Command Link Network and Link

TA: Siddhartha Vardhan and Rahul Gupta

Topics Covered:

- route
- arp
- nmap
- ssh
- scp

route command

route command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or updating the IP/kernel routing table.

Basically if you want to show / manipulate the IP routing table.

$sudo apt-get install net-tools

What is a Routing Table?

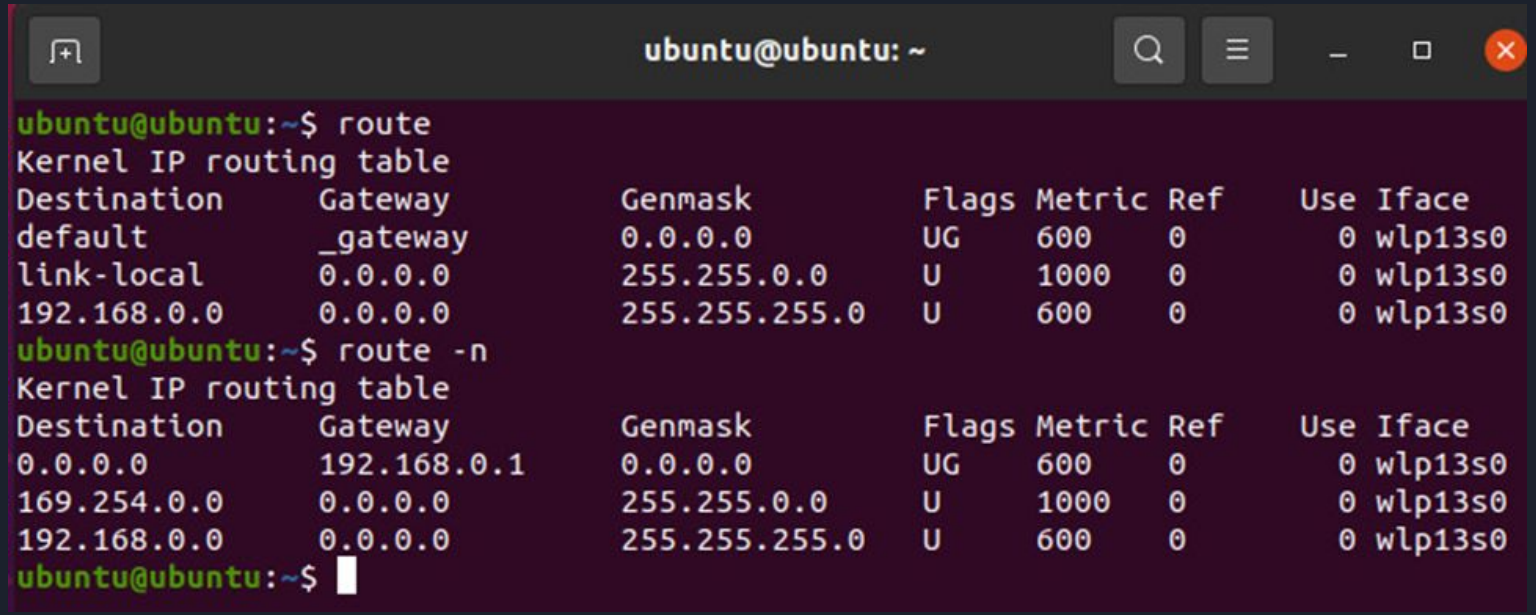
A routing table is a file containing information on how the information or packets should be transferred: the network path to all nodes or devices within a network.

It is a map used by routers and gateways to track paths. The hop-by-hop routing is widely used, the packet contains the routing table to reach the next hop, once reached, it will read the routing table again to reach the next hop.

Hence, by using the route command you can communicate with subnets and different networks, you can also block the traffic between networks or devices by modifying the routing table.

-n : displays routing table in full numeric form.

-v: displays the routing table with verbose information.



We can assign a default gateway using the following command:

```
ubuntu@ubuntu:~$ sudo route add default gw 169.254.0.0
ubuntu@ubuntu:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         169.254.0.0     0.0.0.0         UG    0      0        0 wlp13s0
0.0.0.0         192.168.0.1     0.0.0.0         UG    600    0        0 wlp13s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp13s0
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0 wlp13s0
ubuntu@ubuntu:~$
```

This assigns a gateway address on which all the packets that do not belong to the network are forwarded.

