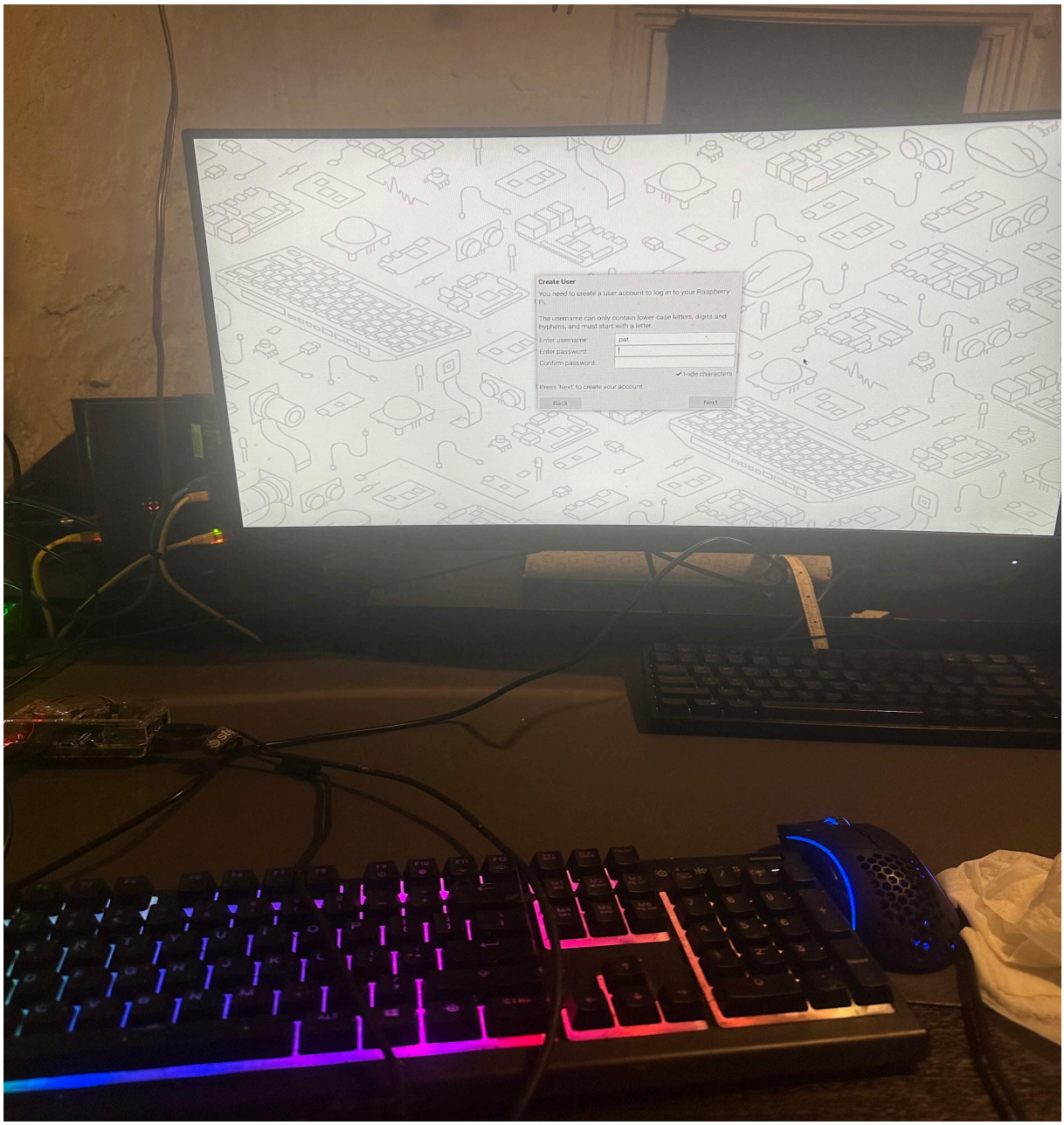
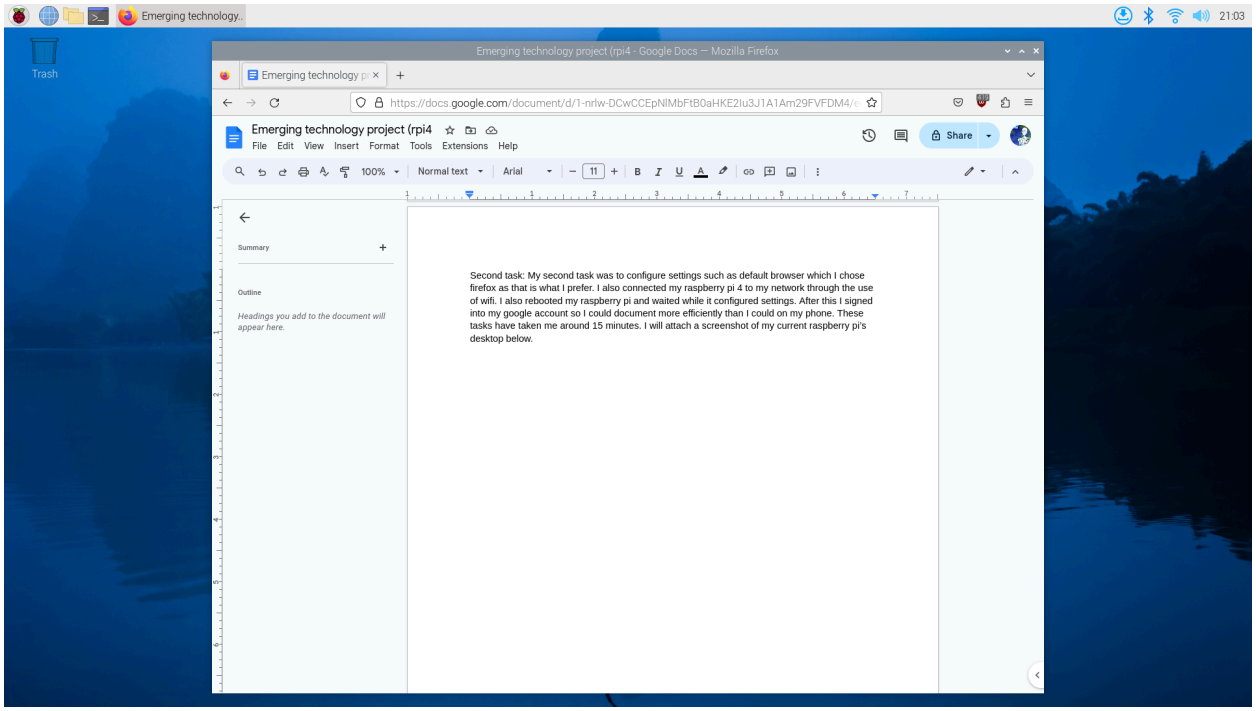
The first task I had to complete was booting the raspberry pi4 and configuring everything that needed to be configured. I booted my raspberry pi and connected all of my peripherals but for some reason my mechanical keyboard wouldn't allow me to type (rk84) so I decided to clean my old keyboard to which it allowed me to type. This took around 20 minutes to troubleshoot the problem and clean the keyboard. I will attach an image below:



