconfig
#
# The network configuration file. This file is currently only used in
# conjunction with the TI-RPC code in the libtirpc library.
#
# Entries consist of:
#
#       <network_id> <semantics> <flags> <protofamily> <protoname> \
#               <device> <nametoaddr_libs>
#
# The <device> and <nametoaddr_libs> fields are always empty in this
# implementation.
#
udp        tpi_clts      v     inet     udp     -       -
tcp        tpi_cots_ord  v     inet     tcp     -       -
udp6       tpi_clts      v     inet6    udp     -       -
tcp6       tpi_cots_ord  v     inet6    tcp     -       -
rawip      tpi_raw       -     inet      -      -       -
local      tpi_cots_ord  -     loopback  -      -       -
unix       tpi_cots_ord  -     loopback  -      -       -
lag@ubuntu:/etc$
```

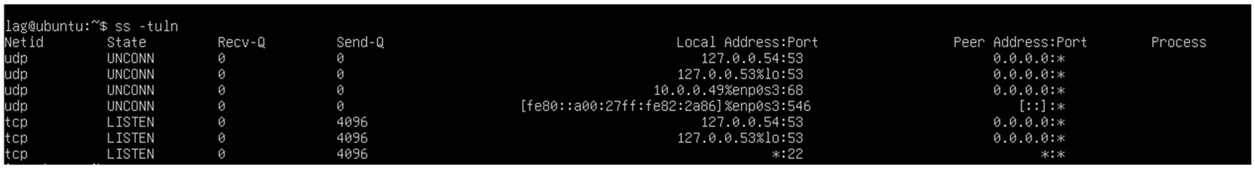
To see summarized information about system sockets use the **ss** command with the summary option (**-s**). The **-tuln** option shows all the listening TCP and UDP ports.

Figure 1.16 The results of the **ss -s** command



```
lag@ubuntu:~$ ss -s
Total: 154
TCP:   3 (estab 0, closed 0, orphaned 0, timewait 0)

Transport Total     IP        IPv6
RAW       1         0         1
UDP       4         3         1
TCP       3         2         1
INET      8         5         3
FRAG      0         0         0
```

Figure 1.17 The results of **ss -tuln**.

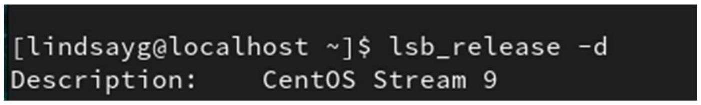


```
lag@ubuntu:~$ ss -tuln
Netid       State        Recv-Q        Send-Q                          Local Address:Port              Peer Address:Port        Process
udp         UNCONN       0             0                                  127.0.0.54:53                     0.0.0.0:*
udp         UNCONN       0             0                               127.0.0.53%lo:53                     0.0.0.0:*
udp         UNCONN       0             0                            10.0.0.49%enp0s3:68                     0.0.0.0:*
udp         UNCONN       0             0               [fe80::a00:27ff:fe82:2a86]%enp0s3:546                  [::]:*
tcp         LISTEN       0             4096                               127.0.0.54:53                     0.0.0.0:*
tcp         LISTEN       0             4096                            127.0.0.53%lo:53                     0.0.0.0:*
tcp         LISTEN       0             4096                                     *:22                          *:*
```

CentOS Server

To identify what version of CentOS is installed, use the **lsb_release -d** command (Rendek, 2015). The version is listed on the Description line.

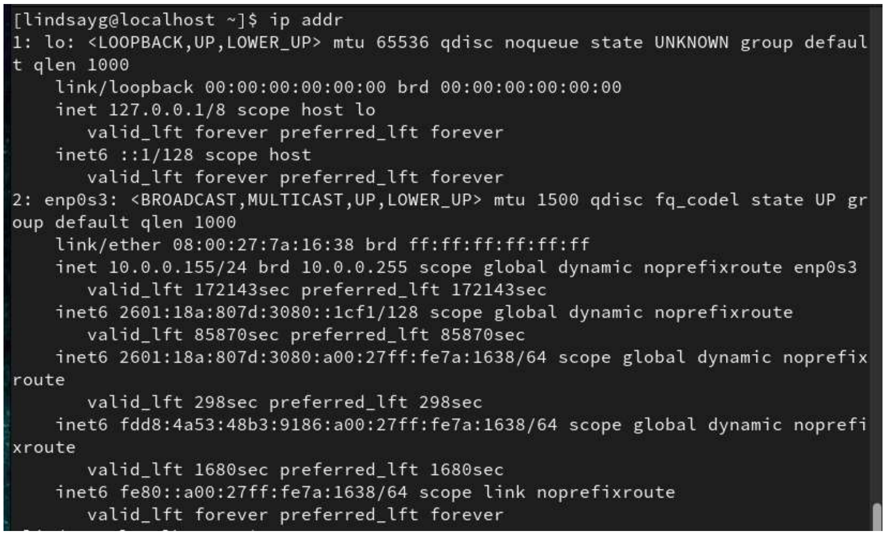
Figure 2.1 Results of **lsb_release -d**



