Topic — Networking

How to scan for IP addresses on your network with Linux

Published January 3, 2019

Written By
Jack Wallen

Table of Contents

Are you having trouble remembering what IP addresses are in use on your network? Jack Wallen shows you how to discover those addresses with two simple commands.

How many times have you tried to configure a static IP address for a machine on your network, only to realize you had no idea what addresses were already taken? If you happen to work with a desktop machine, you could always install a tool like Wireshark to find out what addresses were in use. But what if you're on a GUI-less server? You certainly won't rely on a graphical-based tool for scanning IP addresses. Fortunately, there are some very simple-to-use command line tools that can handle this task.

I'm going to show you how to scan your Local Area Network (LAN) for IP addresses in use with two different tools (one of which will be installed on your server by default). I'll demonstrate on Ubuntu Server 18.04.

Let's get started.

SEE: Server deployment/migration checklist (Tech Pro Research)

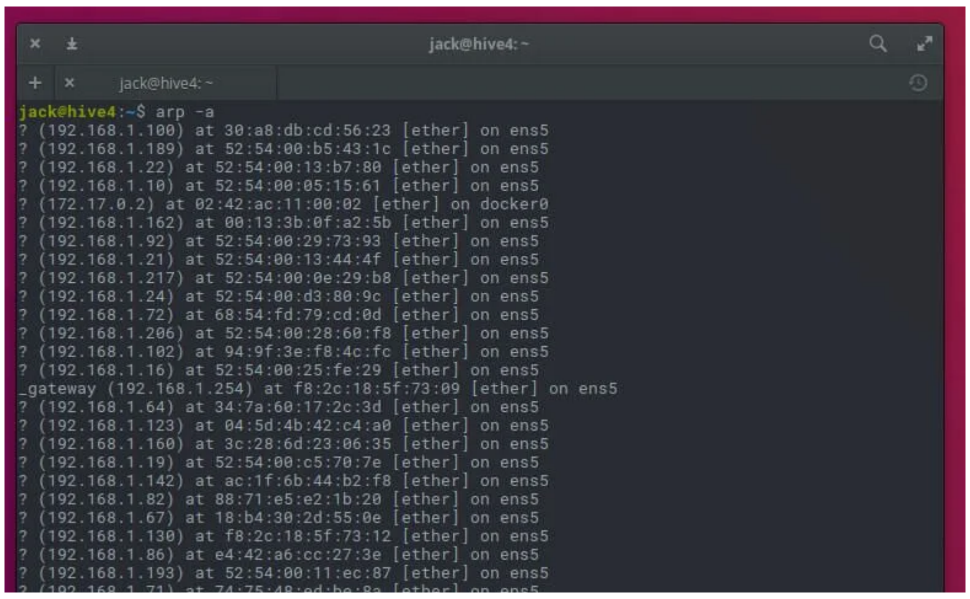
The arp command

The first tool we'll use for the task is the built-in *arp* command. Most IT admins are familiar with *arp*, as it is used on almost every platform. If you've never used *arp* (which stands for Address Resolution Protocol), the command is used to manipulate (or display) the kernel's IPv4 network neighbor cache. If you issue *arp* with no mode specifier or options, it will print out the current content of the ARP table. That's not what we're going to do. Instead, we'll issue the command like so:

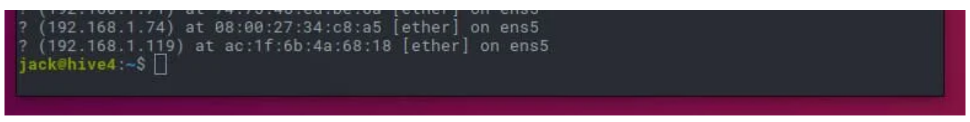
```
arp -a
```

The -a option uses and alternate BSD-style output and prints all known IP addresses found on your LAN. The output of the command will display IP addresses as well as the associated ethernet device (**Figure A**).