We can reject routing to a particular host or network, using the **reject** command.

```
ubuntu@ubuntu:~$ sudo route add -host 192.168.1.51 reject
ubuntu@ubuntu:~$ ping 192.168.1.51
ping: connect: No route to host
ubuntu@ubuntu:~$
```

Now if you will ping to the above-mentioned IP it will display “Network is unreachable”.

ip route: This command will give us the details of the IP routing table.

```
ubuntu@ubuntu:~$ ip route
default via 192.168.238.39 dev wlp13s0 proto dhcp metric 600
169.254.0.0/16 dev wlp13s0 scope link metric 1000
unreachable 192.168.1.51 scope host
192.168.238.0/24 dev wlp13s0 proto kernel scope link src 192.168.238.36 metric
600
```

To get the details of the local table with destination addresses assigned to the localhost

```
ubuntu@ubuntu:~$ ip route show table local
broadcast 127.0.0.0 dev lo proto kernel scope link src 127.0.0.1
local 127.0.0.0/8 dev lo proto kernel scope host src 127.0.0.1
local 127.0.0.1 dev lo proto kernel scope host src 127.0.0.1
broadcast 127.255.255.255 dev lo proto kernel scope link src 127.0.0.1
broadcast 192.168.238.0 dev wlp13s0 proto kernel scope link src 192.168.238.36
local 192.168.238.36 dev wlp13s0 proto kernel scope host src 192.168.238.36
broadcast 192.168.238.255 dev wlp13s0 proto kernel scope link src 192.168.238.36
ubuntu@ubuntu:~$
```

We can display the entries with ipv4 and ipv6 using the following command:

```
ubuntu@ubuntu:~$ ip -4 route
default via 192.168.238.39 dev wlp13s0 proto dhcp metric 600
169.254.0.0/16 dev wlp13s0 scope link metric 1000
unreachable 192.168.1.51 scope host
192.168.238.0/24 dev wlp13s0 proto kernel scope link src 192.168.238.36 metric 60
0
ubuntu@ubuntu:~$ ip -6 route
::1 dev lo proto kernel metric 256 pref medium
fe80::/64 dev wlp13s0 proto kernel metric 600 pref medium
ubuntu@ubuntu:~$
```

arp command

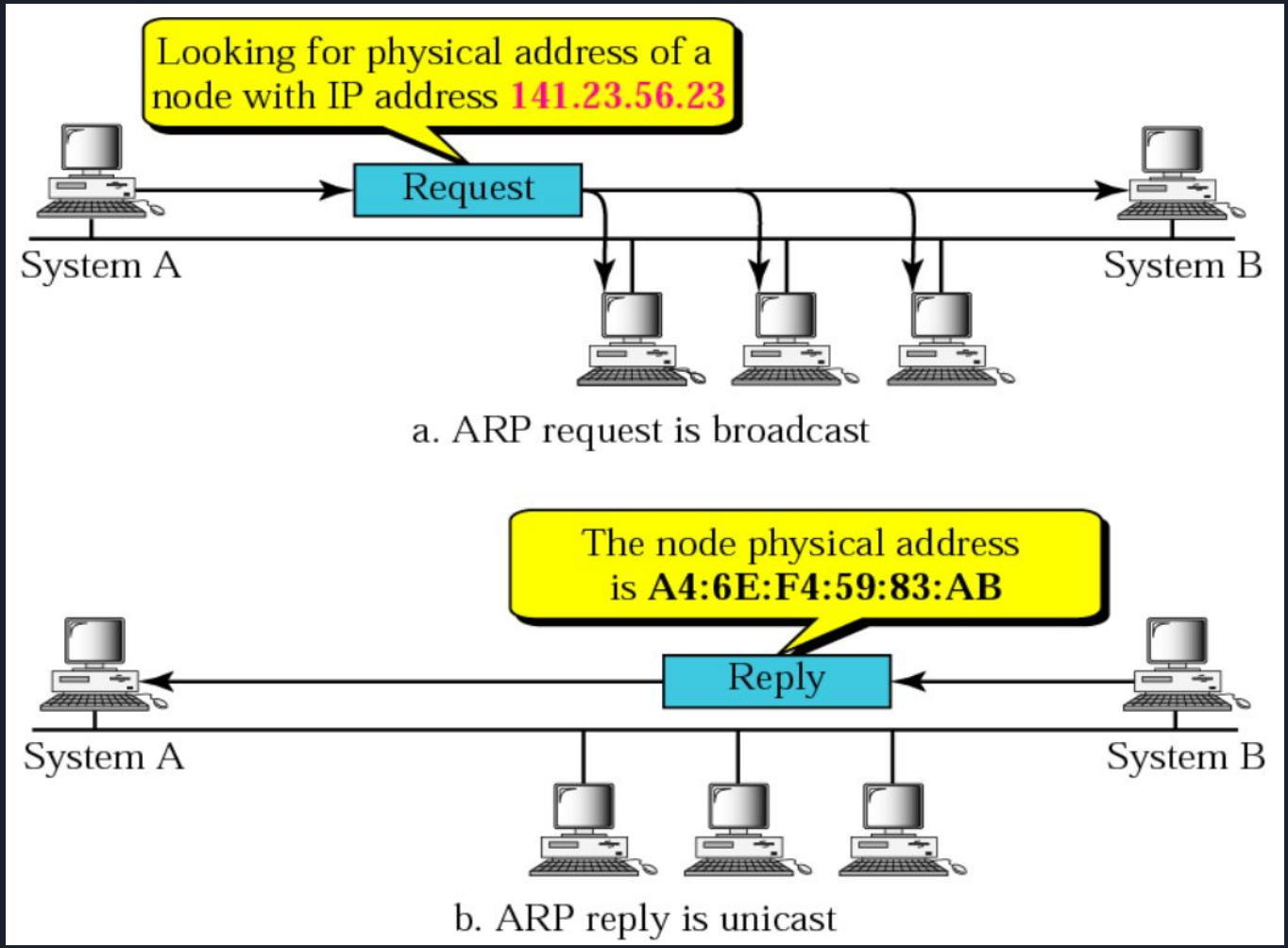
arp command manipulates the System's ARP cache. It also allows a complete dump of the ARP cache.

