Austin Marino

Week 2 – Lab 2 Port Scans

CYB 339 – Cyber Operation Tools

Professor Morales

Contents

[Introduction 3](#_Toc536369672)

[Results and Analysis 3](#_Toc536369673)

[Servers and Scripts 3](#_Toc536369674)

[Apache 4](#_Toc536369675)

[Banner Grabbing 5](#_Toc536369676)

[Nmap Scan 6](#_Toc536369677)

[Conclusion 7](#_Toc536369678)

[References 8](#_Toc536369679)

# Introduction

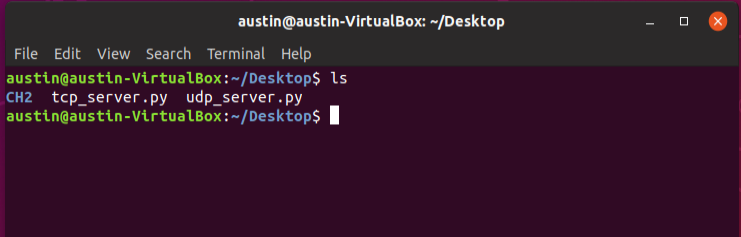
In this lab, the student is required to understand the fundamentals of port scanning and to also scan a given network to find vulnerabilities. Port scanning is an effective tool to utilize while penetration testing a network. While scanning ports, an attacker can easily assess whether or not a network has well protected security precautions implemented. However, if the victim being attacked does not maintain stable port management, an entire network can be compromised. Leaving key ports open to curious non-ethical hackers can lead to backdoor entrances and a headache of issues. During this lab, a small nmap port scanning script will be written to detect whether or not a specified port is open, closed, or being filtered for traffic.

# Results and Analysis

## Servers and Scripts

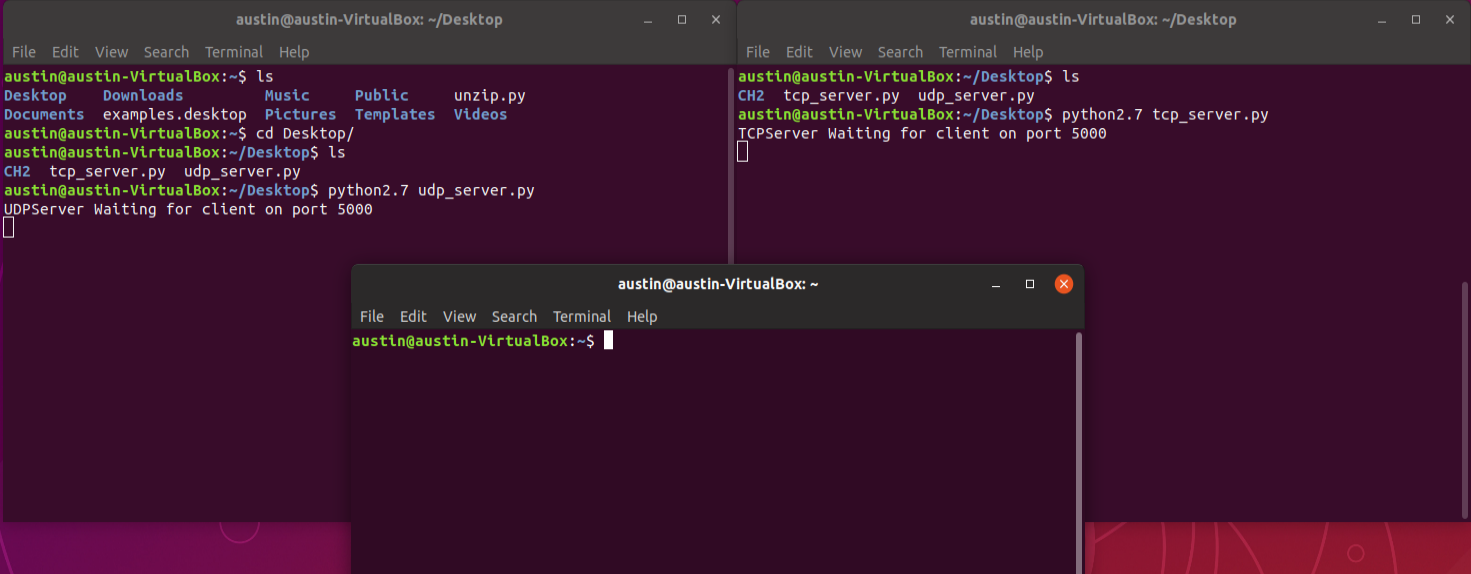
The screenshot below displays the documentation required for this lab. The directory *CH2* contains the Python scripts required to run the port scan and nmap scan. The other two python files are the TCP and UDP servers that are to be utilized during the scanning portion of the lab. These files can be found on the Utica.edu webpage under CYB-339 Week 2 “Reading & Resources”.

