In this lesson you will learn to use a network scanning tool named nmap. This tool is helpful in finding IP addresses of devices connected to a local network and obtaining additional information about the configuration of those devices. You should install nmap on your Raspberry Pi. You might also consider installing nmap on your laptop computer, especially if you think you might use it for the NRC IoT contest.

Installing nmap on Raspberry Pi

1. Open the Terminal
2. sudo apt-get install nmap

Installing nmap on a PC or Apple computer

1. go to https://nmap.org/download.html
2. download nmap for your computer:
   PC - Latest stable release self-installer: nmap-7.60-setup.exe
   Apple - Latest stable release installer: nmap-7.60.dmg

Determine IP address of your RPi

You will need to have the IP address of the local network you wish to scan. This is easily obtained in the Terminal window of your RPi by issuing this command:

```
hostname -I
```

This commend will list both the IPv4 and IPv6 versions of the IP address, the IPv4 version listed first. Recall that version 4 format is a set of four numbers separated by periods.

Determine IP address of your PC

Open the Command Prompt (select Run in the Windows menu and enter the text cmd to open the Command Prompt). In the Command Prompt window enter this command:

```
ipconfig
```

In the information listed, find the IPv4 address.

Determine the IP address of your Apple computer

Open the Terminal (open Finder, select Applications, then select Utilities folder, then Terminal). Enter this command in the Terminal window:

`ifconfig`

There will be a lot of information displayed in the terminal. Look for the line that starts with inet to find the IPv4 address.

Determine the MAC addresses of your Raspberry Pi

In the Terminal window issue this command:

`ifconfig`

Command **ifconfig** will return information about your network connection . Below, I have edited an example to include only specific elements of interest for this lesson. Information following **eth0:** refers to the Ethernet network device (a wired network connection) and information after **wlan0:** is information regarding the WiFi network device (a wireless network connection).

```
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:ac:f3:aa  txqueuelen 1000  (Ethernet)


wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 189.19.63.127  netmask 255.255.255.0  broadcast 189.19.63.255
        inet6 2002:4229:37ed:0:c988:d411:1619:9a37  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::95c5:b105:5a0c:5064  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:f9:b7:9e  txqueuelen 1000  (Ethernet)
```

The above lines are portions from a listing of an RPi model 3, which contains an Ethernet network adapter and WiFi network adapter. Each adapter has a MAC address which never changes. In the example above the MAC address of the Ethernet adapter is **`b8:27:eb:ac:f3:aa`** The MAC address of the WiFi adapter is **`b8:27:eb:f9:b7:9e`** In the above example, the RPi is connected to the network by its WiFi adapter. We know this because we find the IP address listed under the WiFi (wlan0) section: **`inet 189.19.63.127`**

Attach a label to the case of your Raspberry Pi and write on it the MAC addresses of your Ethernet adapter and your WiFi adapter. These numbers will come in handy later.

Suppose you have just arrived for the IoT contest at NRC. In formulating your plan, you decide to use a RPi to control a robot. That RPi will be one of two RPIs to be used for your entry. The other one will be connected to one of GEAR's HDMI monitors. However, the one on the robot will be "headless", *i.e.*, not connected to a display. To make an initial connection to the network at the NRC it may be necessary to temporarily connect a monitor to it. Then connect to the network using the password provided by the contest judge. Now that the password is stored on the RPi, shut it down and disconnect the monitor. When you power up the RPi again, it might be assigned a different IP address by the router at the NRC. If that happens, you can discover the new IP address remotely from another RPi or computer using nmap. Suppose you are using another RPi to do the network scanning. In a terminal window enter a command like the one below, but substitute the first three sections of the IP for the local network at the NRC (the command below uses a fictitious IP network):

`sudo nmap -T4 -F 189.19.63.*`

The fourth section of the IP address contains just the (*) character, which means scan all devices connected to this local network. The scan will return information for each device connected to the local network. Below I list the information for one example device on a fictitious network:

```
Nmap scan report for 189.19.63.3
Host is up (0.0041s latency).
Not shown: 98 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: B8:27:EB:8A:83:BC (Raspberry Pi Foundation)
```

The above listing is for the device assigned an IP of **`189.19.63.3`** on the network. The last line of the listing provides the MAC address of the device assigned that IP address. Suppose that MAC address matches the WiFi network adapter for the RPi you are using on the robot. You have just discovered the IP address for that RPi without connecting a monitor to it.

You may also notice that nmap has identified the above device as a Raspberry Pi. How was that determination possible? MAC addresses that begin with B8:27:EB have been assigned to the Raspberry Pi Foundation for use in their manufacturing processes. Therefore, if the first 6 letters and numbers of a MAC address are B8:27:EB, nmap knows the device was manufactured by the Raspberry Pi Foundation. This is handy when you are scanning a network that has many connected devices. Each device can be identified by its manufacturer.