```
[lindsayg@localhost ~]$ lsb_release -d
Description:    CentOS Stream 9
```

The **ip addr**, or **ip a**, commands display information about the network interfaces (van Vugt, 2016). The first section indicated by "lo" represents the loopback interface. This interface is used for internal communication within the system. For this server, the IPv4 loopback address is 127.0.0.1. The second section indicated by "enp0s3" represents the broadcast and multicast interface (McKay, 2020). According to McKay, "the "en" stands for ethernet or the wireless adapter, "p0" is the bus number of the ethernet card, and "s3" is the slot number." The inet, or IPv4 address for this system is 10.0.0.155.

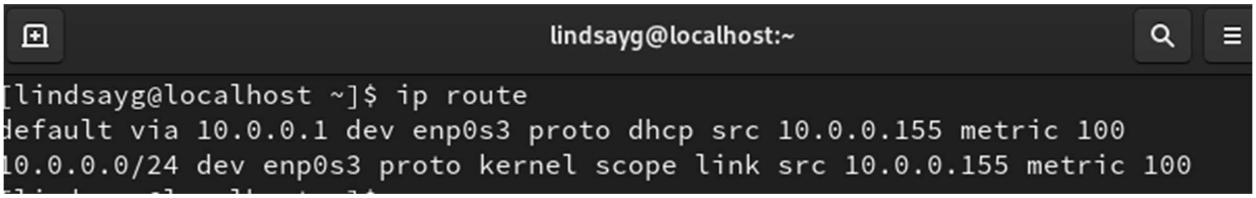
Figure 2.2 Results of **ip addr**



```
[lindsayg@localhost ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:7a:16:38 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.155/24 brd 10.0.0.255 scope global dynamic noprefixroute enp0s3
       valid_lft 172143sec preferred_lft 172143sec
    inet6 2601:18a:807d:3080::1cf1/128 scope global dynamic noprefixroute
       valid_lft 85870sec preferred_lft 85870sec
    inet6 2601:18a:807d:3080:a00:27ff:fe7a:1638/64 scope global dynamic noprefix
route
       valid_lft 298sec preferred_lft 298sec
    inet6 fdd8:4a53:48b3:9186:a00:27ff:fe7a:1638/64 scope global dynamic noprefi
xroute
       valid_lft 1680sec preferred_lft 1680sec
    inet6 fe80::a00:27ff:fe7a:1638/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

The **ip route** command shows the interfaces that data packets are sent through along with which interfaces they should use. To view the routes that are defined in the CentOS server, use the command **ip route**. The first line specifies the default route. In the default route, the "dev enp0s3" specifies that the 10.0.0.1 interface should be used to send packets to the system's router. The "proto dhcp" signifies that the routing protocol indicated that routes will be selected dynamically. The "metric 100" refers to the preference of one route over another.

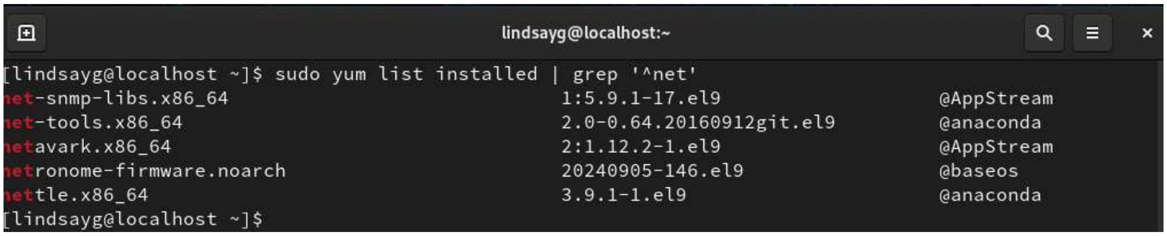
Figure 2.3 Results of **ip route**



```
[lindsayg@localhost ~]$ ip route
default via 10.0.0.1 dev enp0s3 proto dhcp src 10.0.0.155 metric 100
10.0.0.0/24 dev enp0s3 proto kernel scope link src 10.0.0.155 metric 100
```

The command to list all installed packages on the CentOS server is **sudo yum list installed** (Jevtic, 2019). To specifically search for network related packages use the command with grep to search for all packages that start with "net." The **yum list installed** command, used with grep, can search for all network related packages. Below is a list of network related packages that were found with the command **sudo yum list installed | grep '^net.'**

