Instructor Note: Lets use this class to talk about the wonders of passively monitoring the network to see what data is available. If you log onto our environment via ssh, and connect to the maintenance vm (**10.10.0.20**), you should be able to run the command below, which will show all traffic not to/from this vm (otherwise we would just get inundated with the traffic that running this caused!

```
sudo tcpdump -i any -n | grep -v 10.10.0.20
```

Additionally this is a perfect time to talk about:

- Promiscuous mode
- Passive Network Taps
- Network Span Ports
- VM Network Traffic Redirection

Lab 3 - Network Mapping (Passive)

In our previous lab, we were actively mapping our network using Nmap to ping the address space around us, and then using the hosts we discovered to attempt to determine what services those hosts had running by actively interacting with them.

Actively scanning the network assumes a few things:

- We are comfortable being *loud* on our own network, scans are observable
- We assume that all services are configured to respond when queried without authentication

Observing our Networks

It is also advantageous to *passively* observe our networks from a host that can observe the traffic, either via a passive tap, a span port, or a switch that is configured to allow hosts to see traffic beyond what is destined for them.

Sniffing is an easy method to map networks and hosts, we will typically see lots of DNS, Web, and networking protocols that will reveal host names, active subnets, domain names, VLANs. If we monitor over a larger period of time, we can use this method to discover what IP address are in use. There are quite a few tools available to handle this task, **tcpdump** and **tshark** are common Linux command line utilities that are available to us. To achieve this we will have to be able to use **promiscuous mode** on our servers which will normally require root access.

TCPDump and Python

```
import subprocess

p = subprocess.Popen(('sudo', 'tcpdump', '-l'), stdout=subprocess.PIPE)
for row in iter(p.stdout.readline, b''):
    print row.rstrip()
```

The code above results in lines which would can be easily parsed in python, example output is show below, where ssh traffic is seen between the ssh entry server (10.10.0.10) and the template student server (10.10.2.99).

```
11:09:38.300941 IP 10.10.2.99.ssh > 10.10.0.10.46180: Flags [P.], seq 5136796:5138332, ack 295
11:09:38.301052 IP 10.10.0.10.46180 > 10.10.2.99.ssh: Flags [.], ack 5138332, win 7325, option
```

This weeks lab

Your assignment this week is to use python and tcpdump, in a fashion similar to the previous lab, to map the network.

Program requirements:

1. Your program should collect (and process) data until ctrl-C is pressed, you can surround your code with a `try: ... except KeyboardInterrupt: ...`
2. After ctrl-C is pressed it should provide, in a nice usable format, the hosts and ports that it discovered.
3. The output format will include, at a minimum,
    - Host (focus on those within 10.10.x.x for now)
    - Observed Port or Service Name
    - Count of times traffic was observed on this host and port

What to submit

When you have completed the lab **post all requested materials to your webpage**. Then, submit the link via email to your Professor.

The email subject line should be SY402 [Section Number] Lab [X]: Title of Lab (e.g., SY402 1111 Lab 3: Determining Whats Nearby With NMAP). Email sent with a different subject line will reduce the overall grade by 5 points.

The web page link(s) should include::

1. (Attach) your python program (.py).
2. (Attach) the output of your program (as a .txt file) that shows the results of the passive scan of the VM network in an easy to read format.
3. (Include) the location of a folder on your VM and includes the Python script. This will allow your instructor to run the program.

You will be graded both on your ability to scan the network **AND** your ability to present the data in a rational format. Your python script should be able to parse each line from the tcpdump output taking out the individual fields (such as host and port) and working upon that information individually. The ability to easily parse information is a strength of Python and a very useful skill.