Python ARP-Scan

In order to access easily higher order functions on the RPi such as SenseHAT and messaging protocols, we'll now switch to using Python, a good general purpose programming library that's already installed on the RPi,

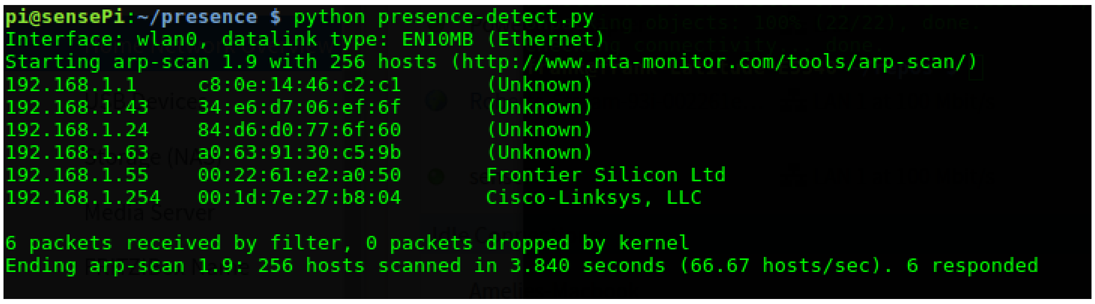
Scanning for MAC addresses with Python

We can call the `arp-scan` program from a Python program using the `subprocess` library.

- In the `presence` directory you created earlier, create a new file called `presence-detector.py` with the following content:

```
#!/usr/bin/env python
#coding=utf-8

import subprocess

def arp_scan():
    output = subprocess.check_output("sudo arp-scan -l", shell=True)
    print output

arp_scan()
```

- Run the program by typing `python presence-detector.py` on the command prompt. You should see the `arp-scan` output printed on the console similar to the following:



```
pi@sensePi:~/presence $ python presence-detect.py
Interface: wlan0, datalink type: EN10MB (Ethernet)
Starting arp-scan 1.9 with 256 hosts (http://www.nta-monitor.com/tools/arp-scan/)
192.168.1.1     c8:0e:14:46:c2:c1       (Unknown)
192.168.1.43    34:e6:d7:06:ef:6f       (Unknown)
192.168.1.24    84:d6:d0:77:6f:60       (Unknown)
192.168.1.63    a0:63:91:30:c5:9b       (Unknown)
192.168.1.55    00:22:61:e2:a0:50       Frontier Silicon Ltd
192.168.1.254   00:1d:7e:27:b8:04       Cisco-Linksys, LLC

6 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9: 256 hosts scanned in 3.840 seconds (66.67 hosts/sec). 6 responded
```

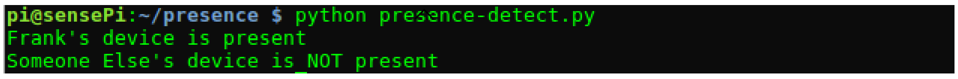
arp scan with python

Using this program we can now get at the MAC address list programatically.

- To search the output for a particular MAC addresses, lets introduce two lists into our program, device owner names (`names`) and corresponding MAC addresses (`macs`). Notice the order of the names and devices correlate(i.e. Frank's device MAC address is `"d4:28:d5:37:7e:a2"`). **As before, you should add a name and device that is present on the local network.**

```
#!/usr/bin/env python
#coding=utf-8

import subprocess

#Names of device owners
names = ["Frank","Someone Else"]

# MAC addresses of devices
macs = ["d4:28:d5:37:7e:a2","xx:xx:xx:xx:xx:xx"]

def arp_scan():
    output = subprocess.check_output("sudo arp-scan -l", shell=True)
    for i in range(len(names)):
        if macs[i] in output:
            print(names[i] + "'s device is present")
        else:
            print(names[i] + "'s device is NOT present")

arp_scan()
```

- Test this program and make sure it works by placing a known device in the arrays. Run the program from the command line as before:



```
pi@sensePi:~/presence $ python presence-detect.py
Frank's device is present
Someone Else's device is_NOT present
```

arp scan with python

.

Next we'll use the SenseHat to output the results.