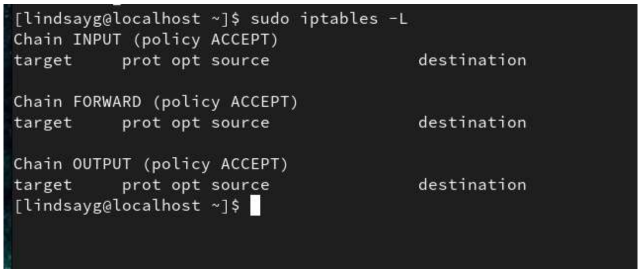
Figure 2.4 Results of **sudo yum list installed | grep '^net'**



```
[lindsayg@localhost ~]$ sudo yum list installed | grep '^net'
net-snmp-libs.x86_64                    1:5.9.1-17.el9                   @AppStream
net-tools.x86_64                        2.0-0.64.20160912git.el9         @anaconda
netavark.x86_64                         2:1.12.2-1.el9                   @AppStream
netronome-firmware.noarch               20240905-146.el9                 @baseos
nettle.x86_64                           3.9.1-1.el9                      @anaconda
[lindsayg@localhost ~]$
```

To view the current iptables rules use the command **sudo iptables -L** (Ubuntu, 2020). Figure 1.5 shows that there are no rules in place.

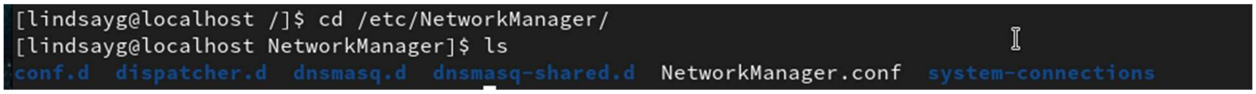
Figure 2.5 Results of **sudo iptables -L**



```
[lindsayg@localhost ~]$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[lindsayg@localhost ~]$
```

To view the network configuration settings navigate to etc/NetworkManager/ directory (Horn, 2022).

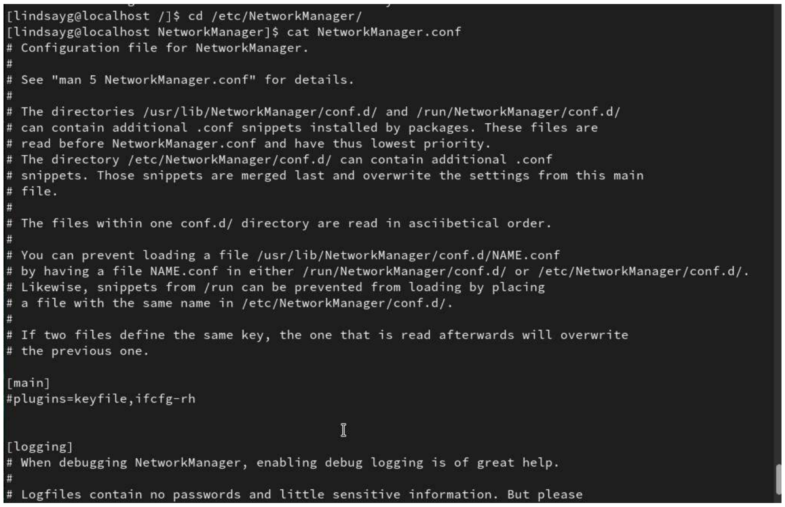
Figure 2.6 Content of NetworkManager directory



```
[lindsayg@localhost /]$ cd /etc/NetworkManager/
[lindsayg@localhost NetworkManager]$ ls
conf.d  dispatcher.d  dnsmasq.d  dnsmasq-shared.d  NetworkManager.conf  system-connections
```

The NetworkingManager.conf file refers to the main configuration file of CentOS. It looks at keyfiles first then ifcfg-rh files next. These files contain the configuration rules.

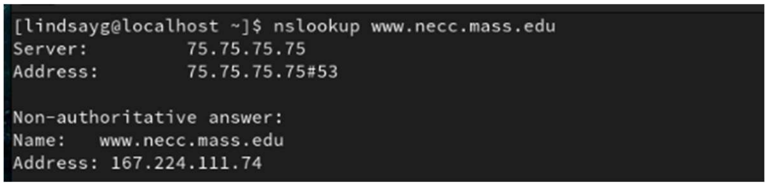
Figure 2.7 Sample of NetworkManager.conf



```
[lindsayg@localhost /]$ cd /etc/NetworkManager/
[lindsayg@localhost NetworkManager]$ cat NetworkManager.conf
# Configuration file for NetworkManager.
#
# See "man 5 NetworkManager.conf" for details.
#
# The directories /usr/lib/NetworkManager/conf.d/ and /run/NetworkManager/conf.d/
# can contain additional .conf snippets installed by packages. These files are
# read before NetworkManager.conf and have thus lowest priority.
# The directory /etc/NetworkManager/conf.d/ can contain additional .conf
# snippets. Those snippets are merged last and overwrite the settings from this main
# file.
#
# The files within one conf.d/ directory are read in asciibetical order.
#
# You can prevent loading a file /usr/lib/NetworkManager/conf.d/NAME.conf
# by having a file NAME.conf in either /run/NetworkManager/conf.d/ or /etc/NetworkManager/conf.d/.
# Likewise, snippets from /run can be prevented from loading by placing
# a file with the same name in /etc/NetworkManager/conf.d/.
#
# If two files define the same key, the one that is read afterwards will overwrite
# the previous one.

[main]
#plugins=keyfile,ifcfg-rh


[logging]
# When debugging NetworkManager, enabling debug logging is of great help.
#
# Logfiles contain no passwords and little sensitive information. But please
```

The command **nslookup** with a web address can show the DNS settings for a particular address (Garn, 2022).