Second task: My second task was to configure settings such as default browser which I chose firefox as that is what I prefer. I also connected my raspberry pi 4 to my network through the use of wifi. I also rebooted my raspberry pi and waited while it configured settings. After this I signed

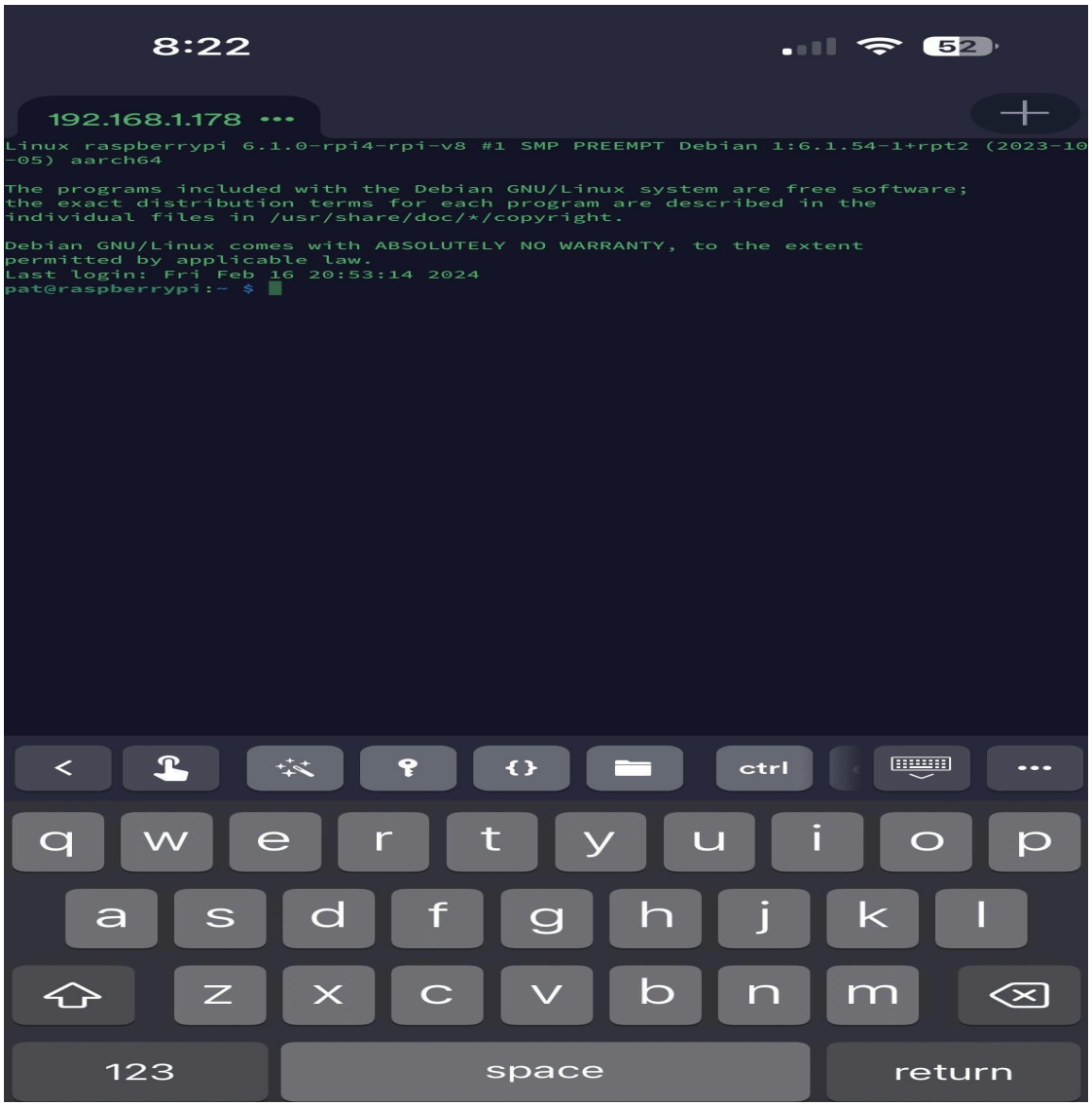
into my google account so I could document more efficiently than I could on my phone. These tasks have taken me around 15 minutes. I will attach a screenshot of my current raspberry pi's desktop below.