Figure A



```
jack@hive4:~$ arp -a
? (192.168.1.100) at 30:a8:db:cd:56:23 [ether] on ens5
? (192.168.1.189) at 52:54:00:b5:43:1c [ether] on ens5
? (192.168.1.22) at 52:54:00:13:b7:80 [ether] on ens5
? (192.168.1.10) at 52:54:00:05:15:61 [ether] on ens5
? (172.17.0.2) at 02:42:ac:11:00:02 [ether] on docker0
? (192.168.1.162) at 00:13:3b:0f:a2:5b [ether] on ens5
? (192.168.1.92) at 52:54:00:29:73:93 [ether] on ens5
? (192.168.1.21) at 52:54:00:13:44:4f [ether] on ens5
? (192.168.1.217) at 52:54:00:0e:29:b8 [ether] on ens5
? (192.168.1.24) at 52:54:00:d3:80:9c [ether] on ens5
? (192.168.1.72) at 68:54:fd:79:cd:0d [ether] on ens5
? (192.168.1.206) at 52:54:00:28:60:f8 [ether] on ens5
? (192.168.1.102) at 94:9f:3e:f8:4c:fc [ether] on ens5
? (192.168.1.16) at 52:54:00:25:fe:29 [ether] on ens5
_gateway (192.168.1.254) at f8:2c:18:5f:73:09 [ether] on ens5
? (192.168.1.64) at 34:7a:60:17:2c:3d [ether] on ens5
? (192.168.1.123) at 04:5d:4b:42:c4:a0 [ether] on ens5
? (192.168.1.160) at 3c:28:6d:23:06:35 [ether] on ens5
? (192.168.1.19) at 52:54:00:c5:70:7e [ether] on ens5
? (192.168.1.142) at ac:1f:6b:44:b2:f8 [ether] on ens5
? (192.168.1.82) at 88:71:e5:e2:1b:20 [ether] on ens5
? (192.168.1.67) at 18:b4:30:2d:55:0e [ether] on ens5
? (192.168.1.130) at f8:2c:18:5f:73:12 [ether] on ens5
? (192.168.1.86) at e4:42:a6:cc:27:3e [ether] on ens5
? (192.168.1.193) at 52:54:00:11:ec:87 [ether] on ens5
? (192.168.1.71) at 74:75:48:ed:be:8a [ether] on ens5
```



```
? (192.168.1.74) at 08:00:27:34:c8:a5 [ether] on ens5
? (192.168.1.119) at ac:1f:6b:4a:68:18 [ether] on ens5
jack@hive4:~$ 
```

You now have a listing of each IP address in use on your LAN. The only caveat, is that (unless you know the MAC address of every device on your network), you won't have a clue as to which machine the IP addresses are assigned. Even without knowing what machine is associated with what address you at least know what addresses are being used.

Nmap

Next, we use a command that offers more options. Said command is *nmap*. You won't find nmap installed on your Linux machine by default, so we must add it to the system. Open a terminal window (or log into your GUI-less server) and issue the command:

```
sudo apt-get install nmap -y
```

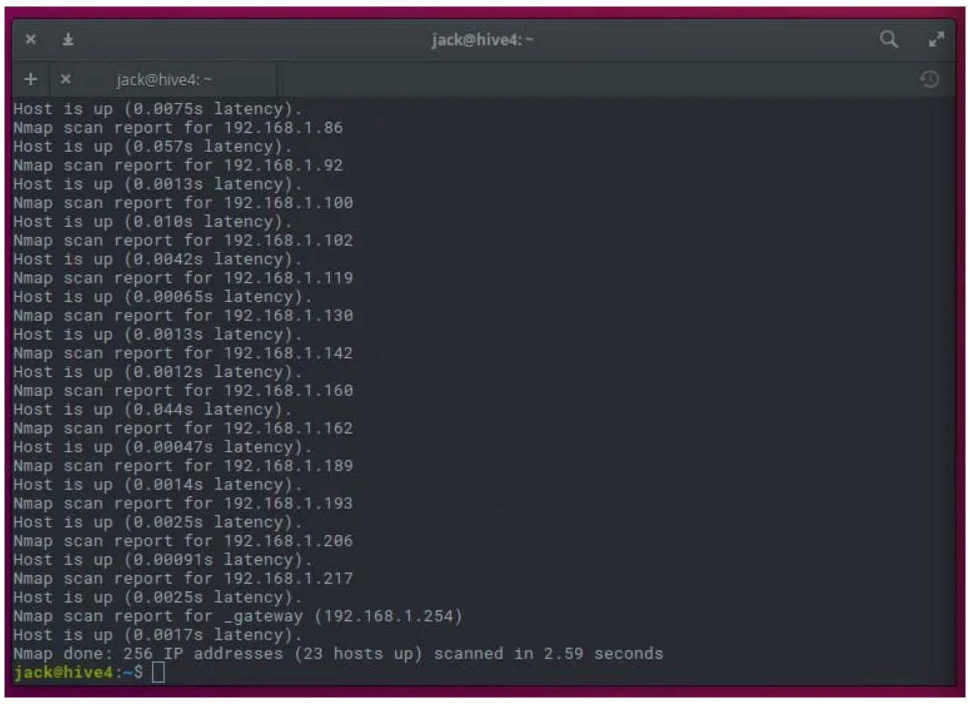
Once the installation completes, you are ready to scan your LAN with nmap. To find out what addresses are in use, issue the command:

```
nmap -sP 192.168.1.0/24
```