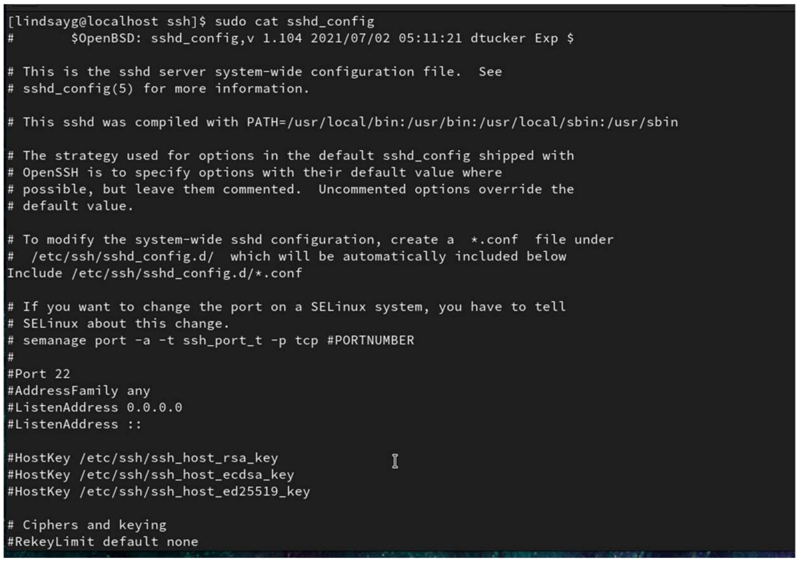
Figure 2.8 Results of **nslookup www.necc.mass.edu**



```
[lindsayg@localhost ~]$ nslookup www.necc.mass.edu
Server:         75.75.75.75
Address:        75.75.75.75#53

Non-authoritative answer:
Name:   www.necc.mass.edu
Address: 167.224.111.74
```

To view the ssh server configuration, open the sshd_config file. This file does not have any rules, only comments (SSH Academy, 2024).

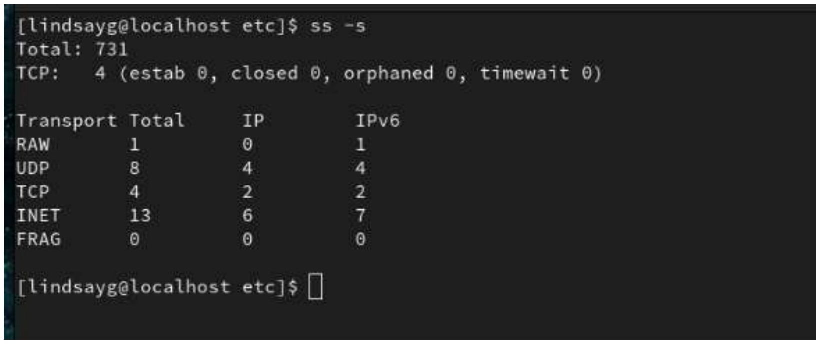
Figure 2.9 results of /etc/ssh/sshd_config



```
[lindsayg@localhost ssh]$ sudo cat sshd_config
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

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

# Ciphers and keying
#RekeyLimit default none
```

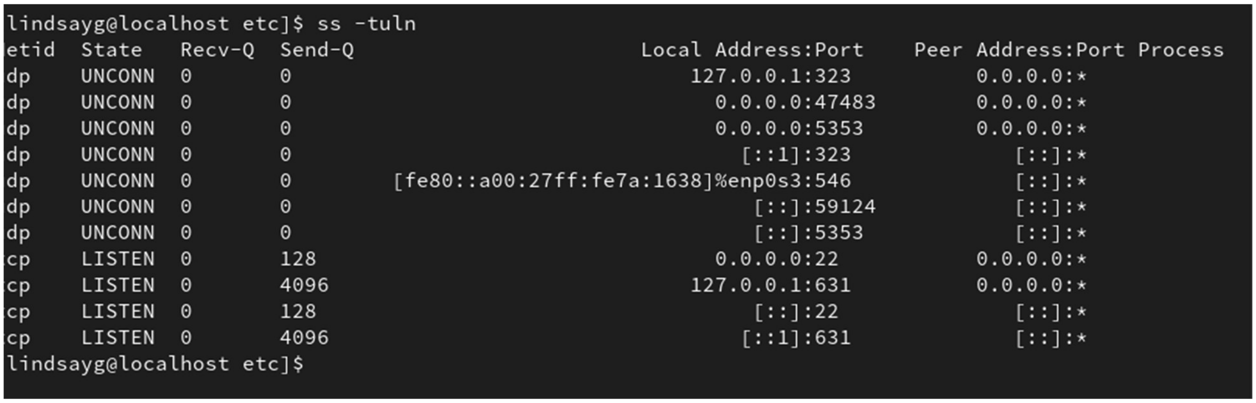
To see summarized information about system sockets use the **ss** command with the summary option (**-s**). The **-tuln** option shows the listening TCP and UDP ports.