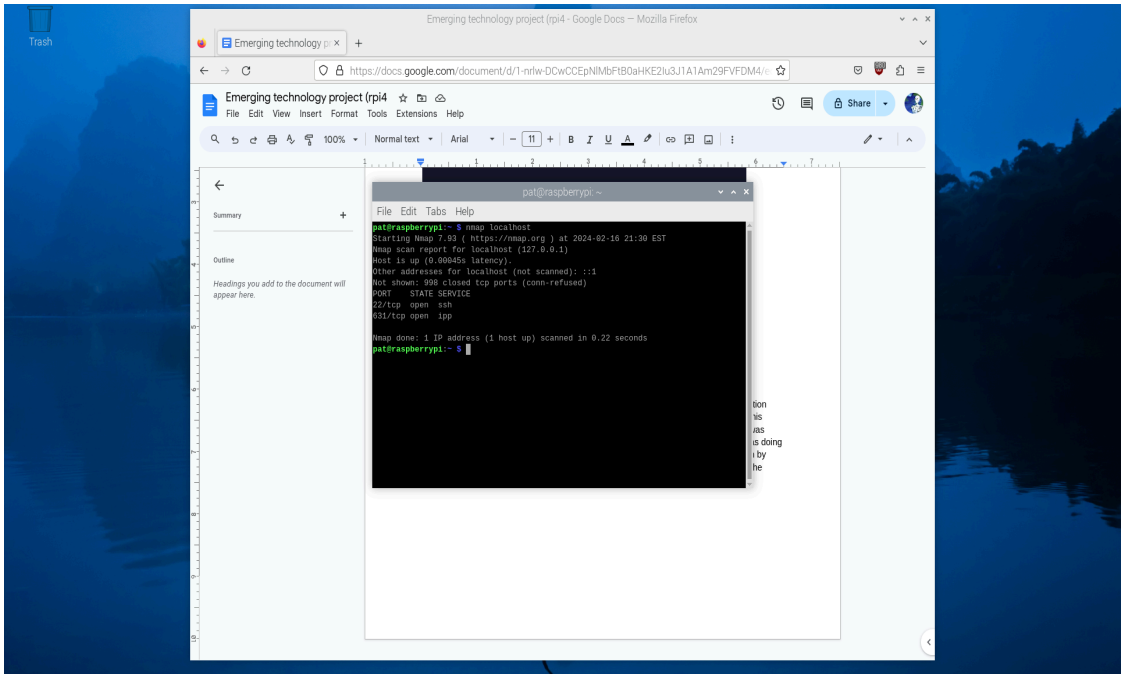
My next test will be making sure that I can connect to my rpi4 remotely as I will be controlling it through my phone using the SSH protocol for remote administration of the system. The things I will be needing to do are gathering my Ip address, and opening port 22 so that I can ssh into my RPI4.

For my third task which is configuring the SSH server I have had some issues getting the firewall to open the SSH port so that I can remote into my server. So I will be doing research on this issue. After doing some research on this issue I have found through https://all3dp.com/2/enable-ssh-raspberry-pi/#

I have found that I can use SSH using my phone's terminal client. I will attach a screenshot below.



Researching, troubleshooting, and getting the SSH service running and allowing connection from other clients on my LAN took around 40 minutes. The tools I used to troubleshoot this problem was Nmap to check for an open SSH port. I used Nmap to scan my localhost to confirm that the SSH service was running and I found out that it wasn't. This is what took me to my step of action which was doing research on why this problem occuring. I found that it is because you need to enable ssh by going through to preferences > raspberry pi configuration > Interfaces. After clicking on the interfaces tab you will then need to click on the ssh button to enable the ssh connection. I used nmap to make sure that the port was open and I will attach a screenshot showcasing the results.



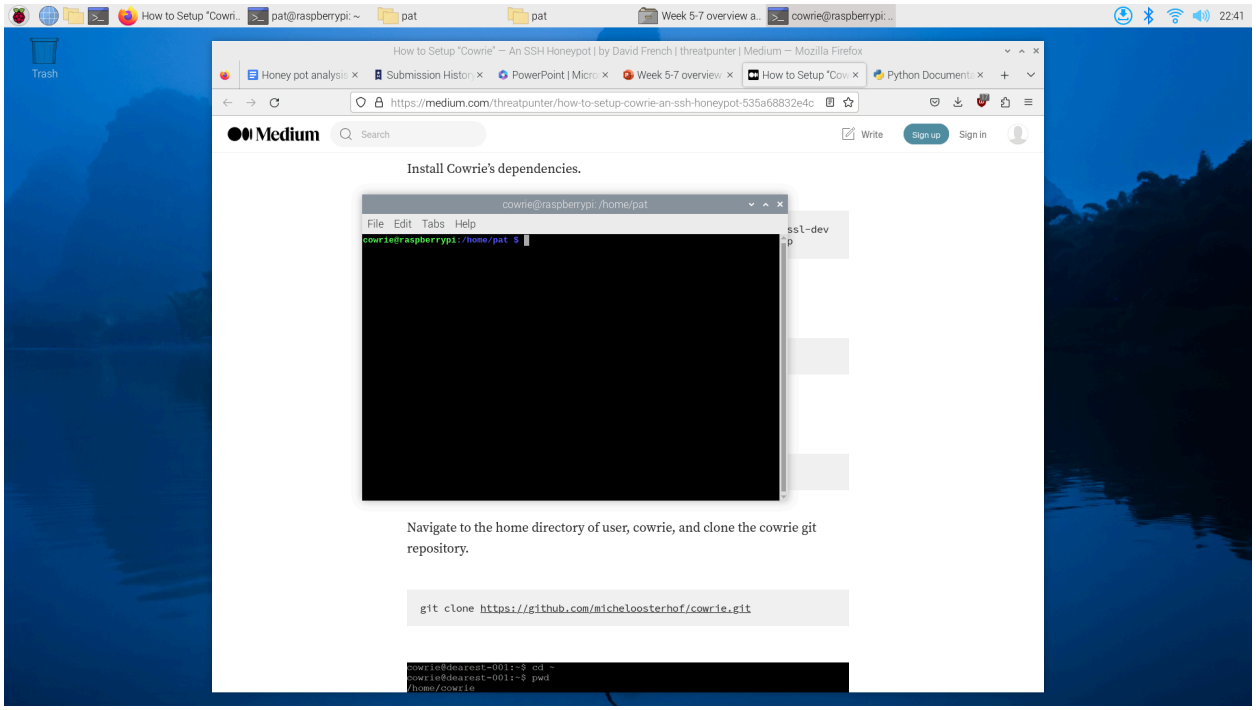
It is hard to see but you can see that port 22 is open to allow ssh connections alongside Ipp. The documentation took around 15 minutes to think and type out.`

My next step is to implement the honeypot to which I will be following a tutorial to setup the first honeypot. At the end of this documentation for week 6 I will add a sources cited page for resources that I have used throughout this. I have created my first test video using OBS software. I wanted to include audio for my screen recordings. I have tried installing different audio drivers and using different headphones but for some reason audio isn't coming in or out. I also tried plugging my micro hdmi into a different micro hdmi port which was unsuccessful. This troubleshooting process took me around 15 minutes. I used this forum for troubleshooting: https://forums.raspberrypi.com/viewtopic.php?t=260936

So for the videos I will be documenting while I record to show what I'm thinking.

