Closing report:

The last couple weeks I have been working on figuring out the specifics of linux commands and what I can and can't through ssh commands. Honestly after draining a bunch of hours into this the end product though, could be improved I believe it is incredibly simple and almost fully function for modifying to a IoT device which was my initial goal.

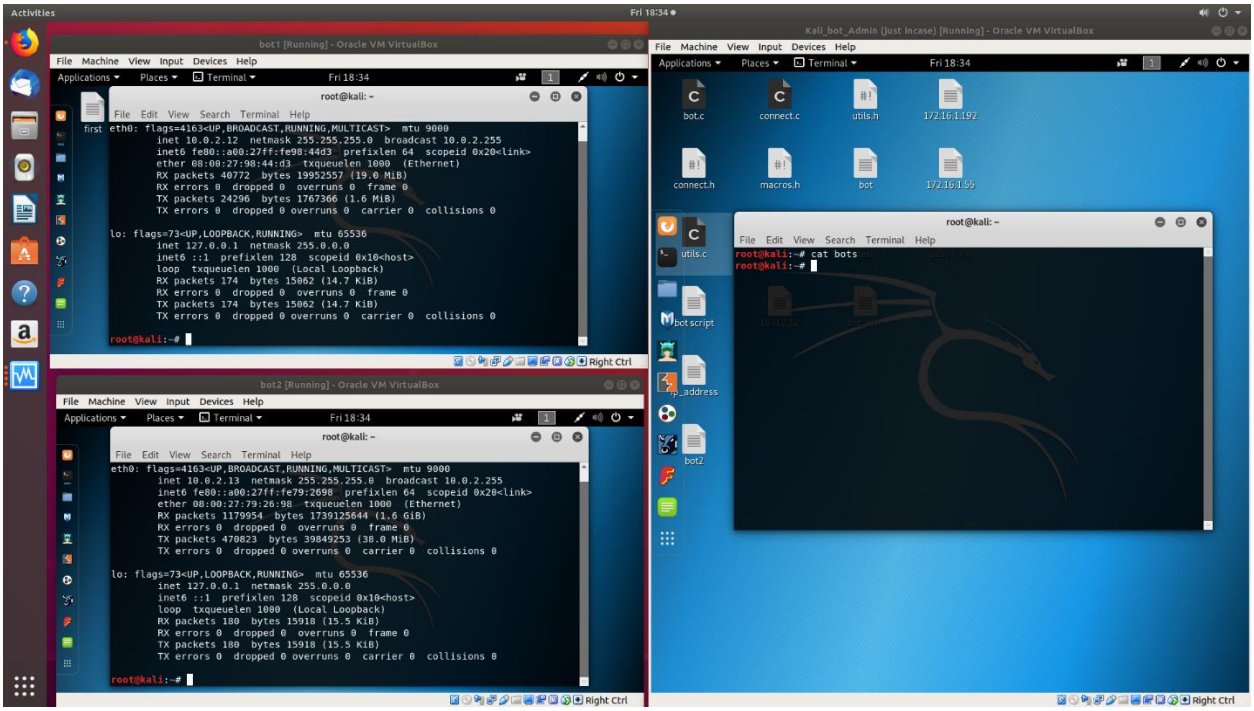
I didn't end up doing this on an IoT device for a couple reasons. First plain and simple I was having a heck of a time getting my hands on one and as soon as I found a router with a Linux operating I found out that I was at a complete loss on how to get started. That's when I figured I should go ahead try and get this done locally hosted between multiple virtual machines. This proved to be a lot of research and proved to me how uncomfortable I was with bash commands. I then switched over to trying to make a fully operation bot net this way. Once created it should be able to be easily modified to work on an IoT device.

The first thing I want to do is walk through on my final project and discuss what I did. Then I want to follow up on what I want to do moving forward and the problems I foresee. Eventually I would then like to use this in a real situation restricted on a closed home network.