Figure 2.10 The results of the **ss -s** command



```
[lindsayg@localhost etc]$ ss -s
Total: 731
TCP:   4 (estab 0, closed 0, orphaned 0, timewait 0)

Transport Total     IP        IPv6
RAW       1         0         1
UDP       8         4         4
TCP       4         2         2
INET      13        6         7
FRAG      0         0         0

[lindsayg@localhost etc]$ 
```

Figure 2.11 The results of **ss -tuln**



```
lindsayg@localhost etc]$ ss -tuln
etid  State   Recv-Q  Send-Q                      Local Address:Port    Peer Address:Port Process
dp    UNCONN  0       0                               127.0.0.1:323          0.0.0.0:*
dp    UNCONN  0       0                                 0.0.0.0:47483        0.0.0.0:*
dp    UNCONN  0       0                                 0.0.0.0:5353         0.0.0.0:*
dp    UNCONN  0       0                                   [::1]:323             [::]:*
dp    UNCONN  0       0       [fe80::a00:27ff:fe7a:1638]%enp0s3:546             [::]:*
dp    UNCONN  0       0                                    [::]:59124           [::]:*
dp    UNCONN  0       0                                    [::]:5353            [::]:*
cp    LISTEN  0       128                               0.0.0.0:22           0.0.0.0:*
cp    LISTEN  0       4096                            127.0.0.1:631          0.0.0.0:*
cp    LISTEN  0       128                                  [::]:22              [::]:*
cp    LISTEN  0       4096                                [::1]:631             [::]:*
lindsayg@localhost etc]$
```

Network Information Script

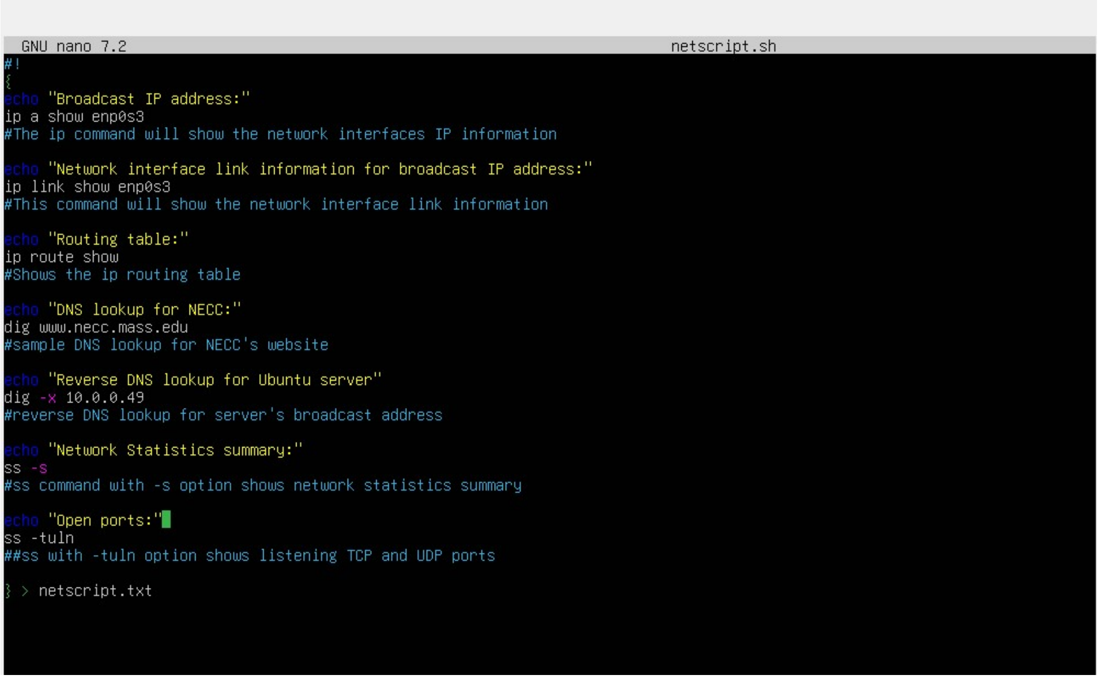
The simple networking script in Figure 3.1 illustrates commonly used network commands. The first two commands, **ip a show enp0s3** and **ip link show enp0s3**, focus on the enp0s3 address and its associated network interface link. The enp0s3 interface is a physical interface, often connected to an ethernet cable or wireless adapter (McKay, 2020). It is the address used to communicate with other network devices and the internet. When using **scp** to transfer files from my laptop to the server, this is the address that I use. If a connectivity issue were to occur, the enp0s3 interface would be helpful for troubleshooting as it is the primary gateway for external communication.