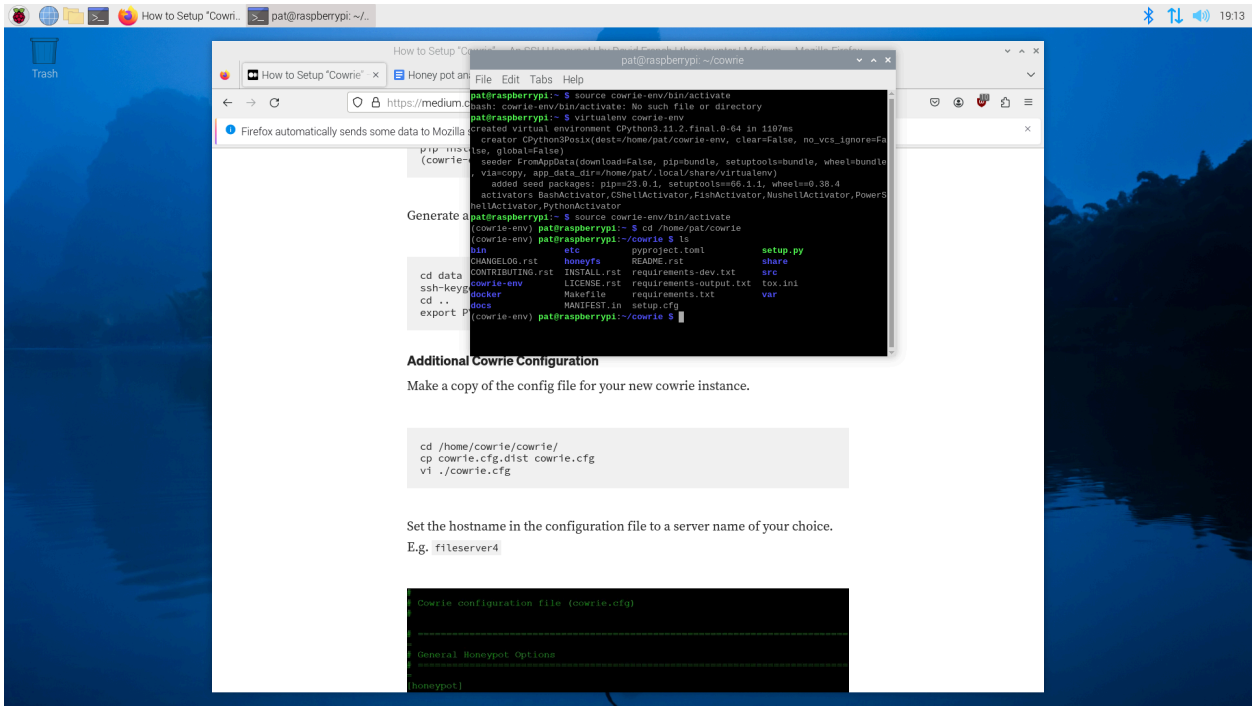
I will be creating the first honeypot using the resource within my powerpoint file

For the second video I went through some of the process of setting up the cowrie honeypot. I had some troubles getting the dependencies installed so I will be troubleshooting that to help set this up.(The two videos ended up getting corrupted on my rpi4)



I switched to the user but realized I couldn't install software packages under this user as it wasn't in my sudoers file. I switched back to my regular user account and sudo apt installed all of the dependencies for this honeypot. Doing this took around 10 minutes due to the speed of my internet.

In the middle of the honeypot setup I realized that it wasn't properly setting up. So far in the current troubleshooting process I have realized it is the current packages not being installed. I'm going through the error message currently to see what the problem is. After coming back to it I realized that the problem was due to me not changing to the proper directory.

