Chad Ballay

**Week 3 – Wireshark**

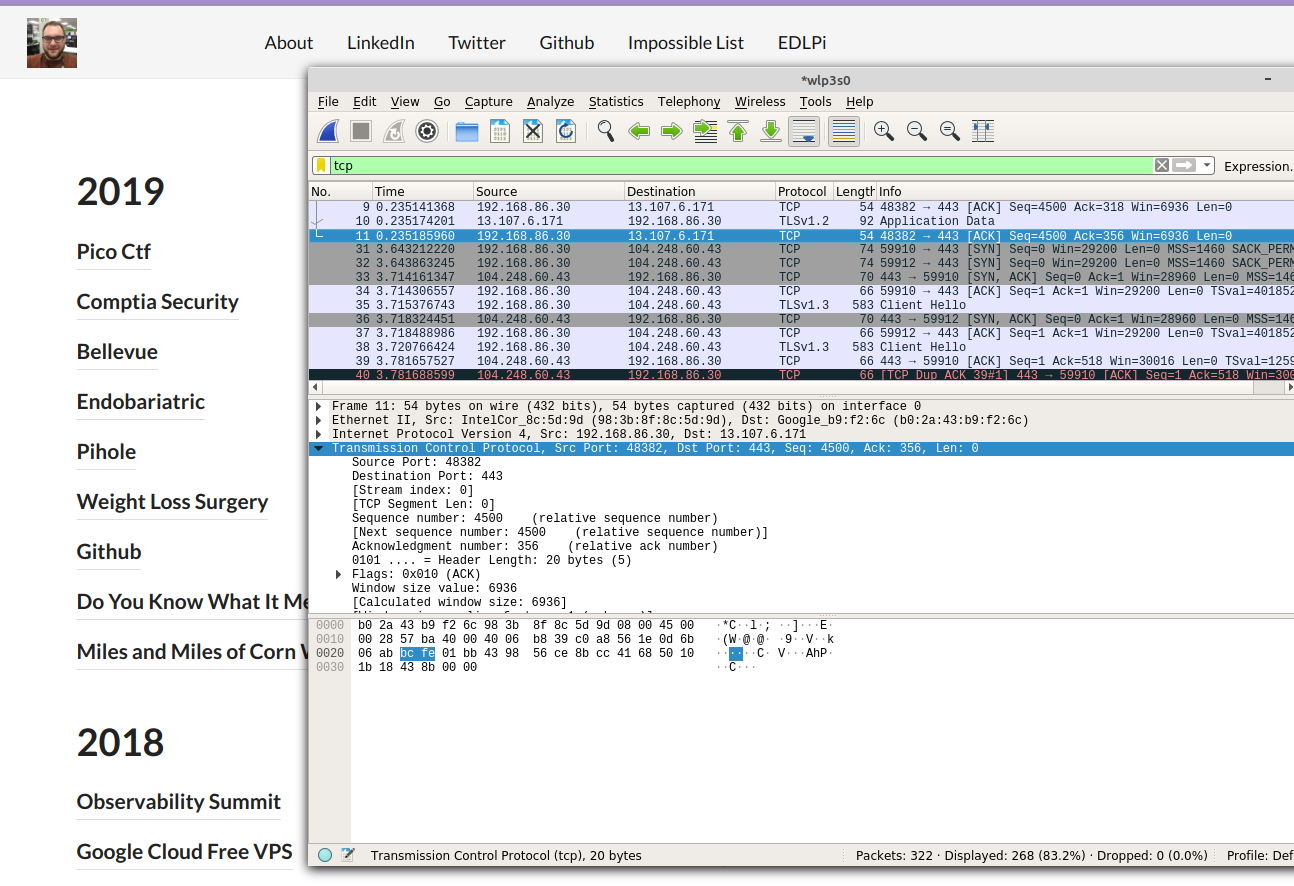
*On your personal computer, download, install, and run Wireshark (found at* [*http://www.wireshark.org/*](http://www.wireshark.org/)*). Wireshark is a network protocol analyzer that let's you capture and view network traffic. You may want to watch the videos found on that site first. With Wireshark running, go to a standard website (e.g., Google, Yahoo, Amazon, etc.) and report what it shows you. Do you see the standard TCP/IP handshake? Do you see your http traffic? How could a network administrator use Wireshark? Include screen captures of the Wireshark showing your results.*

I’m going to go a bit different here but follow along with me because I really think there is something to be said for this and I’m trying to pursue it as a side project. First part will be the assignment as prescribed. The second part is me sharing what I've found as a potential idea.