The next command, **ip route show**, provides information about the routing table, which defines where network traffic is forwarded. The results of this command list the routes that are configured on the server. Traffic is routed based on defined rules. If no rule applies, the default route is used. For example, if a new network interface card is installed on a system, a new route may need to be added to enable its use. By reviewing the current routes, administrators can determine if any new routes need to be added, or others removed.

Next, the script retrieves information about the domain name system (DNS) records with the **dig** command. The **dig** command is used to gather details about domain names and their associated records (Dancuk, 2023). In this instance, the script uses **dig www.necc.mass.edu** to query the NECC web address. It also performs a reverse lookup on the server's IP address with the command **dig -x 10.0.0.49**. Network administrators commonly use the **dig** command for troubleshooting DNS issues or verifying DNS configurations.

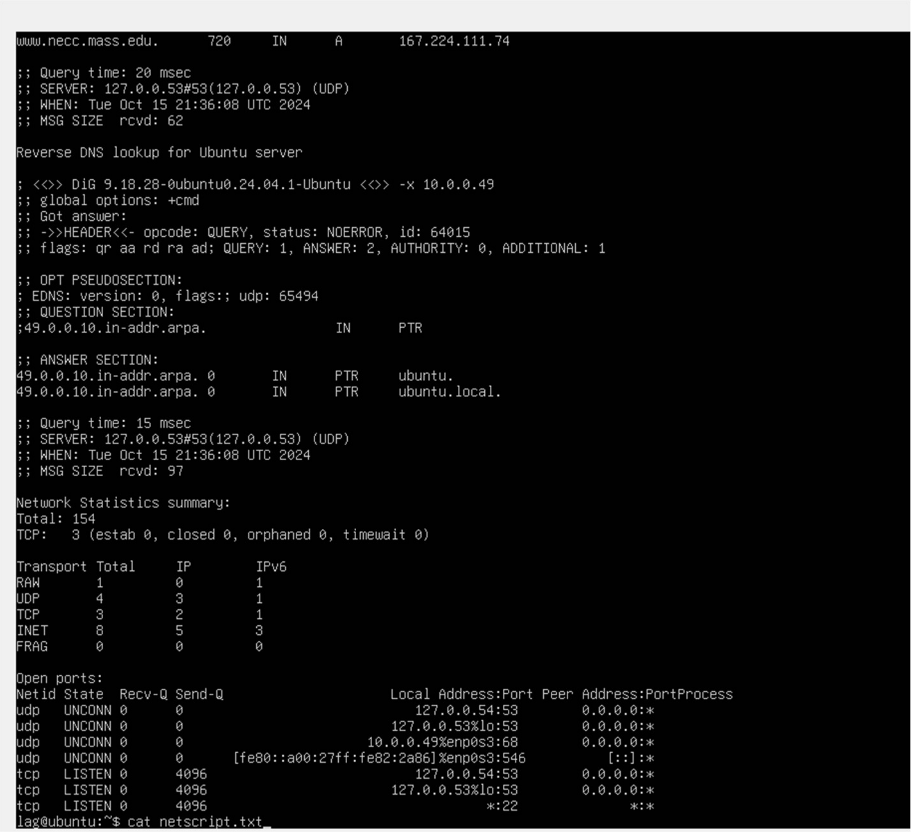
Lastly, the **ss** command was used to display network statistics, including information about network sockets. The **-s** option with **ss** provides summary statistics, while **-tuln** lists all listening TCP and UDP ports on the system (Wake, 2024). This can be helpful for identifying unexpected listening ports, which might indicate a security risk. The commands were enclosed in brackets to group them together, and their output was appended to a new file named netscript.txt.