ARP stands for Address Resolution Protocol. The primary function of this protocol is to resolve the IP address of a system to its mac address, and hence it works between level 2(Data link layer) and level 3(Network layer).

The Sender asks the receiver to announce its physical address when needed. ARP is designed to this purpose. Anytime, a host(router) needs to find the physical address of another host(router) on its network, it sends an ARP Request and the query packet is broadcast over the network.

Host will ignore the ARP request, if the IP address is not the host. Otherwise the host will reply to the sender. Every host(router) on the network receives and processes the ARP query packet. But only the intended recipient recognizes its IP address and sends back an ARP Reply.

Source: https://gsephrioth.github.io/Ch7-Address-Resolution-Protocol(ARP)/


Looking for physical address of a node with IP address 141.23.56.23
Request
System A
System B
a. ARP request is broadcast
The node physical address is A4:6E:F4:59:83:AB
Reply
System A
System B
b. ARP reply is unicast

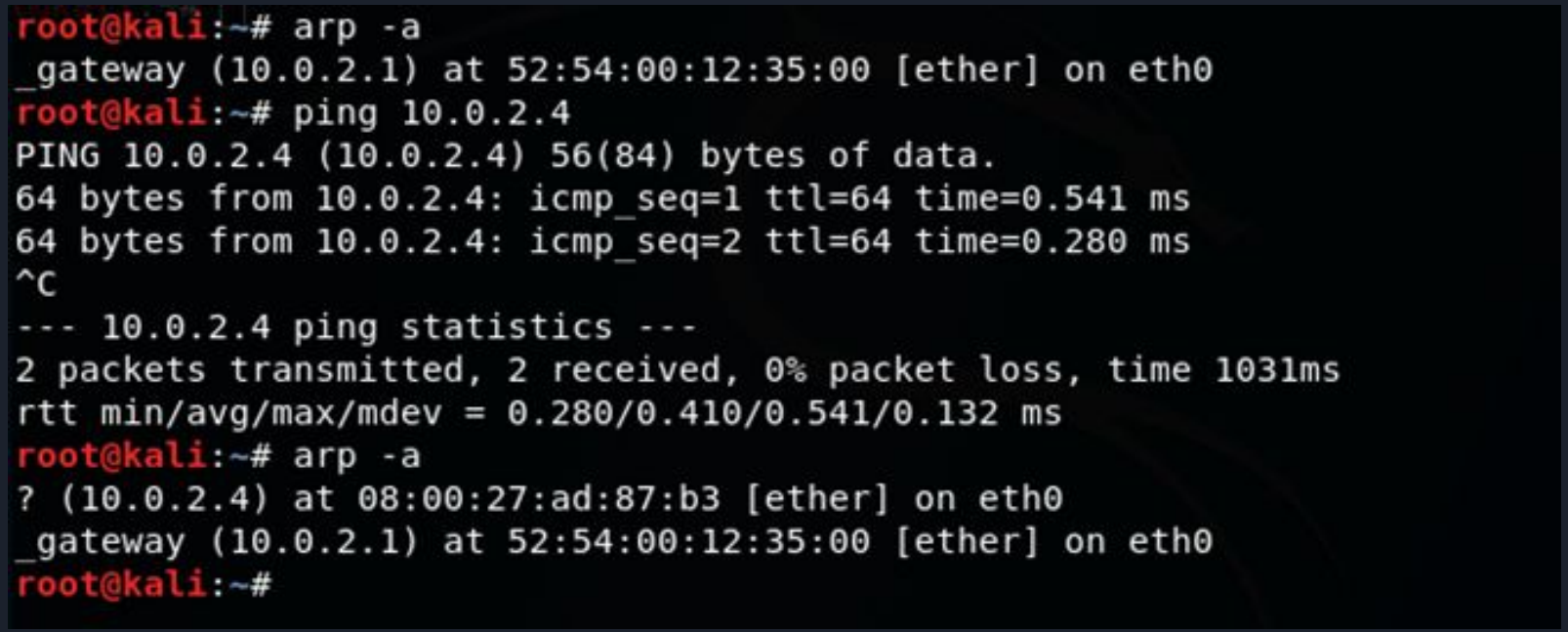
The arp program displays and modifies the Internet-to-MAC address translation tables used by the address resolution protocol.

With no flags, the program displays the current ARP entry for hostname. The host may be specified by name or by number, using Internet dot notation.

Options that modify the ARP translation tables (–d, –f, and –s) can be used only when the invoke command is granted the sys_net_config privilege.

Source: https://docs.oracle.com/cd/E88353_01/html/E72487/arp-8.html

arp cache is basically the memory location where we have recently resolved IP and mac addresses.



- **-v, –verbose:** This option shows the verbose information.
- **-n, –numeric:** This option shows numerical addresses instead of symbolic host, port or usernames

```
root@kali:~# arp -v
Address                  HWtype  HWaddress           Flags Mask            Iface
machine1                 ether   08:00:27:ad:87:b3   C                     eth0
machine2                 ether   08:00:27:27:d6:c7   C                     eth0
10.0.2.3                 ether   08:00:27:e5:fd:ed   C                     eth0
_gateway                 ether   52:54:00:12:35:00   C                     eth0
Entries: 4      Skipped: 0      Found: 4
root@kali:~# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.4                 ether   08:00:27:ad:87:b3   C                     eth0
10.0.2.5                 ether   08:00:27:27:d6:c7   C                     eth0
10.0.2.3                 ether   08:00:27:e5:fd:ed   C                     eth0
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
```

nmap command

Nmap is Linux command-line tool for network exploration and security auditing. This tool is generally used by hackers and cybersecurity enthusiasts and even by network and system administrators. It is used for the following purposes:

- ❏ Real time information of a network
- ❏ Detailed information of all the IPs activated on your network
- ❏ Number of ports open in a network
- ❏ Provide the list of live hosts
- ❏ Port, OS and Host scanning

$ sudo apt update
$sudo apt-get install nmap

Nmap is a powerful network scanning tool for security audits and penetration testing. It is one of the essential tools used by network administrators to troubleshooting network connectivity issues and port scanning .