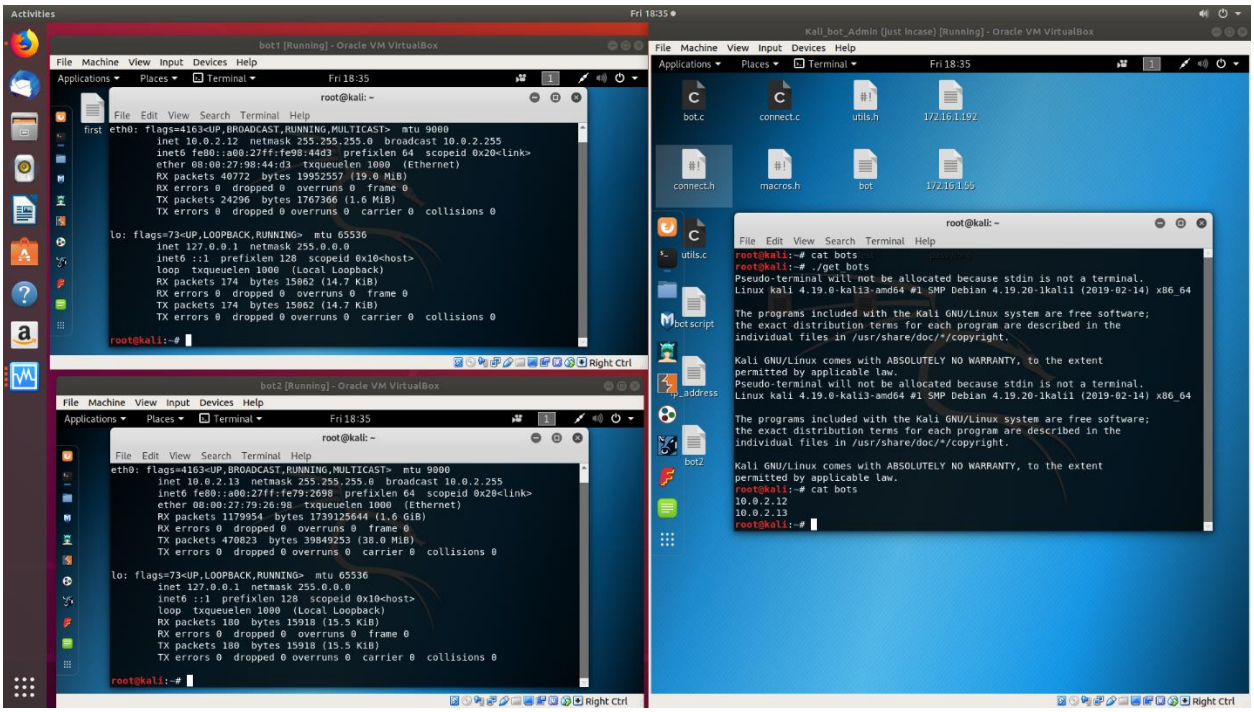
Final Product:

This is just going to be a brief walkthrough via screen shot on running the final product.

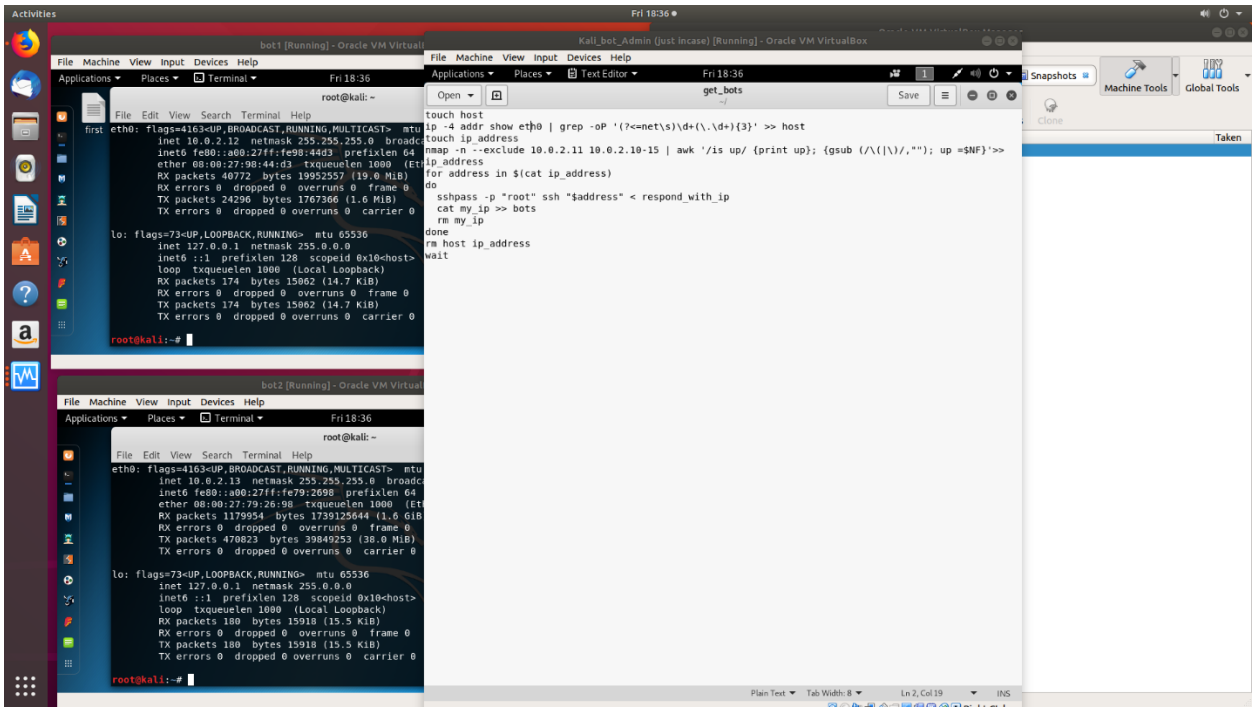
1) This is showing the IP addresses of both VMs as well as the showing that the file that saves the bots IPs on the command machine.