Figure 3.1 Simple networking script



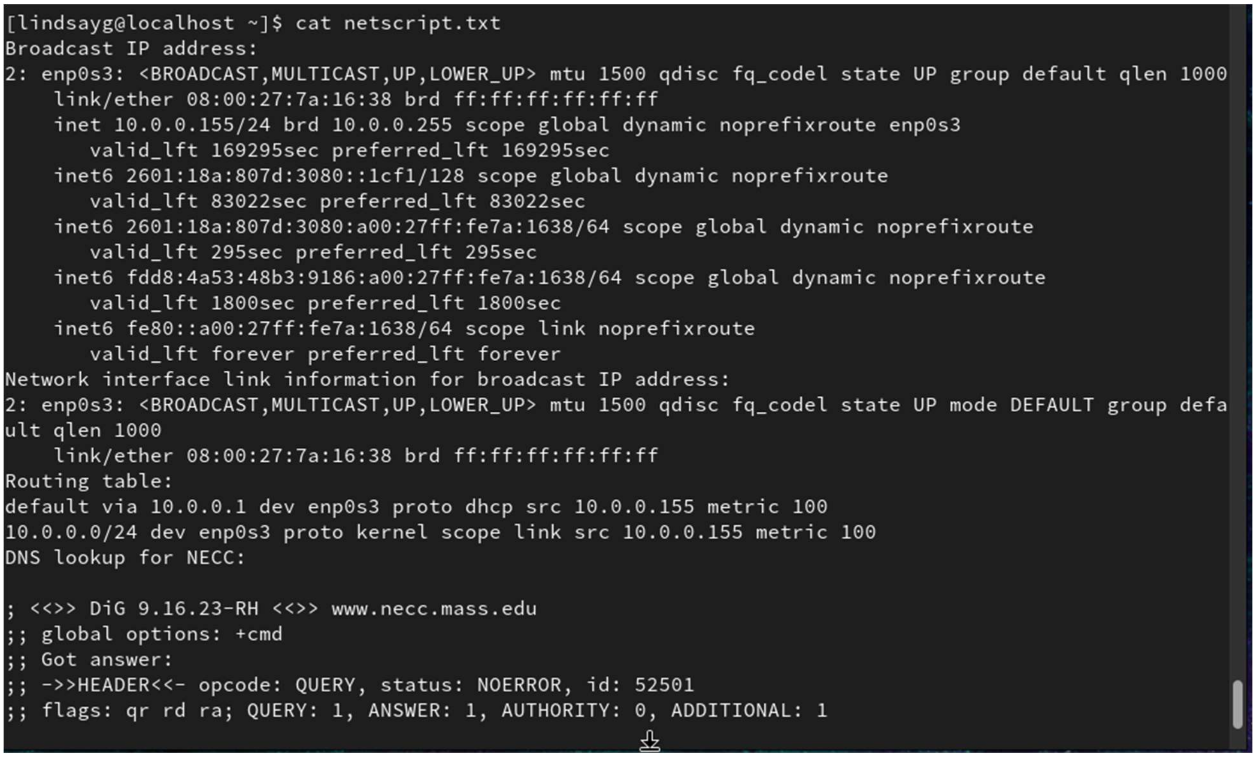
```
  GNU nano 7.2                                                                     netscript.sh
#!
{
echo "Broadcast IP address:"
ip a show enp0s3
#The ip command will show the network interfaces IP information

echo "Network interface link information for broadcast IP address:"
ip link show enp0s3
#This command will show the network interface link information

echo "Routing table:"
ip route show
#Shows the ip routing table

echo "DNS lookup for NECC:"
dig www.necc.mass.edu
#sample DNS lookup for NECC's website

echo "Reverse DNS lookup for Ubuntu server"
dig -x 10.0.0.49
#reverse DNS lookup for server's broadcast address

echo "Network Statistics summary:"
ss -s
#ss command with -s option shows network statistics summary

echo "Open ports:"
ss -tuln
##ss with -tuln option shows listening TCP and UDP ports

} > netscript.txt
```

Figure 3.2 Sample of results from cat netscript.txt output file on Ubuntu Server



```
www.necc.mass.edu.      720     IN      A       167.224.111.74

;; Query time: 20 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Oct 15 21:36:08 UTC 2024
;; MSG SIZE  rcvd: 62

Reverse DNS lookup for Ubuntu server

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> -x 10.0.0.49
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64015
;; flags: qr aa rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;49.0.0.10.in-addr.arpa.                IN      PTR

;; ANSWER SECTION:
49.0.0.10.in-addr.arpa. 0       IN      PTR     ubuntu.
49.0.0.10.in-addr.arpa. 0       IN      PTR     ubuntu.local.

;; Query time: 15 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Oct 15 21:36:08 UTC 2024
;; MSG SIZE  rcvd: 97

Network Statistics summary:
Total: 154
TCP:   3 (estab 0, closed 0, orphaned 0, timewait 0)

Transport Total     IP        IPv6
RAW       1         0         1
UDP       4         3         1
TCP       3         2         1
INET      8         5         3
FRAG      0         0         0

Open ports:
Netid State  Recv-Q Send-Q                     Local Address:Port Peer Address:PortProcess
udp   UNCONN 0      0                             127.0.0.54:53        0.0.0.0:*
udp   UNCONN 0      0                          127.0.0.53%lo:53        0.0.0.0:*
udp   UNCONN 0      0                        10.0.0.49%enp0s3:68        0.0.0.0:*
udp   UNCONN 0      0       [fe80::a00:27ff:fe82:2a86]%enp0s3:546          [::]:*
tcp   LISTEN 0      4096                          127.0.0.54:53        0.0.0.0:*
tcp   LISTEN 0      4096                       127.0.0.53%lo:53        0.0.0.0:*
tcp   LISTEN 0      4096                                *:22              *:*
lag@ubuntu:~$ cat netscript.txt_
```