**Installation of Wireshark**

<https://www.wireshark.org/docs/wsug_html_chunked/ChapterBuildInstall.html>

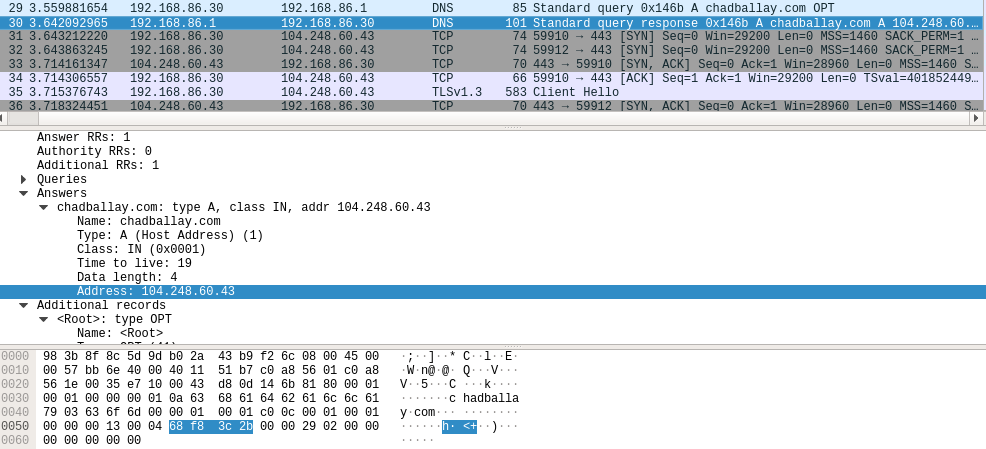
**Running Wireshark (**[**http://chadballay.com**](http://chadballay.com)**)**



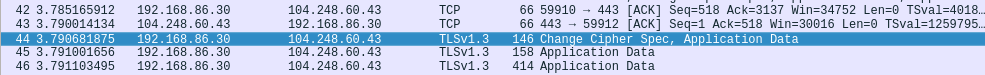
**SYN, SYN ACK, ACK**



**DNS Query**

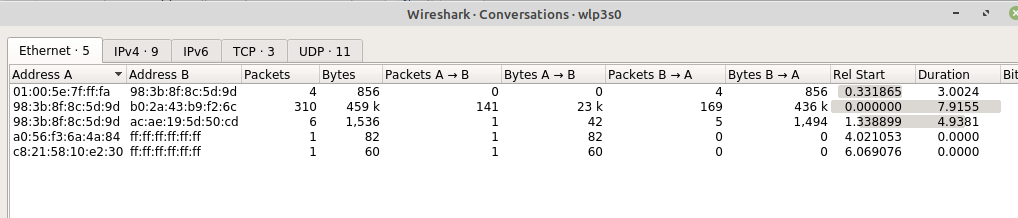


**SSL Handshake and Data**



**Network Administrator Usage**

Besides seeing the traffic I was expecting to, I also got to see how much other traffic was coming through on that interface. My printer, my Roku, my Google Home are all communicating it seems.



And as the Labtainer Wireshark-Intro down below shows, if the data isn’t encrypted then it’s fair game. (Telnet being the mechanism they use to show this.)

**Labtainer**

Now onto the Labtainer environment. This is a self-contained lab environment that contains several topic based sandboxes for getting hands on experience. It comes with 45 prebuilt scenarios and comes with a guide on building more. So the end user doesn’t have to worry about Vagrant/Docker/Terraform/VM configuration. Instead they run either a single VM or download the framework and it handles the configuration nightmare itself.