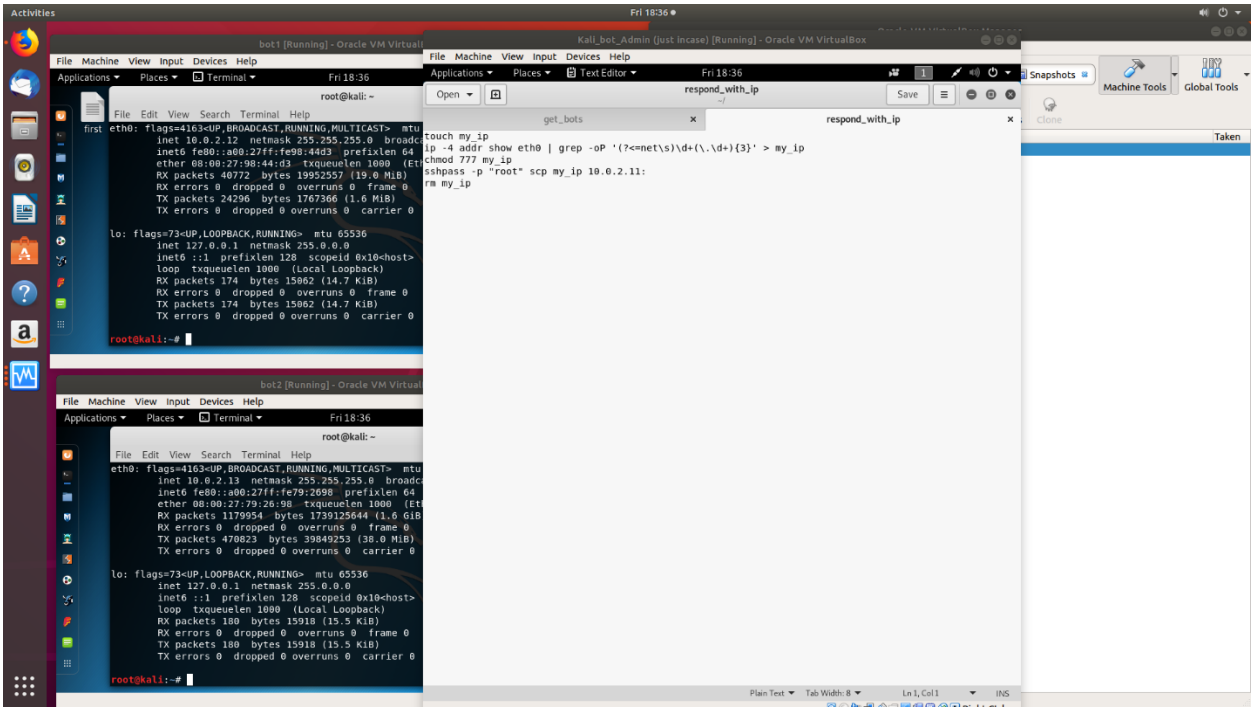
2) This screen shot shows me running the get_bots command. This command has been designed as a do all in unison with the respond_with_IP script. Running the command will run nmap on the command machine and log all of the available ip addresses then the machine will attempt to log onto them using the root password. If successful the respond_with_ip command will run and the attacked machine will return and log the ip to the bots file on the command mach



3) This is a picture of the get_bots command. Pretty much this is the automated process of doing everything from using nmap to issuing the other script file on the attacked machine. First we create a file for holding the host ip. I found this easier to run the nmap exclude with. Next we have the nmap command that finds the potential bot machines IPs. Then we ssh into the potential machines and run the script that will log and return it. Lastly we clean up the spare files.



4) This is the script that returns the IPs. This script simple pulls the systems IP and sends it back to the command machine. Once done it simple cleans up after itself to not leave a trace.



Problems with this design: There are a few issues with this design that I would like to address before closing out. First there are a few different tools I used to help make this easier for me. If the potential systems were linux based this wouldn't be an issue. The tools I used are nmap, and sshpass. Still pretty simple and shouldn't cause an issue if the machine has full linux support. Next I could not figure out a great way of overcoming the accept connection message when you ssh for the first time. I'll need to

looks more into this moving forward. Lastly I would like to experiment with making this hop connections more. Through this implementation you can get full control to the other machines which means you definitely can but I didn't make an implementation that was meaningful enough to post it on here.

Conclusion: This project was really great for my education. After plenty of hours of work it ended up being so simple but I was just lacking so many of the base concepts of what I was doing I had to spend a ton of time stumbling around in the dark to work through that. Something I would like to focus on is the concept of a command server. If you actually implemented this you would want to infiltrate another machine (or a couple) and use it as your command and control center. This helps keep you anonymous. I'm confident with a decent amount more of work I could turn this into the real thing which is a really great take away.

I'd also like to follow up on what would be the next step here which would be using the bot net for something such a denial of service. With this implementation I proved this would be possible by sending back files from the bots. That being said I tried a denial of service but plain and simple I lacked the fire power with my two bots on my local network.