Also notice that the above listing includes the ports on the device that are open (port 22 for ssh service and port 80 for http). This happens to be a RPi running a web server on port 80. This is also handy information. Suppose you are attempting to connect to this device remotely from another computer using ssh. If port 22 is not open for ssh service, then that would explain why you can't connect!

Let us look at another device that was listed when I issued the **`sudo nmap -T4 -F 189.19.63.*`** command:

```
Nmap scan report for 189.19.63.10
Host is up (0.016s latency).
Not shown: 94 closed ports
PORT     STATE SERVICE
80/tcp   open  http
443/tcp  open  https
515/tcp  open  printer
631/tcp  open  ipp
8080/tcp open  http-proxy
9100/tcp open  jetdirect
MAC Address: FC:3F:DB:B3:4B:C6 (Hewlett Packard)
```

The above listing is for a HP laser printer connected to the network. The printer has been assigned the IP of `189.19.63.10` and it has 6 open ports that are used for communications. The MAC address of the printer is also listed.

Let us take a look again at the command that yielded the listings: **`sudo nmap -T4 -F 189.19.63.*`** That command will yield a listing for every device on the network, although I only provided the listings for two devices. The nmap command has many options and all of them are listed if you just enter the nmap command by itself. In the example I gave, the options used were -T4 and -F, which yield the listings in the format seen above. Let us try another nmap command with additional options (see next page).

sudo nmap -sV -T4 -O -F --version-light 189.19.63.*

```
Nmap scan report for 189.19.63.3
Host is up (0.0060s latency).
Not shown: 98 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     (protocol 2.0)
80/tcp open  http    Apache httpd 2.4.25 ((Raspbian))
MAC Address: B8:27:EB:8A:83:BC (Raspberry Pi Foundation)
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 3.2 - 4.6
Network Distance: 1 hop
```

In this example the options -sV and -O are added plus the specification for the light version, which perhaps does not yield as much information as the regular (default) version. Here we see the same information we saw before, but with some additional information. Now we see Apache listed on port 80. Apache is the brand of web server running on port 80. Also notice that information on the operating system software is provided. Now let's take a look at an additional device listing provided with the same namp command.

```
Nmap scan report for 189.19.63.25
Host is up (0.0032s latency).
Not shown: 95 filtered ports
PORT     STATE SERVICE      VERSION
80/tcp   open  http         Microsoft IIS httpd 10.0
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: A0:D3:C1:3D:A4:1C (Hewlett Packard)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1
closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP (85%)
```

```
OS CPE: cpe:/o:microsoft:windows_xp::sp2
Aggressive OS guesses: Microsoft Windows XP SP2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: GEORGE-HP; OS: Windows; CPE: cpe:/o:microsoft:windows
```

The above listing is for a HP desktop computer running Windows 10. Notice that nmap is warning that its attempt to discern the operating system may be unreliable. But on port 455 there is some software operating that is Windows 7 – 10. In any case, nmap has determined this is a computer running Windows software. The MAC address of its Ethernet adapter belongs to the group manufactured by Hewlett Packard, so this is a HP computer, which is correct. Also notice that nmap was able to determine the host name of the computer: **GEORGE-HP**, which is correct. I also found that nmap could find the host name of another computer on the network that was running Windows 8. However, nmap did not list the host names of the two Raspberry Pis that were on the network. Therefore, if we are trying to identify the host name of a RPi, we will have to try some other tool.

Below is the listing for the other RPi on the network. This happens to the RPi I used to run the nmap scan. Notice that in this case the MAC address is not listed. Apparently, nmap will not list the MAC address of the device that is being used to run the scan.

```
Nmap scan report for 189.19.63.127
Host is up (0.000053s latency).
Not shown: 99 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     (protocol 2.0)
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 - 4.1
Network Distance: 0 hops

Nmap done: 256 IP addresses (9 hosts up) scanned in 97.33 seconds
```

The line directly above is the last line displayed in the scan output. Of the 256 possible addresses (0 through 255) on the network of 189.19.63, nine devices were found to be connected and operating. The scan took 97.33 seconds to complete. The length of time that a scan takes will depend on the number of hosts (devices) found and how much detail you ask for with the options you apply. If you want a lot of detail, the scan might take a long time. If all you need are the IP addresses and MAC addresses, then the next is example is what you want.

```
sudo  nmap  -sn   189.19.63.*

     Nmap scan report for 189.19.63.3
     Host is up (0.015s latency).
     MAC Address: B8:27:EB:8A:83:BC (Raspberry Pi Foundation)

     Nmap scan report for 189.19.63.127
     Host is up.

     Nmap done: 256 IP addresses (9 hosts up) scanned in 6.49 seconds
```

In the above example the only option specified is **-sn**. This option prevents the scanning of each active port on each device. Accordingly, the scan took only 6.49 seconds to complete. I have included only the listings for the two RPis on the network. The scan was issued from the RPi at IP 189.19.63.127. Notice that the only information supplied is that the host is up. For the other RPi (Grant's) we get the MAC address.