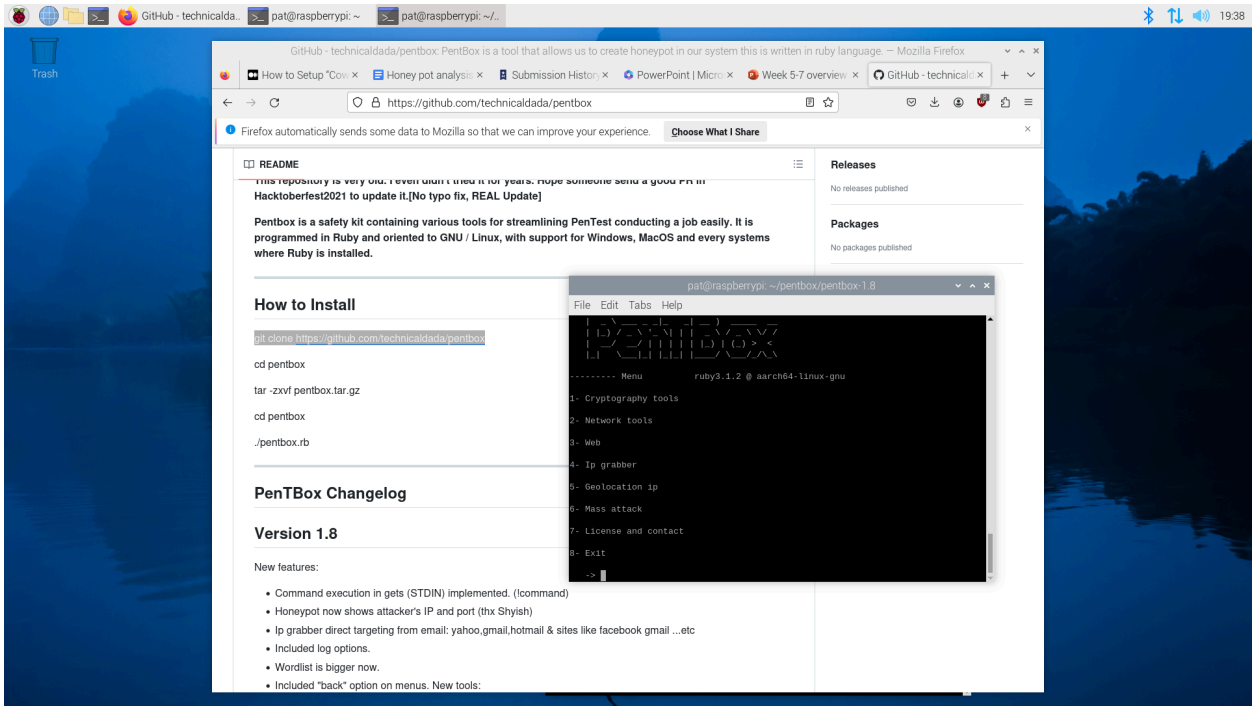


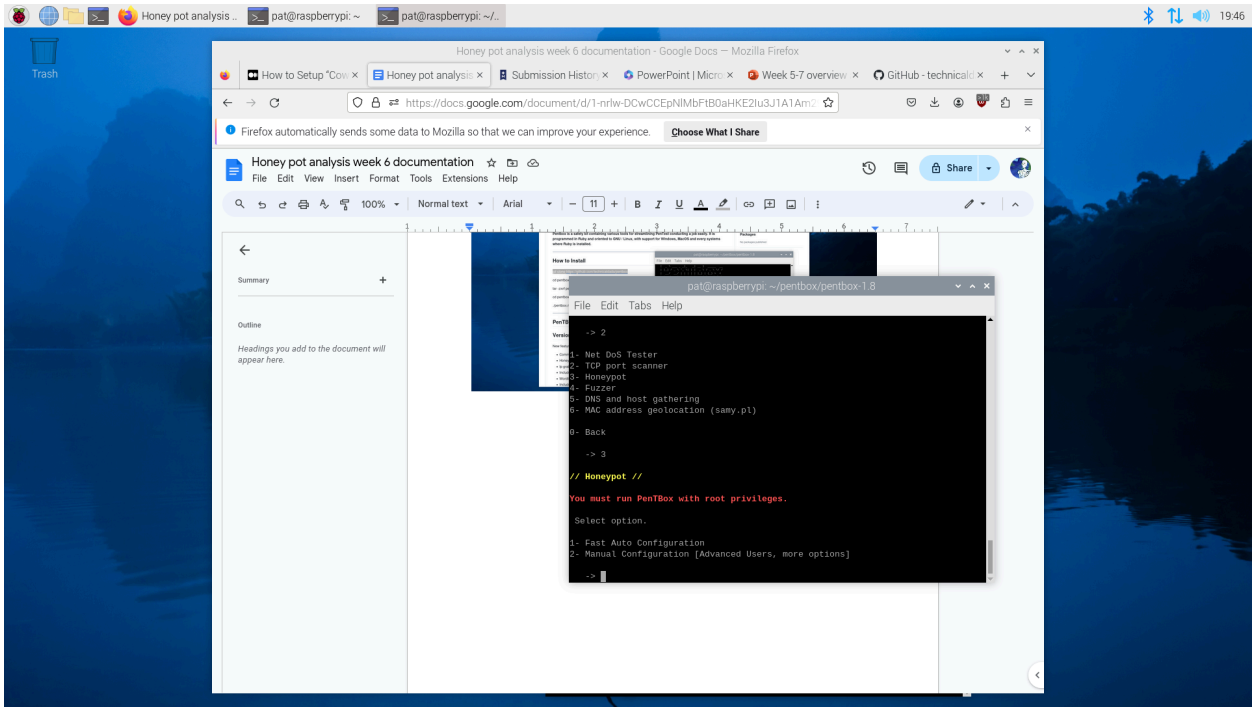
After setting up this honeypot I realized that there are better honeypots available so I will be looking for a better honeypot. I have also decided that instead of using the honeypot I was going to use, which was Kako, I'm going to use a FOSS tool called pentbox to help set up a honeypot. The resource I'm going to use is:

How To Configure Honeypot with PentBox in Kali Linux - 2020

To start the process of installing this I typed in github pentbox and came across the url:https://github.com/technicaldada/pentbox After coming across the url I followed the installation instructions by first opening my terminal and git cloning the github url.