Note: You will need to alter the IP address scheme to match yours.

The output of the command (**Figure B**), will show you each address found on your LAN.

Figure B



```
Host is up (0.0075s latency).
Nmap scan report for 192.168.1.86
Host is up (0.057s latency).
Nmap scan report for 192.168.1.92
Host is up (0.0013s latency).
Nmap scan report for 192.168.1.100
Host is up (0.010s latency).
Nmap scan report for 192.168.1.102
Host is up (0.0042s latency).
Nmap scan report for 192.168.1.119
Host is up (0.00065s latency).
Nmap scan report for 192.168.1.130
Host is up (0.0013s latency).
Nmap scan report for 192.168.1.142
Host is up (0.0012s latency).
Nmap scan report for 192.168.1.160
Host is up (0.044s latency).
Nmap scan report for 192.168.1.162
Host is up (0.00047s latency).
Nmap scan report for 192.168.1.189
Host is up (0.0014s latency).
Nmap scan report for 192.168.1.193
Host is up (0.0025s latency).
Nmap scan report for 192.168.1.206
Host is up (0.00091s latency).
Nmap scan report for 192.168.1.217
Host is up (0.0025s latency).
Nmap scan report for _gateway (192.168.1.254)
Host is up (0.0017s latency).
Nmap done: 256 IP addresses (23 hosts up) scanned in 2.59 seconds
jack@hive4:~$ 
```

Let's make *nmap* more useful. Because it offers a bit more flexibility, we can also discover what operating system is associated with an IP address. To do this, we'll use the options -sT (TCP connect scan) and -O (operating system discovery). The command for this is:

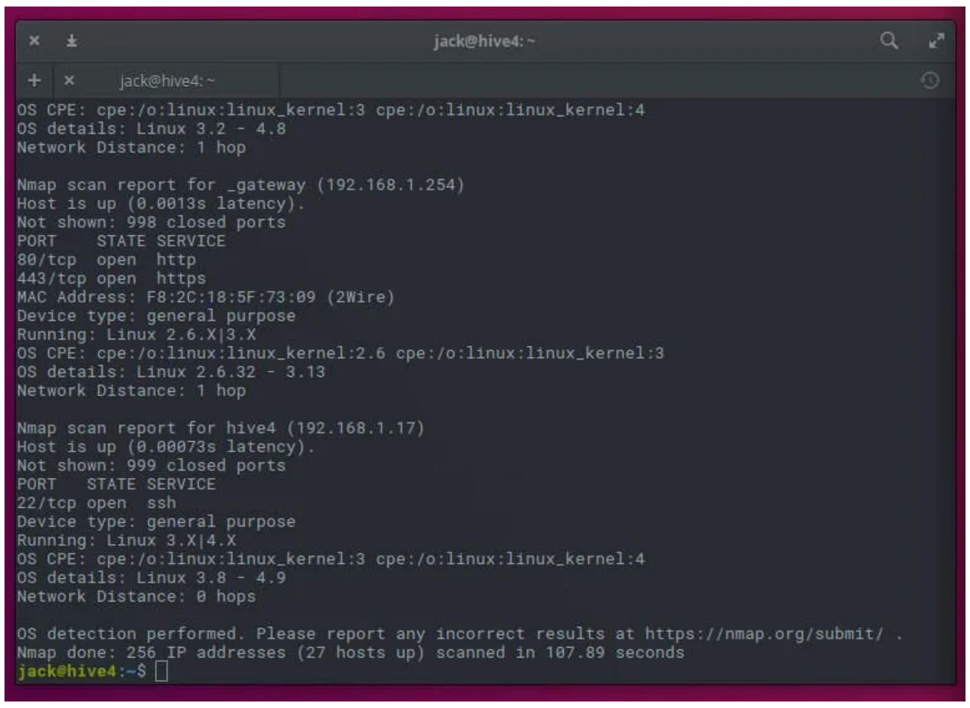
```
sudo nmap -sT -O 192.168.1.0/24
```

Depending on the size of your network, this command can take some time. And if your network is large, consider sending the output of the command to a file like so:

```
sudo nmap -sT -O 192.168.1.0/24 > nmap_output
```

You can then view the file with a text editor to find out what operating system is attached to an IP address (**Figure C**).