Finding the host names of Raspberry Pis on a network.

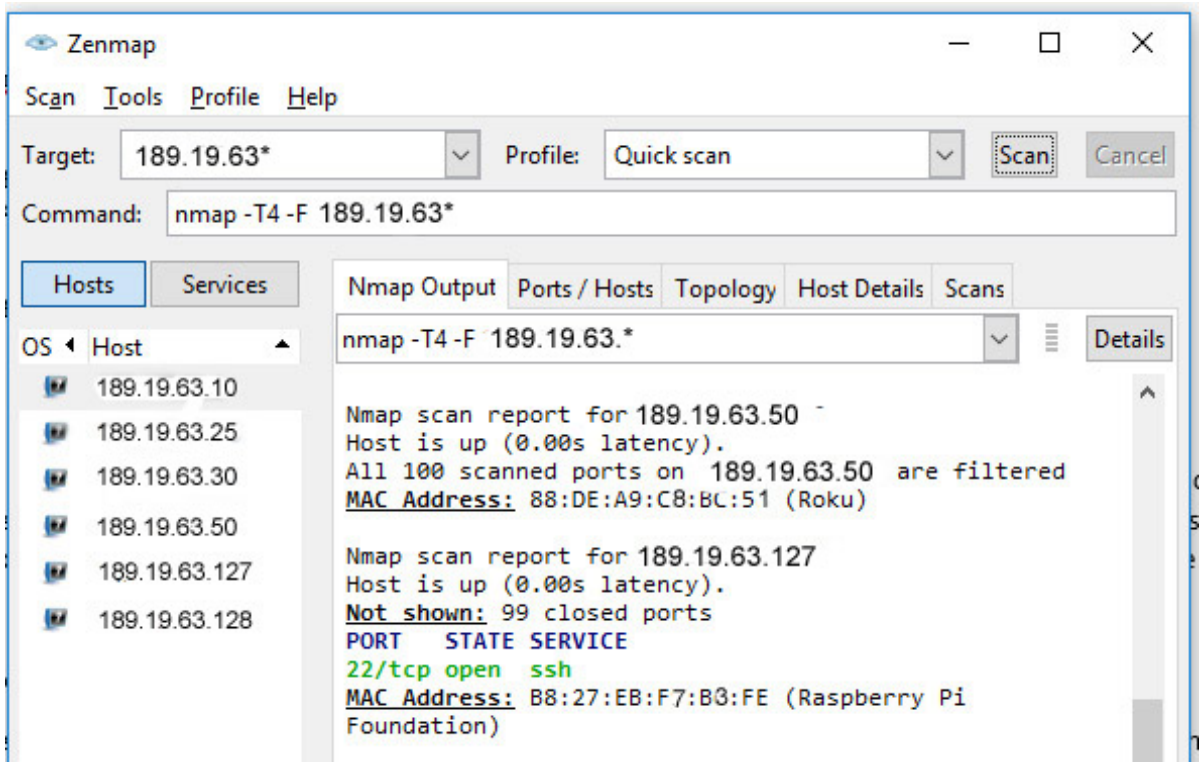
As we have already seen, nmap won't provide the host names of RPis on the network. However, if we know the host name we can match it to an IP address by issuing a command in the terminal of another RPi. Suppose we know that a RPi with host name of Robot1 is connected to the network, but we don't know the IP address. Then enter this command in the terminal:

```
ping Robot1.local
```

The IP address will be returned by the ping command.

Running nmap on your PC or Apple computer

You might like to try nmap on your PC or Apple computer. There is a GUI (graphical user interface) version installed with nmap named Zenmap that is handy. Let us see how that works. When I installed the software on my PC, I selected the option to put a short-cut on the desktop. The short-cut is named **Nmap - Zenmap GUI.** (see next page for more on Zenmap)



The image above is a screen capture of Zenmap. In the **Target** slot notice the exact way I specified the IP of the network. In the **Profile** drop-down slot I selected a "Quick Scan." In the **Command** slot is automatically listed for me the equivalent nmap command. Notice that the options are -T4 and -F. On the left side of the Zenmap window is a convenient list of all IP addresses found by the scan. On the right side of the window the details of the scan are listed. In the image above you can see the results for only two IP addresses (in the actual application window you can scroll to see all listings).

If you have Nmap - Zenmap installed on your computer, then give it a try.

1. Enter the IP in the **Target** slot, specifying numbers in first three segments and then * for the fourth segment
2. Select the type of scan you wish to perform in the **Profile** drop-down
3. Click the **Scan** button