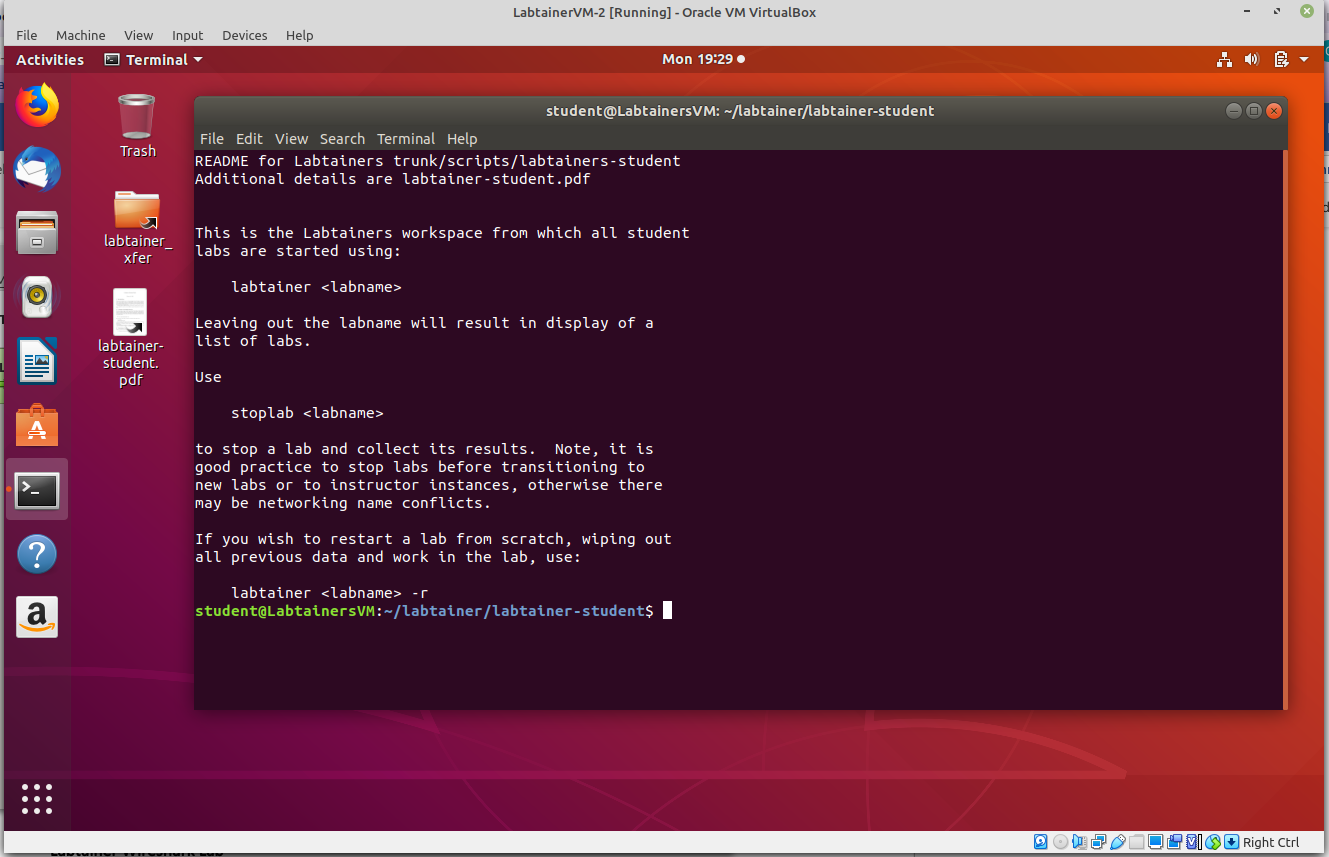
Project: <https://my.nps.edu/web/c3o/labtainers>

Demo Video: <https://youtu.be/JDV6jGF3Szw>

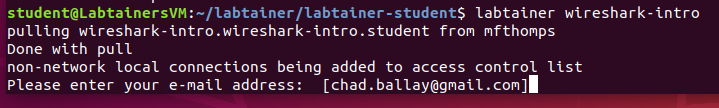
**Starting the Labtainer Wireshark Lab**