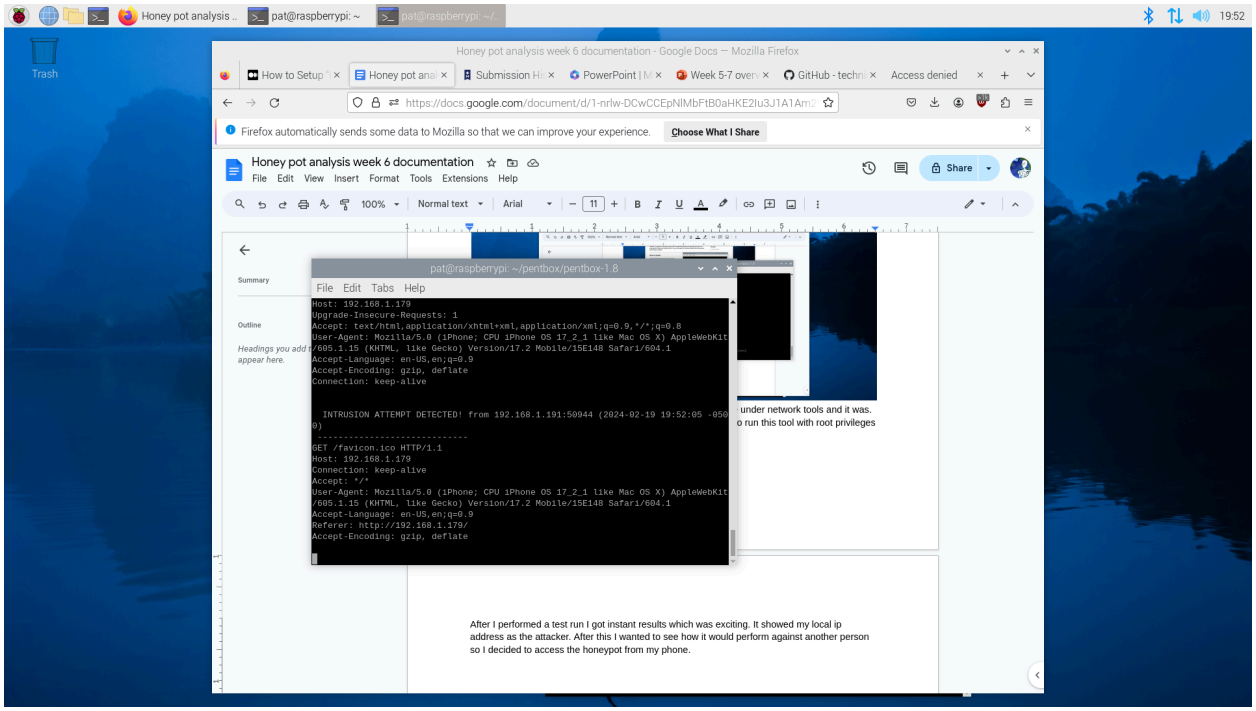
I troubleshooted and learned that I needed to install a couple of dependencies as well as needing to install the ruby language to my raspberry pi so that the honeypot file could be executed. It took around 45 minutes to set up the pentbox offensive security tool. I will attach a screenshot below showing this hacking/honey pot tool. I will also be installing kali linux on my desktop computer to act as an insider threat actor attempting to exploit vulnerabilities within the honeypot server. The documentation for this task took around 5 minutes.





I entered option 2 to find the honeypot as I assumed it would be under network tools and it was. After that I entered 3 for the honeypot and found that I needed to run this tool with root privileges to work properly.

After I performed a test run I got instant results which was exciting. It showed my local ip address as the attacker. After this I wanted to see how it would perform against another person so I decided to access the honeypot from my phone. This simple test shows the ip addresses and even the type of device I was using, which was my Iphone. I will attach the picture below to show the results of the testing phase of this. Performing the setup and analysis of the results took 35 minutes



This honeypot from what I have learned is very responsive with the detection of intrusion. It captured the IP addresses of my localhost (rpi4) and my Iphone.

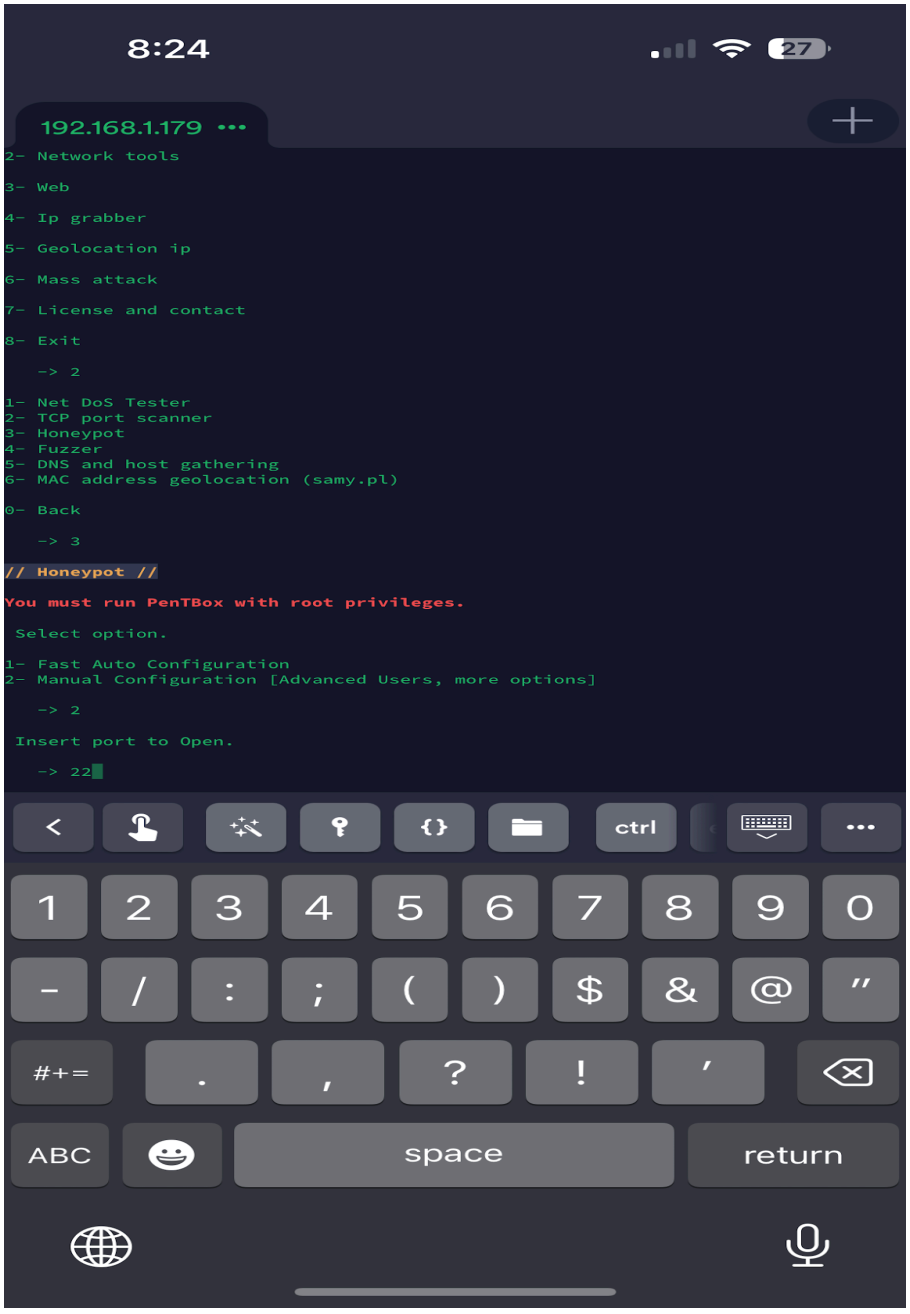
I want to keep this project interesting so I will be setting up three kali linux virtual machines (Cloned from a single template) instead of one to simulate how a honeypot would have multiple threat actors attempting to compromise the honeypot. Analyzing and documenting the results took around 10 minutes