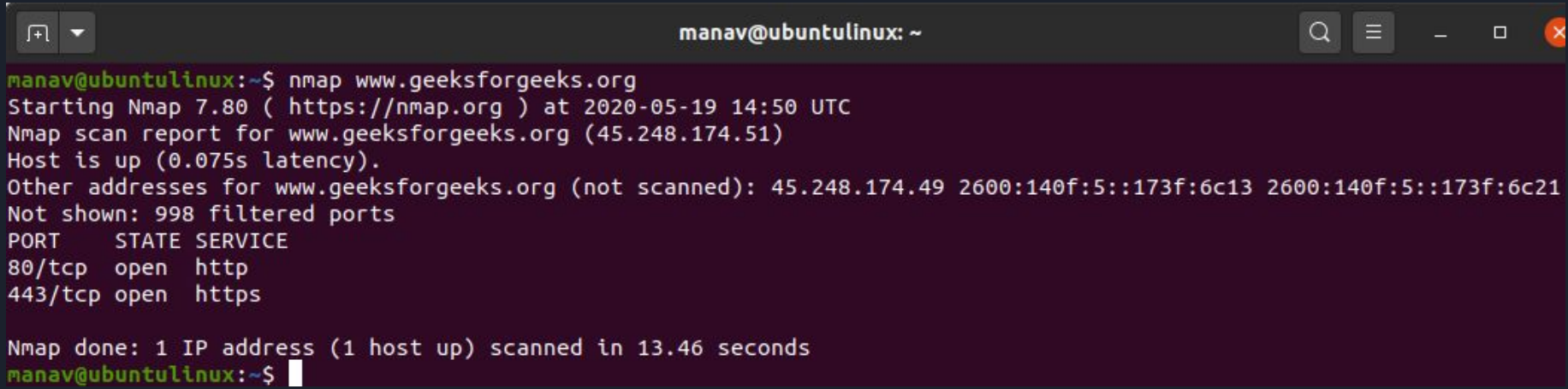
Nmap can also detect the Mac address, OS type, service version, and much more.

The simplified syntax of the nmap command is as follows:

```
nmap [Options] [Target...]
```

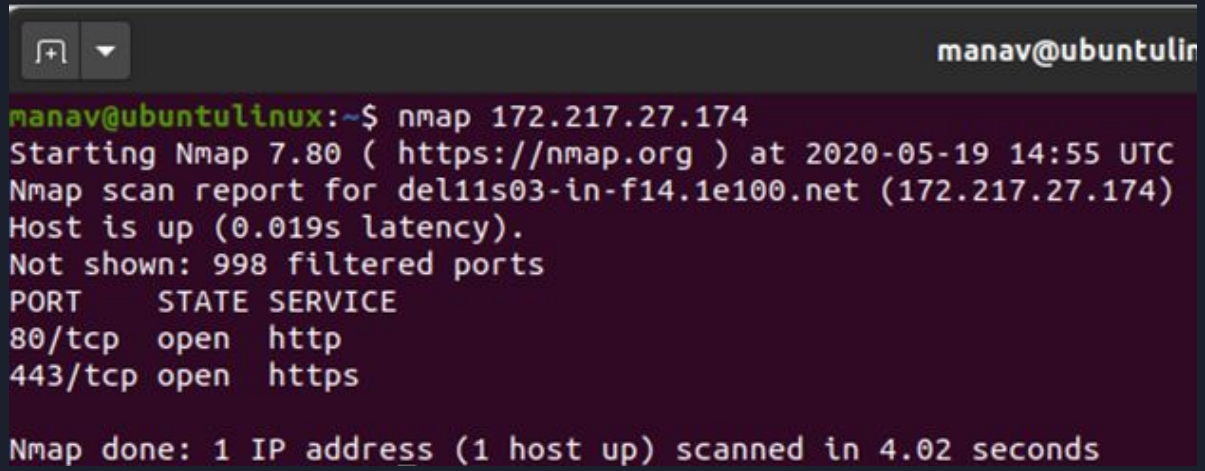
When invoked as a non-root user that does not have raw packet privileges, nmap runs TCP connect scan. The (-sT) is turned on by default in unprivileged mode.

The output will look something like this, including basic information about the scan and a list of open and filtered TCP ports.



Source image: https://www.geeksforgeeks.org/nmap-command-in-linux-with-examples/

We can also scan using IP Address:

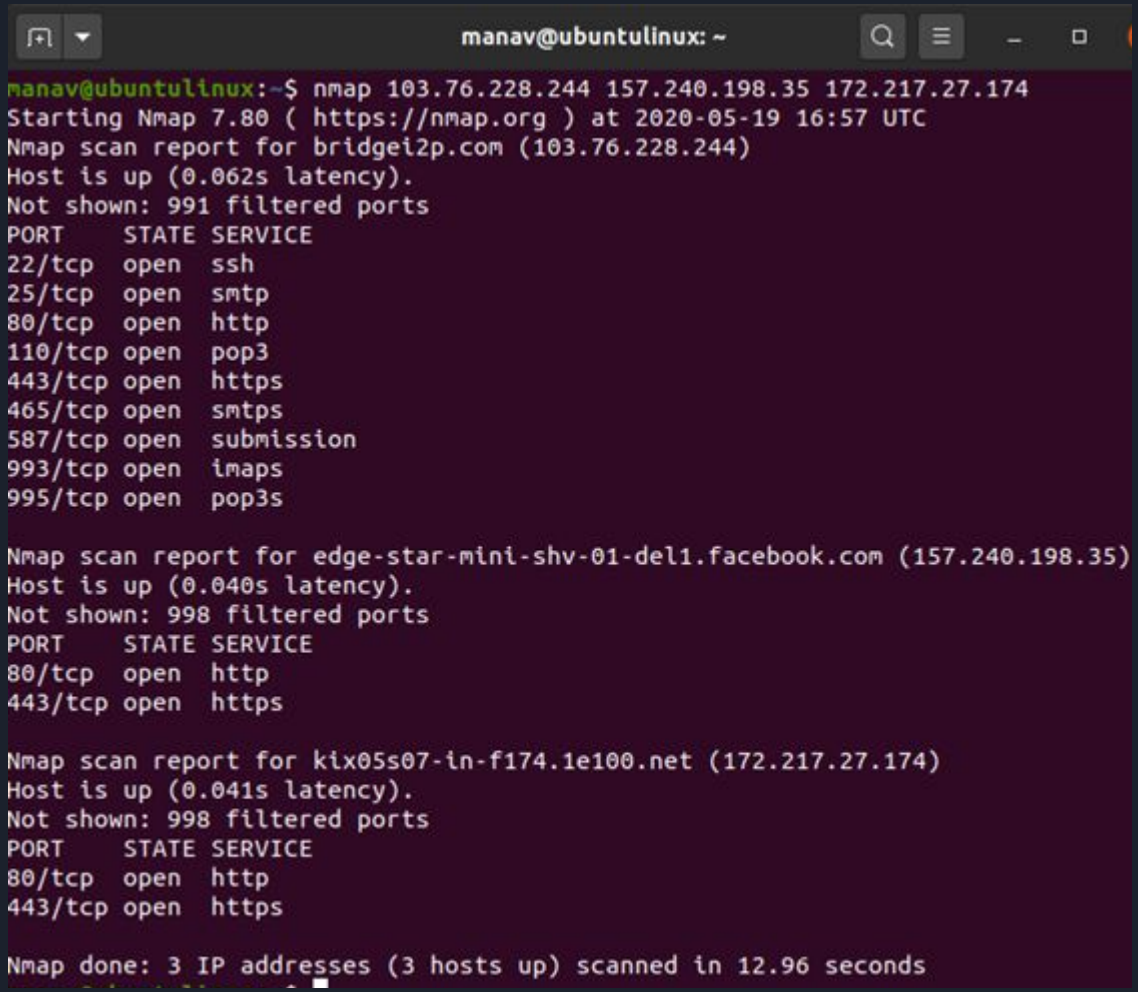


The nmap command allows scanning a system in various ways. In this we are performing a scan using the hostname as “geeksforgeeks” and IP address “172.217.27.174”, to find all open ports, services, and MAC addresses on the system.

We can also scan using “-v” option, to get more verbose information.

We can scan multiple hosts by writing IP addresses or hostnames with nmap.

We can do this by just writing the multiple IP addresses with a space in between .



To scan whole subnet:

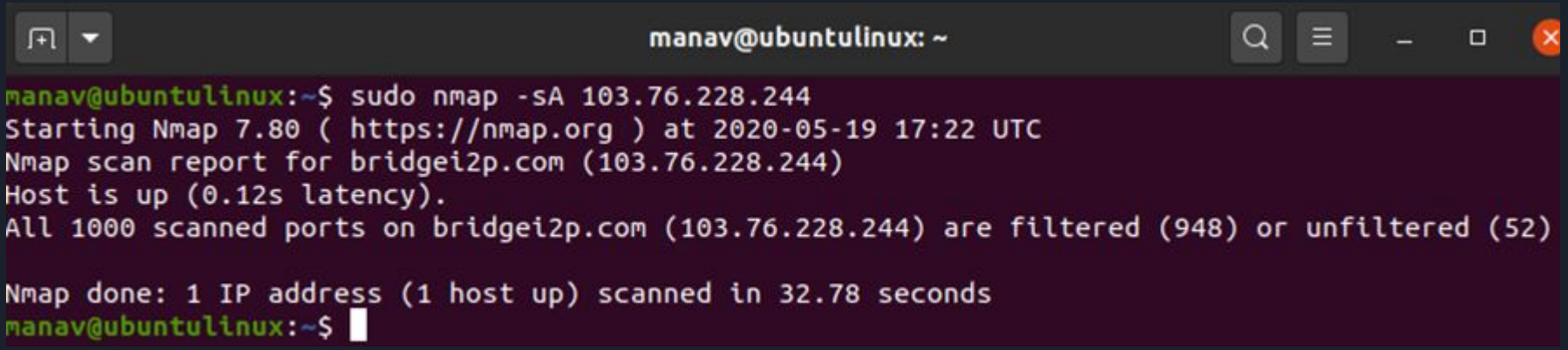
nmap 103.76.228.*

We can scan a whole subnet or IP range with nmap by providing “*” with it.

It will scan a whole subnet and give the information about those hosts which are Up in the Network.

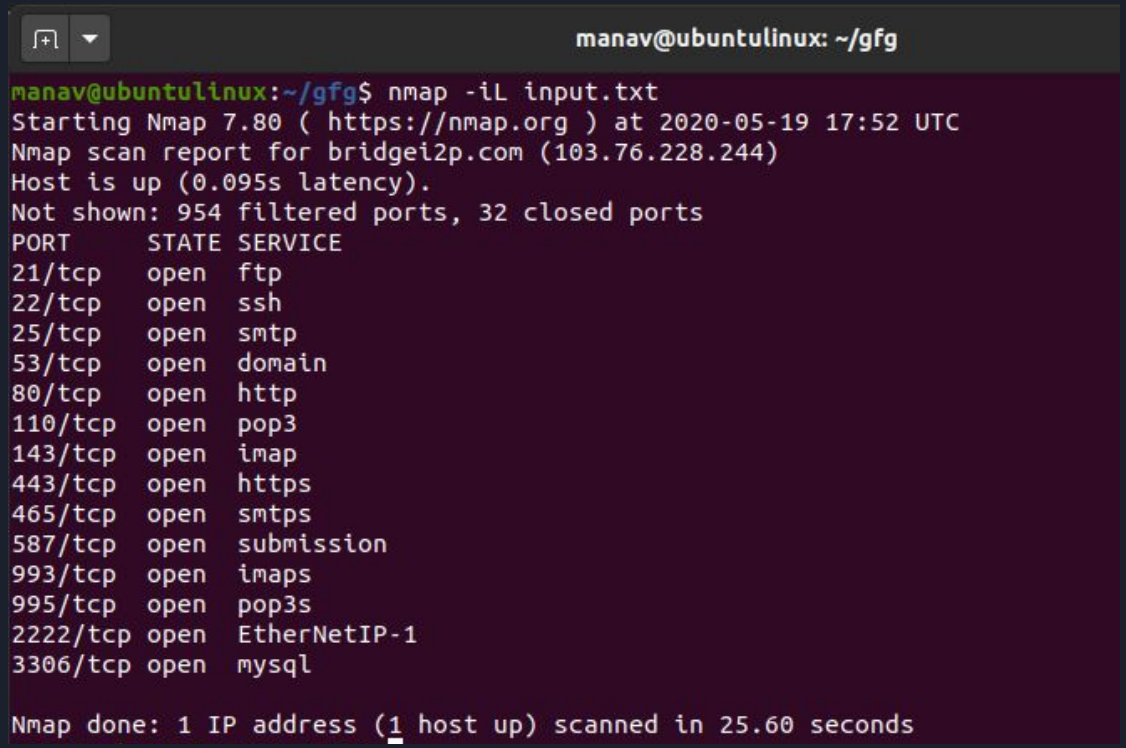
We can also check the Firewall settings using the “-sA” option.

This will provide you with information about firewall being active on the host. It uses an ACK scan to receive the information. Detecting firewall settings can be useful during penetration testing and vulnerability scans.



–iL option:

If we have a long list of addresses that we need to scan, we can directly import a file through the command line. It will produce a scan for the given IP addresses.