<https://github.com/mfthomps/Labtainers/tree/master/labs/wireshark-intro>

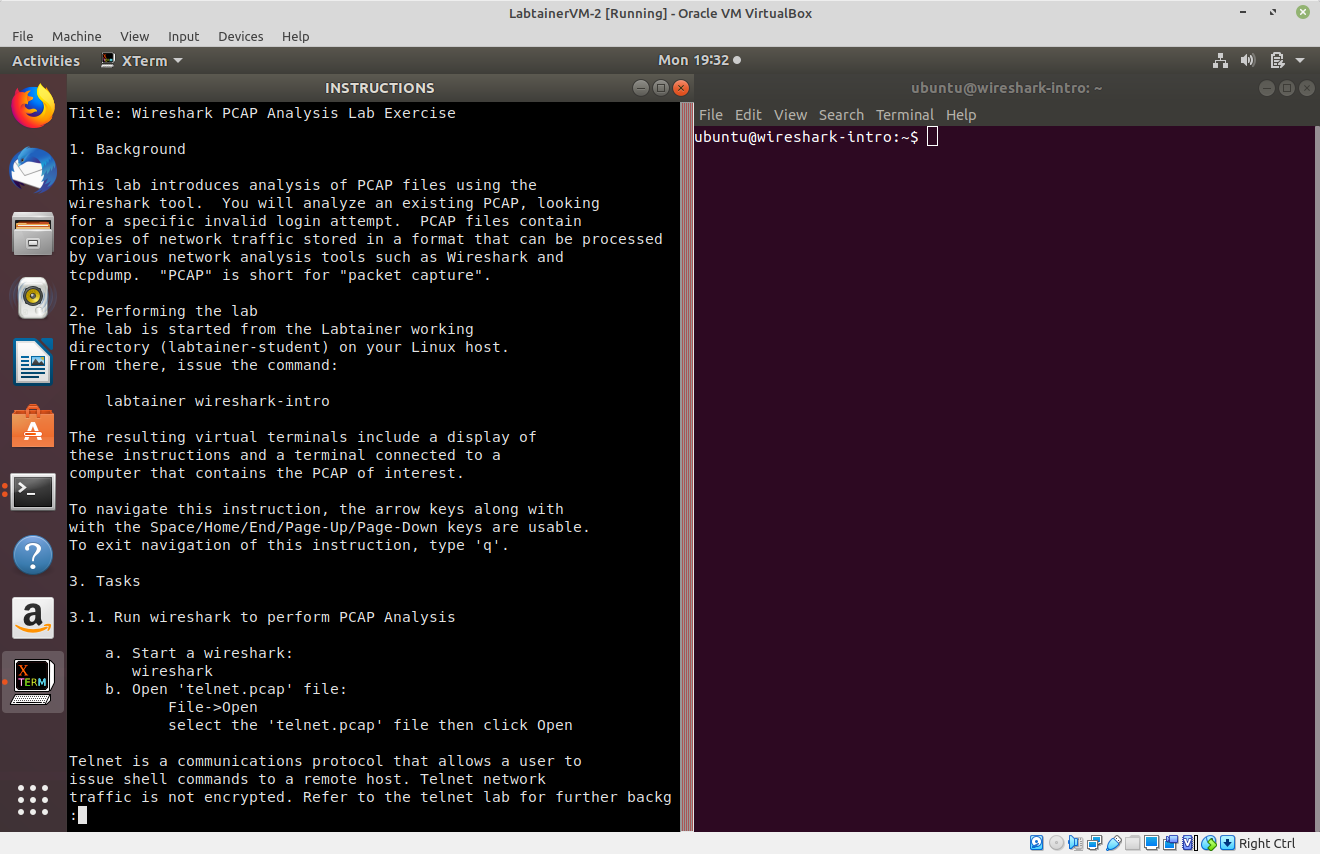
Here is the source for the Wireshark lab scenerio. Spinning up the Labtainer VM dumps you into this desktop.



To spin up the specific Wireshark-into one all the user has to do is run this.