My next task I will be installing kali linux and creating two clones of the original kali linux vm to interact with the Honeypot. The first step is to install the virtualbox iso image file for kali linux.

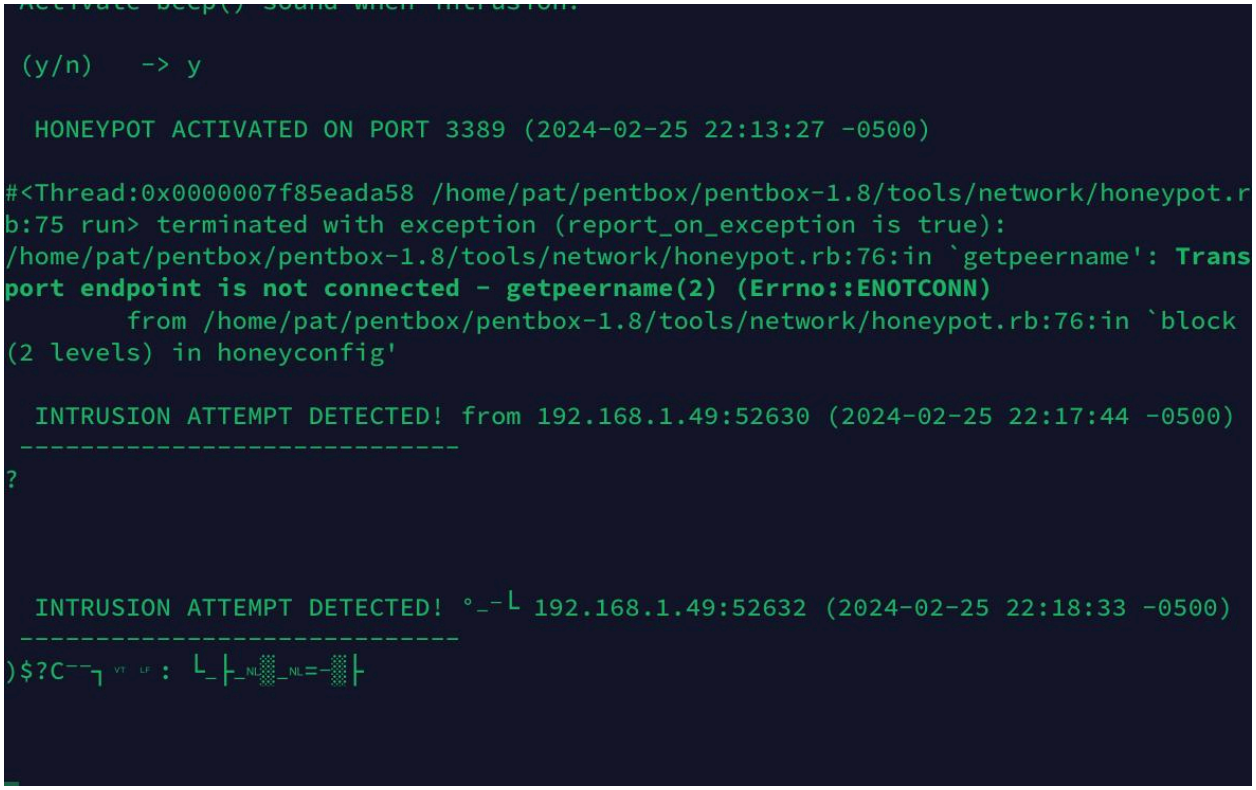
The iso image came as a .7z file. After installing the file I extracted the .7z file into my emerging technologies folder within the D:\ directory. The next step for me was to install the first version of kali linux.

Installation of the .iso image and extraction took 15 minutes After installing Kali 1 I made two full clones based off the image named Kali 2 & Kali 3. The installation and configuration process took around 40 minutes to complete and cloning the virtual machines took around 30 minutes to complete due to them being full instead of linked clones.

My next step was to reset the honeypot up. I decided to try the advanced configuration which allows me to set up a custom port. I chose port 22 for ssh. I will be attaching a screenshot below. I once again am using the terminus client application that allows me to ssh into my raspberry pi 4. Setting up the honeypot took around