OS, Service and Version Detection

Nmap can detect the remote host operating system using TCP/IP stack fingerprinting. To run OS detection, invoke the command with the -O option:

sudo nmap -O scanme.nmap.org

If Nmap can detect the host OS, it will print something like below:

```
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.4
Network Distance: 18 hops

OS detection performed. Please report any incorrect results at https://nmap.org/subr
Nmap done: 1 IP address (1 host up) scanned in 26.47 seconds
```

Source: https://linuxize.com/post/nmap-command/

SSH (Secure Shell)

- SSH is a protocol for secure remote access to a machine over untrusted networks.
- It provides secure transmission.
- Improvement over Telnet as telnet used to send data as plain text and host between sender and receiver can see the traffic.
- It provides secure command shell, it allows us to transfer a file securely.
- Diffie Hellman key exchange algorithm is used to implement SSH
- It provides strong encryption and server authentication.
- Server normally listens for connections on port 22.

Working of SSH

1. When the SSH is started, as mentioned before, it starts listening on port 22 for a socket.
2. When that particular socket gets in connection established state, the secure shell spawns a child process using fork(), exec() system calls.
3. It generates a secure key using Diffie hellman key exchange algorithm
4. Once, this key gets generated, the Secure shell daemon is ready for the local client to connect to another secure shell daemon or waits for a connection from remote host.

Read more at https://datatracker.ietf.org/doc/html/rfc4253#section-14

```
(base) shashankrustagi@Shashanks-MacBook-Air ~ % ssh iiitd@192.168.2.233
iiitd@192.168.2.233's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-76-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Oct  7 11:46:58 IST 2021

  System load:  0.0               Processes:           98
  Usage of /:   18.8% of 15.68GB  Users logged in:     0
  Memory usage: 27%               IP address for ens32: 192.168.2.233
  Swap usage:   3%


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

50 updates can be applied immediately.
26 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Thu Oct  7 11:24:50 2021 from 192.168.1.99
iiitd@ubuntu:~$
```

Secure Copy (SCP)

SCP (secure copy) is a command-line utility that allows you to securely copy files and directories between two locations.

With `scp`, you can copy a file or directory:

- From your local system to a remote system.
- From a remote system to your local system.
- Between two remote systems from your local system.

When transferring data with `scp`, both the files and password are encrypted so that anyone snooping on the traffic doesn't get anything sensitive.

In this tutorial, we will show you how to use the `scp` command through practical examples and detailed explanations of the most common scp options.

scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2

`scp` provides a number of options that control every aspect of its behavior. The most widely used options are:

- `-P` - Specifies the remote host ssh port.
- `-p` - Preserves files modification and access times.
- `-q` - Use this option if you want to suppress the progress meter and non-error messages.
- `-C` - This option forces `scp` to compresses the data as it is sent to the destination machine.
- `-r` - This option tells `scp` to copy directories recursively.

The `scp` command relies on `ssh` for data transfer, so it requires an ssh key or password to authenticate on the remote systems.

Copy Files and Directories Between Two Systems with `scp`

Copy a Local File to a Remote System with the `scp` Command

To copy a file from a local to a remote system run the following command:

```
$ scp file.txt remote_username@10.10.0.2:/remote/directory
```

Where `file.txt` is the name of the file we want to copy, `remote_username` is the user on the remote server, `10.10.0.2` is the server IP address. The `/remote/directory` is the path to the directory you want to copy the file to. If you don't specify a remote directory, the file will be copied to the remote user home directory.

You will be prompted to enter the user password, and the transfer process will start.

```
Output
remote_username@10.10.0.2's password:
file.txt                                      100%    0     0.0KB/s   00:00
```

```
(base) shashankrustagi@Shashanks-MacBook-Air Desktop % scp file1.txt iiitd@192.168.2.233:
iiitd@192.168.2.233's password:
file1.txt                                     100%   18     0.1KB/s   00:00
```

Omitting the filename from the destination location copies the file with the original name. If you want to save the file under a different name, you need to specify the new file name:

```
$ scp file.txt remote_username@10.10.0.2:/remote/directory/newfilename.txt
```

If SSH on the remote host is listening on a port other than the default 22 then you can specify the port using the `-P` argument:

```
(base) shashankrustagi@Shashanks-MacBook-Air Desktop % scp -P 233 file1.txt iiitd@192.168.2.233:
ssh: connect to host 192.168.2.233 port 233: Connection refused
lost connection
(base) shashankrustagi@Shashanks-MacBook-Air Desktop %
```

The command to copy a directory is much like as when copying files. The only difference is that you need to use the `-r` flag for recursive.

```
(base) shashankrustagi@Shashanks-MacBook-Air Desktop % scp -r folder iiitd@192.168.2.233:
iiitd@192.168.2.233's password:
```

THANK YOU :)