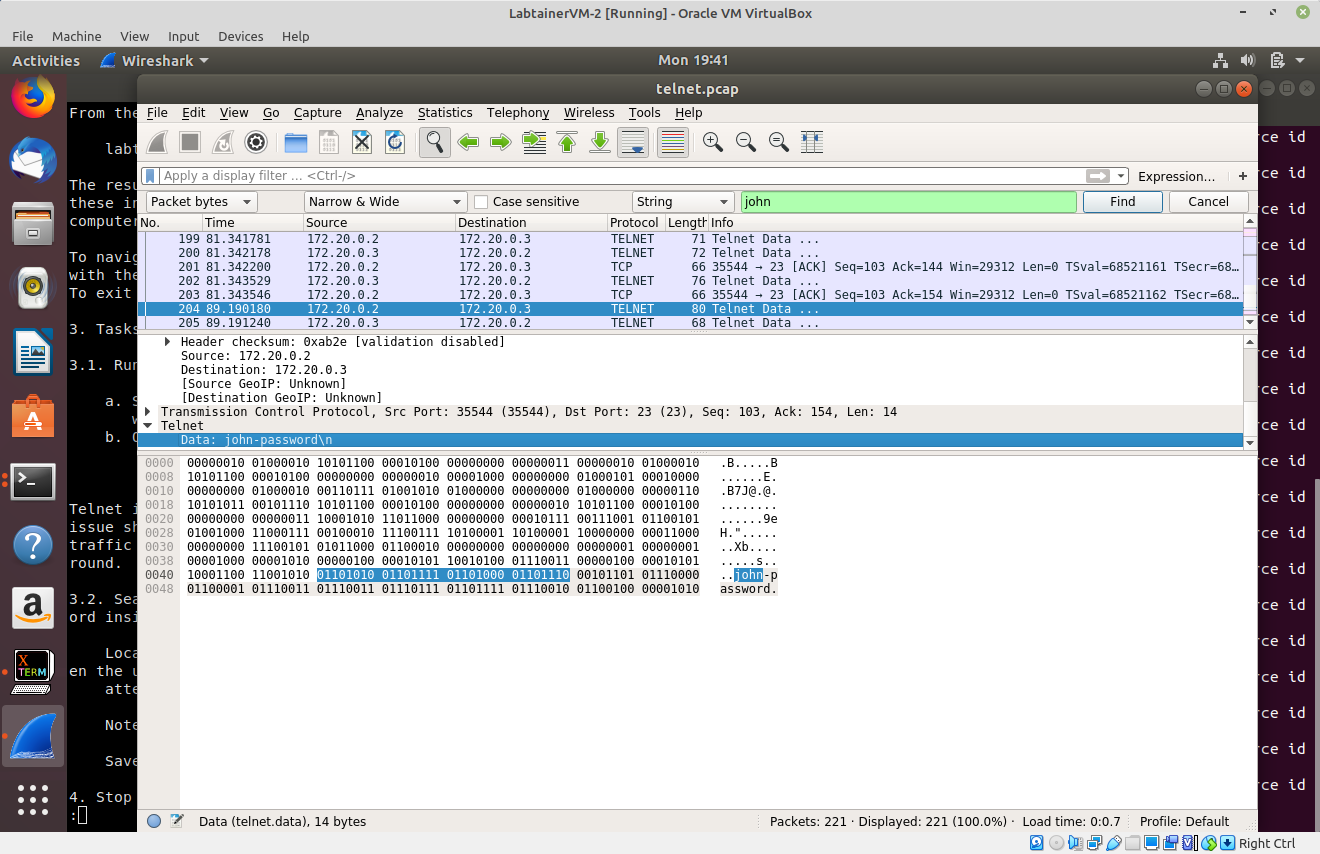


At which point it spawns two terminals. One with the instructions and one being a docker based image that has been setup to accomplish the lab goals.



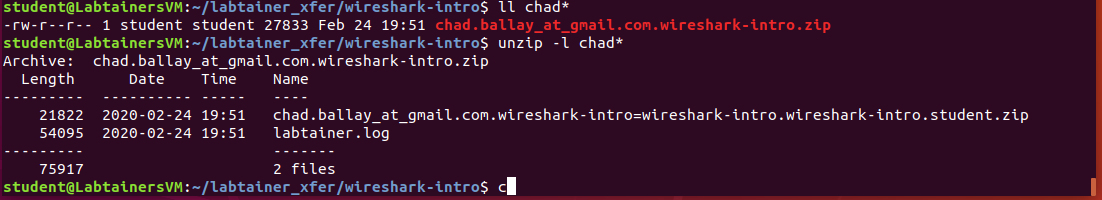
**Working the Wireshark Lab**

This lab ended up being a true intro to Wireshark. Start it up, search for a string, export the packet and you are done. Here is me doing so.



**Grading the Wireshark Lab**

At the end of the project, Labtainer creates a handy zip file with the results and various metadata around the student.



This zip file is then handed to the instructor to grade using the supplied scripts.

Instructor Manual: <https://my.nps.edu/documents/107523844/109121513/labtainer-instructor.pdf/595fe4b8-9858-48a7-a822-9e9374f1ddd9>

Reading the source for a couple of other labs, there is the ability to dynamically generate the goal so as to personalize the lab experience uniquely for each different student. (Mostly looks to be around hashing their email address into the string that is the answer.)

**Side Project Idea**

Besides meeting the stated goals of the project, I envision this being something I can use for other tests for general Linux skillset knowledge. I’ve got candidates that either flat out lie on their resume or are banking on network lag to allow them to lip synch the answers given by someone else in the room. The recruiters aren’t able to vet the applicants technical skills and HR isn’t budging on salary/job title offering to create a better pool of applicants. I can’t fly in every applicant that I have but the signal to noise ratio is killing my productivity.

What I’m trying to put together for an internal hack day project is spinning something like this up behind an Apache Guacamole instance or some other VDI situation. Generate a lab for the specific applicant and give them a timed window to work it. It won’t solve this problem completely but once that is in place I can at least have the applicants in Bangalore and other remote sites have to be physically present in the recruiters office with a form of ID.

