Lab 2 - Determining Whats Nearby With NMAP

The objective of this lab is to build a baseline understanding of the hosts present on your network. **How can we defend out networks if we do not have an understanding of what they look like?**

Preparing for the lab

All labs are designed to be completed individually, but feel free to help each other out as long as you cite any support your receive.

We expect you to figure things out as you go, your instructor can help guide you in the right direction, but the discovery should be yours. Do not be afraid to make mistakes, this is the best way to learn, but do not expect the instructor to walk you through each step, there is **no** learning that way. Here is a nice **link** to a SANS Nmap cheat sheet.

To perform this lab you will connect to your virtual machine via SSH, there is **no need** to connect via the web interface unless we break something.

All of your scripts are to be run on your server only.

Using Open Source Tools

Lets start with the standard Open Source scanning tool, NMAP, and scan all of the 10.x.x.x networks that your machine can see (**Hint:** take a look at the diagram from the first lab). As a quick example, the following line will ping a range of IP addresses and save the results to a text file (ping.txt).

```
nmap -sP 10.10.0.0/24 -oG ping.txt
```

You should see an output similar to:

```
Starting Nmap 7.01 ( https://nmap.org ) at 2017-01-12 15:47 EST
Nmap scan report for 10.10.2.x
Host is up (0.0014s latency).
Nmap done: 256 IP addresses (# hosts up) scanned in 1.91 seconds
```

The **ping.txt** file that was created should be something similar to

```
# Nmap 7.01 scan initiated Sun Jan 12 15:47:01 2017 as: nmap -sP -oG outputfile.
txt 10.10.2.0/24
Host: 10.10.2.x ()  Status: Up
Host: 10.10.2.x ()  Status: Up
Host: 10.10.2.x ()  Status: Up
```

Using Python to call and process the data

If you look at the output provided from **Nmap** above, you can see that this data is perfectly **parseable** by Python, and Python's ability to process text is why it is so prevalent when working with security data.

There are multiple ways to process and retrieve the output, this example is a quick and dirty method:

```
# Quick example of how to run a command and get that command's output in python
import os
data = os.popen('/usr/bin/nmap -sP 10.10.0.0/24').readlines()
```

There are other, and much better, methods to call a program and retrieve the results, this is just one example. You could process the output directly from the command, or in the case of Nmap, you could open the text file that was created and process the results that way, which may be easier depending on how the results are formated.

Your task

In this lab, you will write a python script that takes in from the command line the IP addresses (or ranges) that you want to search, example:

```
python3 scanit.py 10.10.2.0/24
```

This script should conduct a search via Nmap for the available IP addresses and then walk through each of those IP addresses to gather additional information. At a minimum, your program will need to output the following information:

1. Number of hosts discovered, and the IP address of each host found
2. The Host OS information for each IP (get version or kernel numbers, if available)
3. Services provided and the associated version numbers for each IP

I strongly encourage you to use a small subset, or single IP address, in the beginning to test your code as this can take quite some time.

Getting Ahead

Although not required now, **wouldn't it be nice** to be able to store this information in such a way that we could check for differences easily in the future... When you write your program, we **strongly** encourage you to keep this in mind.

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 2: 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 scan of the VM network in an **easy to read format (a table!)**.

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 Nmap output taking out the individual fields (such as port#, tcp/udp, service, version, etc.) and working upon that information individually. The ability to easily parse information is a strength of Python and a very useful skill.