Figure C



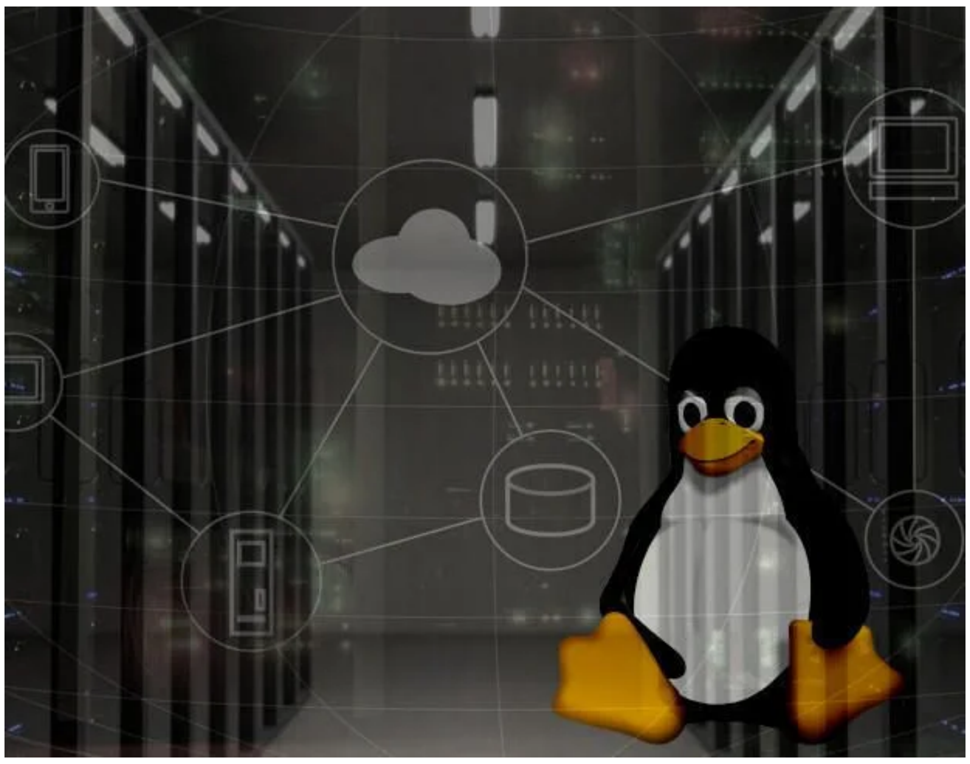
```
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.8
Network Distance: 1 hop

Nmap scan report for _gateway (192.168.1.254)
Host is up (0.0013s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
MAC Address: F8:2C:18:5F:73:09 (2Wire)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.13
Network Distance: 1 hop

Nmap scan report for hive4 (192.168.1.17)
Host is up (0.00073s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.8 - 4.9
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (27 hosts up) scanned in 107.89 seconds
jack@hive4:~$
```

With the help of these two simple commands, you can locate IP addresses on your network that are in use. Now, when you're assigning a static IP address, you won't accidentally assign one already in use. We all know what kind of headaches that can cause.



Subscribe to the Developer Insider Newsletter

From the hottest programming languages to commentary on the Linux OS, get the developer and open source news and tips you need to know. Delivered Tuesdays and Thursdays

Email Address Work Email Address

□ By signing up to receive our newsletter, you agree to our Terms of Use and Privacy Policy. You can unsubscribe at any time.

Subscribe

Share Article

Jack Wallen

Jack Wallen is an

award-winning writer for TechRepublic, The New Stack, and Linux New Media. He's covered a variety of topics for over twenty years and is an avid promoter of open source. For more news about Jack Wallen, visit his website jackwallen.com.

See all of Jack's content →