Figure 3.3 Sample of results from cat netscript.txt on CentOS Server



```
[lindsayg@localhost ~]$ cat netscript.txt
Broadcast IP address:
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:7a:16:38 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.155/24 brd 10.0.0.255 scope global dynamic noprefixroute enp0s3
       valid_lft 169295sec preferred_lft 169295sec
    inet6 2601:18a:807d:3080::1cf1/128 scope global dynamic noprefixroute
       valid_lft 83022sec preferred_lft 83022sec
    inet6 2601:18a:807d:3080:a00:27ff:fe7a:1638/64 scope global dynamic noprefixroute
       valid_lft 295sec preferred_lft 295sec
    inet6 fdd8:4a53:48b3:9186:a00:27ff:fe7a:1638/64 scope global dynamic noprefixroute
       valid_lft 1800sec preferred_lft 1800sec
    inet6 fe80::a00:27ff:fe7a:1638/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
Network interface link information for broadcast IP address:
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group defa
ult qlen 1000
    link/ether 08:00:27:7a:16:38 brd ff:ff:ff:ff:ff:ff
Routing table:
default via 10.0.0.1 dev enp0s3 proto dhcp src 10.0.0.155 metric 100
10.0.0.0/24 dev enp0s3 proto kernel scope link src 10.0.0.155 metric 100
DNS lookup for NECC:

; <<>> DiG 9.16.23-RH <<>> www.necc.mass.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52501
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
```

References

Boucheron, B., Camisso, J., & Abid, E. (2024, February 27). *How to Set Up a Firewall with UFW on Ubuntu | DigitalOcean*. Www.digitalocean.com. https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu

Dancuk, M. (2023, September 13). *Linux Network Commands Cheat Sheet | phoenixNAP KB*. Knowledge Base by PhoenixNAP. https://phoenixnap.com/kb/linux-network-commands#ftoc-heading-7

Debian. (2024, August 1). *NetworkConfiguration - Debian Wiki*. Debian.org. https://wiki.debian.org/NetworkConfiguration#Defining_the_.28DNS.29_Nameservers

Garn, D. (2022, April 4). *Linux troubleshooting commands: 4 tools for DNS name resolution problems*. Enable Sysadmin. https://www.redhat.com/sysadmin/DNS-name-resolution-troubleshooting-tools

Gite, V. (2024, August 16). *How do I see what packages are installed on Ubuntu Linux?* NixCraft. https://www.cyberciti.biz/faq/apt-get-list-packages-are-installed-on-ubuntu-linux/

Horn, C. (2022). *RHEL 9 networking: Say goodbye to ifcfg-files, and hello to keyfiles*. Redhat.com. https://www.redhat.com/en/blog/rhel-9-networking-say-goodbye-ifcfg-files-and-hello-keyfiles

Jevtic, G. (2019, April 29). *3 Options to List All Installed Packages in CentOS {Easiest Way}*. Knowledge Base by PhoenixNAP. https://phoenixnap.com/kb/how-to-list-installed-packages-on-centos

Kinsta. (2024, May 7). *How To Check Your Ubuntu Version (Using the Command Line and GUI)*. Kinsta®. https://kinsta.com/knowledgebase/check-ubuntu-version/

McKay, D. (2020, March 4). *How to Use the ip Command on Linux*. How-to Geek. https://www.howtogeek.com/657911/how-to-use-the-ip-command-on-linux/

Rendek, L. (2015, August 4). *How to check CentOS version*. LinuxConfig. https://linuxconfig.org/how-to-check-centos-version

SSH Academy. (2024). *sshd_config is the OpenSSH server configuration file. How to configure and troubleshoot. Avoid getting accidentally locked out of remote server.* Www.ssh.com. https://www.ssh.com/academy/ssh/sshd_config

Ubuntu. (2020, April 11). *IptablesHowTo - Community Help Wiki*. Ubuntu.com. https://help.ubuntu.com/community/IptablesHowTo

Ubuntu . (2024). *Firewall*. Ubuntu Server. https://documentation.ubuntu.com/server/how-to/security/firewalls/?_ga=2.258631562.14676876.1728760528-583595323.1728760528&_gl=1

Ubuntu Core. (2024). *NetworkManager and netplan*. Ubuntu.com. https://ubuntu.com/core/docs/networkmanager/networkmanager-and-netplan

van Vugt, S. (2016, January 4). *The Linux ip command makes network config easy | TechTarget*. Data Center. https://www.techtarget.com/searchdatacenter/tip/The-Linux-ip-command-makes-network-config-easy

Wake, T. (2024, August 22). *Linux Incident Response - Using ss for Network Analysis | SANS*. Sans.org. https://www.sans.org/blog/linux-incident-response-using-ss-for-network-analysis/