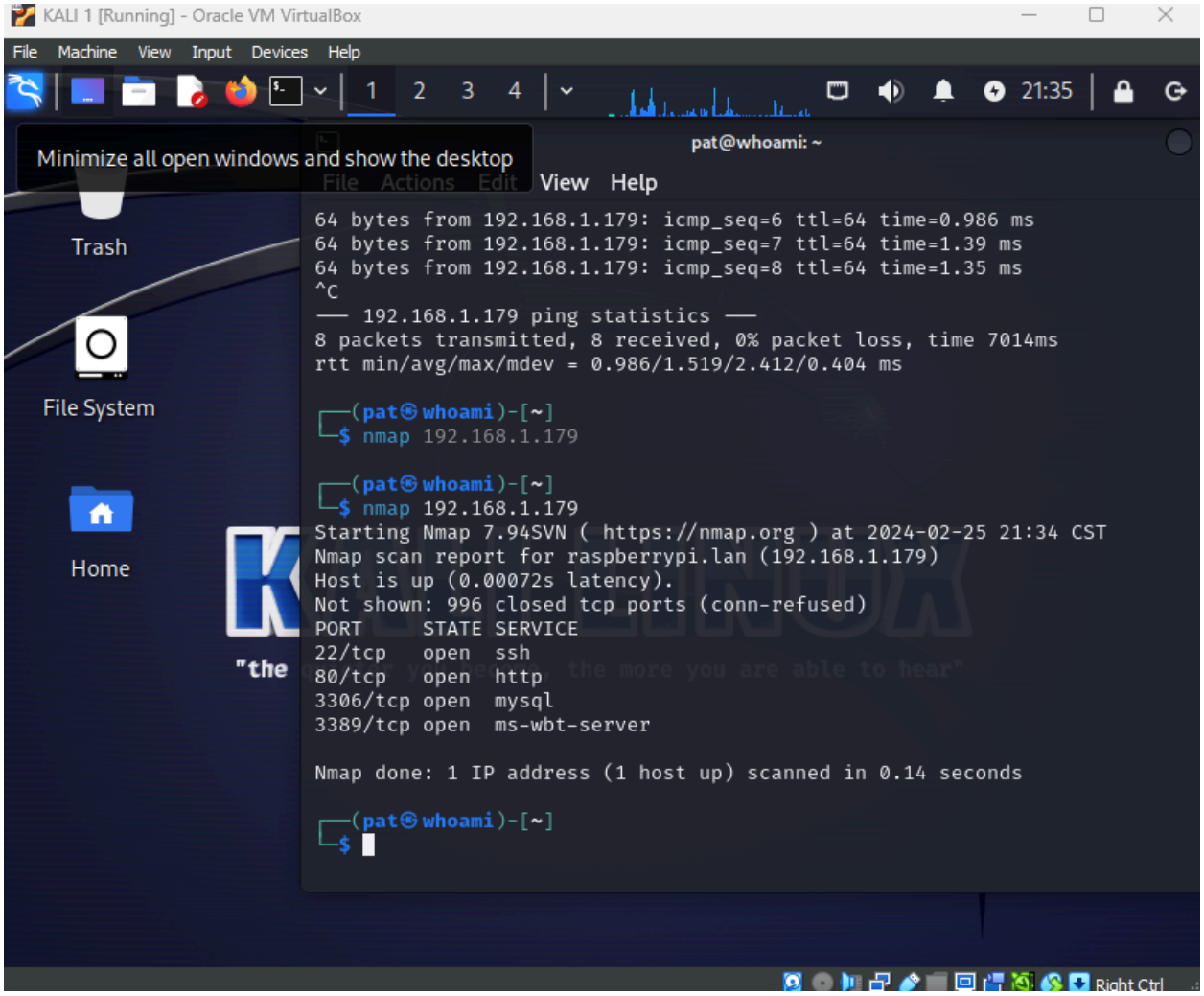
This screenshot is me setting up an SSH honeypot which I will be attempting to brute force on all three of my kali linux VMs. This took around 10 minutes to set up but wouldn't allow me to use it as the port was already in use.



I set up a RDP honeypot instead of the SSH honeypot due to the fact that the SSH port was already in use which stopped me from being able to connect. I will be sending the video showing the process and research I went through.
This process took me around 25 minutes (including the 14 minutes of video) to set up/test and analyze the results. In the video I initially attempted to trigger it through my attack VMs. After that didn't work I used my workstation to attempt a RDP connection and that set it off.

My next task is to set up another honeypot using a different service. For this I will be setting up a honeypot that uses port 3306 which is the port for mysql. Databases are a common thing for attackers to attack which is why I thought it would be a good idea to create a honeypot with this specific port. I will start by first initiating a SSH connection to my RPI4 with my phone. In a real life scenario this may tempt threat actors to use tools such as sqlmap.

My first step was to make sure that the 3306 port was open on the RPI4 which it was. I used Nmap to perform a scan which detected and confirmed that port 3306(mysql) was open. I will attach a screenshot below.



My next step is to perform research on how to trigger this honeypot. I will need to find a way to attempt intrusion against this service.

I will be using this source to help assist in the pentesting of the fake mysql server.
https://book.hacktricks.xyz/network-services-pentesting/pentesting-mysql

After 35 minutes of attempting to use sqlmap it did not trigger any alarms of intrusion detection within the honeypot.

Reflection of honey pot project:

For the most part my tasks aligned with what I planned with the interaction of the honey pot and being on time. My project took 6 hours and 5 minutes which is around how long I planned for my project to be. Some of the things that did not go according to plan are using Kako. I scrapped kako due to the fact that pentbox was a much more smooth and easy to use tool for setting up honeypots. I also set up and configured Cowrie as an SSH honeypot but I didn't get as good of results compared to the pentbox honeypots I configured. I also attempted to create an ssh honeypot using the pentbox tool which didn't work due to the fact that ssh(port 22) was already in use. The last thing that didn't work was getting the intrusion detection of the honeypot to be set off when attempting the pentest against the mysql honeypot service configured on my RPI4. Some of the lessons I learned throughout this project were to remain patient and always keep trying. This project was met with a lot of problems/obstacles but staying patient and using troubleshooting skills got me through it. I have used honeypots before in my Intrusion detection and prevention systems fundamentals class, but I wanted to learn more deeply about how to set up honeypots and interact with them which is why I chose to use honeypots as my project.

Skills I learned:

- How to setup a raspberry Pi 4
- Configuring a SSH honeypot using Cowrie on my Raspberry Pi
- Configuring a honeypot on pentbox with the ports 80(HTTP), 3306(mysql), and 3389(RDP) on my Raspberry Pi
- Using offensive security to pentest active honeypot services running on my raspberry pi
- How to use incident response/intrusion detection to analyze threat actors and their attack patterns(Which in this case I was the threat actor)
- Using OSINT tools Cowrie, Nmap, sqlmap, and Pentbox to perform offensive security tasks