A secondary goal is to establish internal training opportunities. Sitting someone down in front of a Udemy course is all well and good. But I’ve got help desk and other junior candidates that need practical work to understand the theory they are being trained on. If they are already in the system as an employee and are motivated enough to try these labs then I’ve got most of the staffing headache solved. They’ve demonstrated a fit for the company culture and something to work with.

**Creating Scenarios**

Thankfully this thing comes with a lot of prebuilt labs since the lab builder documentation is volumunous and unclear. The gist is that you can create full tech stacks that’ll be instatiated from scratch everytime the student spins up or resets the lab.

<https://my.nps.edu/documents/107523844/109121513/labdesigner.pdf/9ab93fc3-7cfe-48f4-987b-2efda8d99346>

**Possible Scenarios**

Use vi/find/grep/strings to pull a flag from a file buried hundreds of directories deep.

Use fuser to identify what process has a lock on a file.

Use lsof to identify what process has a lock on a port.

Find and identify what process is consuming the most CPU/memory/diskIO.

Get tomcat to start and read a string on webpage it serves.

Configure telegraf to write metrics to a influxDB endpoint.

(Pretty much I would troll through PicoCTF for various ideas and build up wards from there.)

*On your personal computer, download, install, and run Nmap (found at* [*http://nmap.org/*](http://nmap.org/)*). Nmap or "Network Mapper" is a free, open-source tool for network discovery, system security testing, port availability, and system monitoring. Before using Nmap, you should study the Reference Guide (*[*http://nmap.org/book/man.html*](http://nmap.org/book/man.html)*). If you are using a Windows operating system, you'll need to download ZenMap (*[*http://nmap.org/zenmap/*](http://nmap.org/zenmap/)*). For this exercise, run three different types of scans on another computer in your local network. MAKE SURE YOU HAVE THE SYSTEM OWNER'S PERMISSION FIRST! Provide screen shots of the command used and the results. What do the results tell you? How can a network security administrator use a tool like Nmap?*

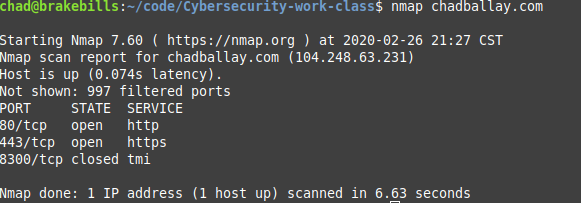
Same as before. I’ll first do the exercise and then relate it to using the Labtainer setup for learning.

**Installation Instructions**

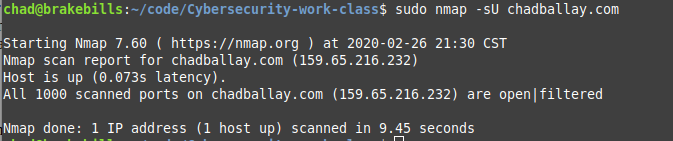
<https://nmap.org/book/install.html>

**Running nmap (**[**http://chadballay.com**](http://chadballay.com)**)**

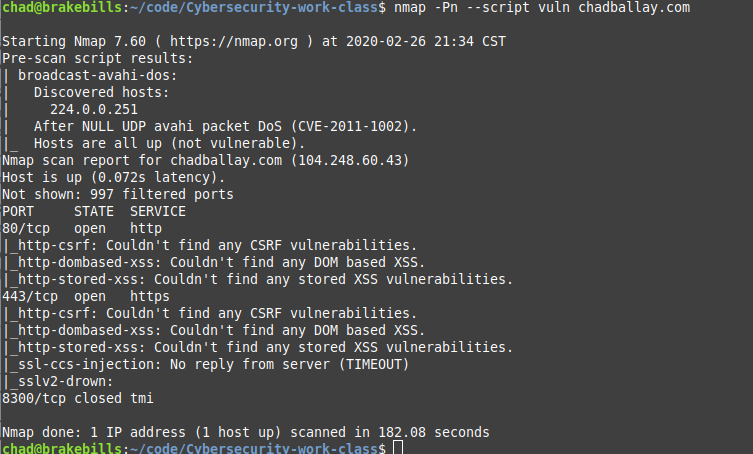
Basic usage will do TCP scan of 1000 common ports.



Basic usage but this time hitting UDP.



CVE detection scan



**Results**

My server is serving regular http as well as https traffic on TCP. Nothing on UDP. Plus, it looks like the hosting company is keeping the system patched.