(Image 2.1)



As mentioned before, this lab will require the user to have three Ubuntu terminal (Command Line Interfaces) open. One terminal that contains the TCP server, the second containing the UDP server, and the third being the users current work space to run scripts. As shown below, make sure the user is in the current directory that contains the two servers and run the commands *python2.7 udp\_server.py* and *python2.7 tcp\_server.py*. These two terminals will be listening on port 5000.

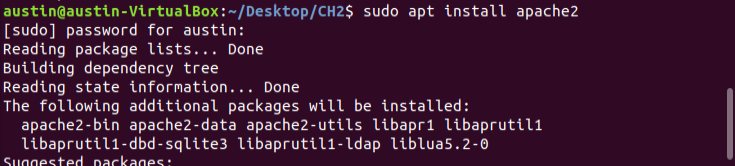
(Image 2.2) UDP Server – Left TCP Server - Right



## Apache

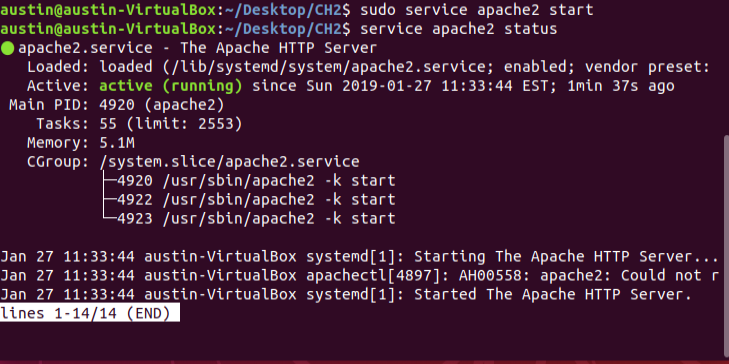
Another component to this lab is the installation and utilization of *apache2.* This is an open source webserver provided by Linux and Unix derived systems (Rouse). To ensure the user has apache2 installed and updated use the command *sudo apt install apache2* to receive the latest version. Reference image 2.3 for further assistance.

(Image 2.3)



After the installation of apache the user will need to bring the webserver online. Using the command *sudo service apache2 start* will bring the webserver online. This is what will allow the user to survey the internet to run port scans on a given IP address. Making sure the webserver is active and running, use *service apache2 status.* Running this command will prompt the user with information about the current webserver status as seen in image 2.4.

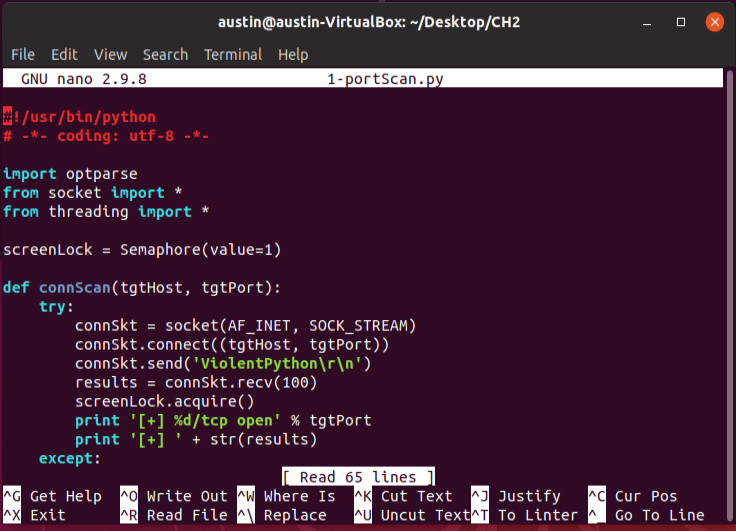
(Image 2.4)



## Banner Grabbing

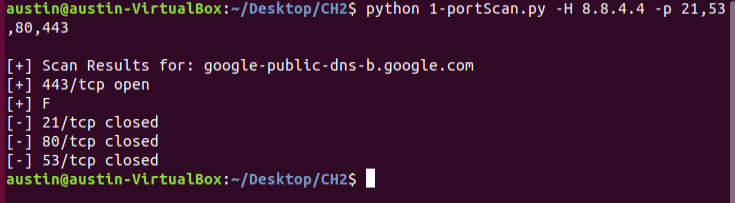
Banner Grabbing is a different form of port scanning that allows a penetration tester to gain information on possible system information. This information can be something like hardware, software, or network information about a target IP. In image 2.5, the code was provided from the textbook Violent Python (O’Connor Pg. 38).

(Image 2.5)



After the script has been created, the user will run this script with the command *python 1-portscan.py -H 8.8.4.4 -p 21,53,80,43*. This script will scan for information on the IP address 8.8.4.4 on ports 443, 21, 53, and 80. As a result on *google-public-dns-b.google.com* (Image 2.6) no vulnerable information was found. However the user can now see that port 443 is open, and the other three ports are closed.

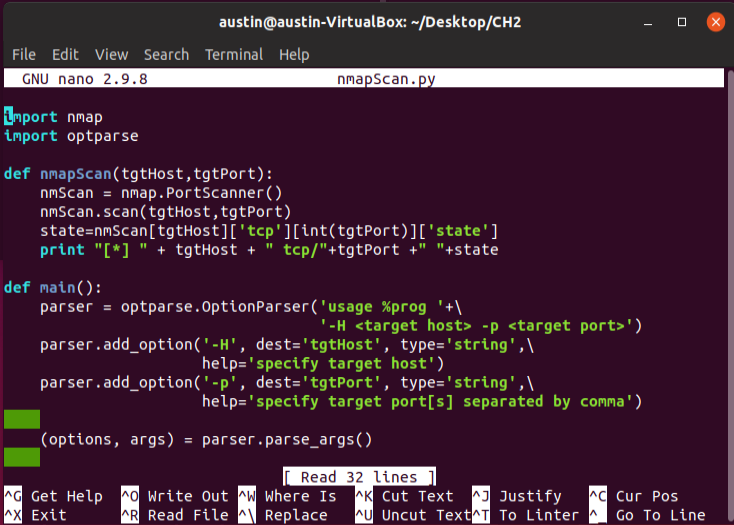
(Image 2.6)



## Nmap Scan

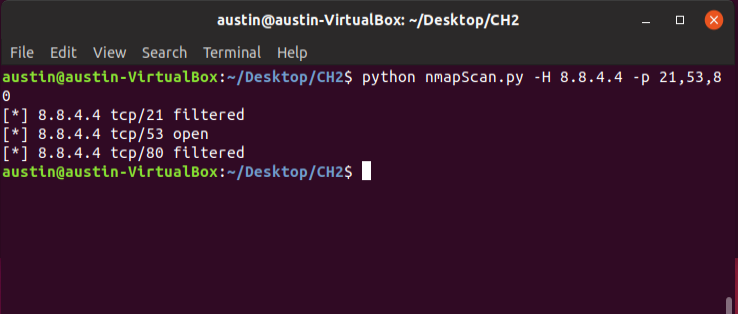
Using the Python nmap library works similar to the previous example. The nmap scan provides a bit more of information on how the traffic is handled. The image below displays a portion of the code utilized in the nmapScan.py script. This information is provided in the textbook Violent Python (O’Connor Pg. 40).

(Image 2.7)



To run the nmap port scan, use the command *python nmapScan.py -H 8.8.4.4 -p 21,53,80*. This will scan the status on port 21 FTP, 53 DNS, and 80 HTTP. On the IP address 8.8.4.4, FTP and HTTP currently have filtered traffic and DNS is closed (Image 2.8).

(Image 2.8)



# Conclusion

Port scanning in penetration testing is one of the most important steps to take while conducting the reconnaissance on a given target. This can lead to insight on open vulnerabilities that some networks may not be aware of. During this lab I had a few issues while running the scans. For some reason my python nmap file had not saved or may not have been configured properly from the previous lab. All I had to do was run a new *wget* command to re-install the library. Before I started troubleshooting this, I was getting an error stating that “Nmap could not be imported”. Another issue I was having was actually running the port scan and nmap scan scripts. When supplying the argument of an IP address, I used 10.50.60.125 which is an internal IP address. Not paying full attention and just following the text book, I realized I needed to supply an external IP address. After I figured this out it was smooth sailing and I had no issues. From a pen tester stand point, it still amazes me on how small some of these scripts are. These scripts are fairly simple and do not require much work, but they are extremely powerful.

# References

O’Connor, TJ “Violent Python, A Cookbook for Hackers, Forensic Analysts, Penetration Testers, and Security Engineers” 2013

Rouse, Margaret “Apache” September 2005, [*https://.whatis.techtarget.com*](https://.whatis.techtarget.com)