Lab #8: Encryption
CSC432-A

Franklin Nuth

19 March 2019

Abstract

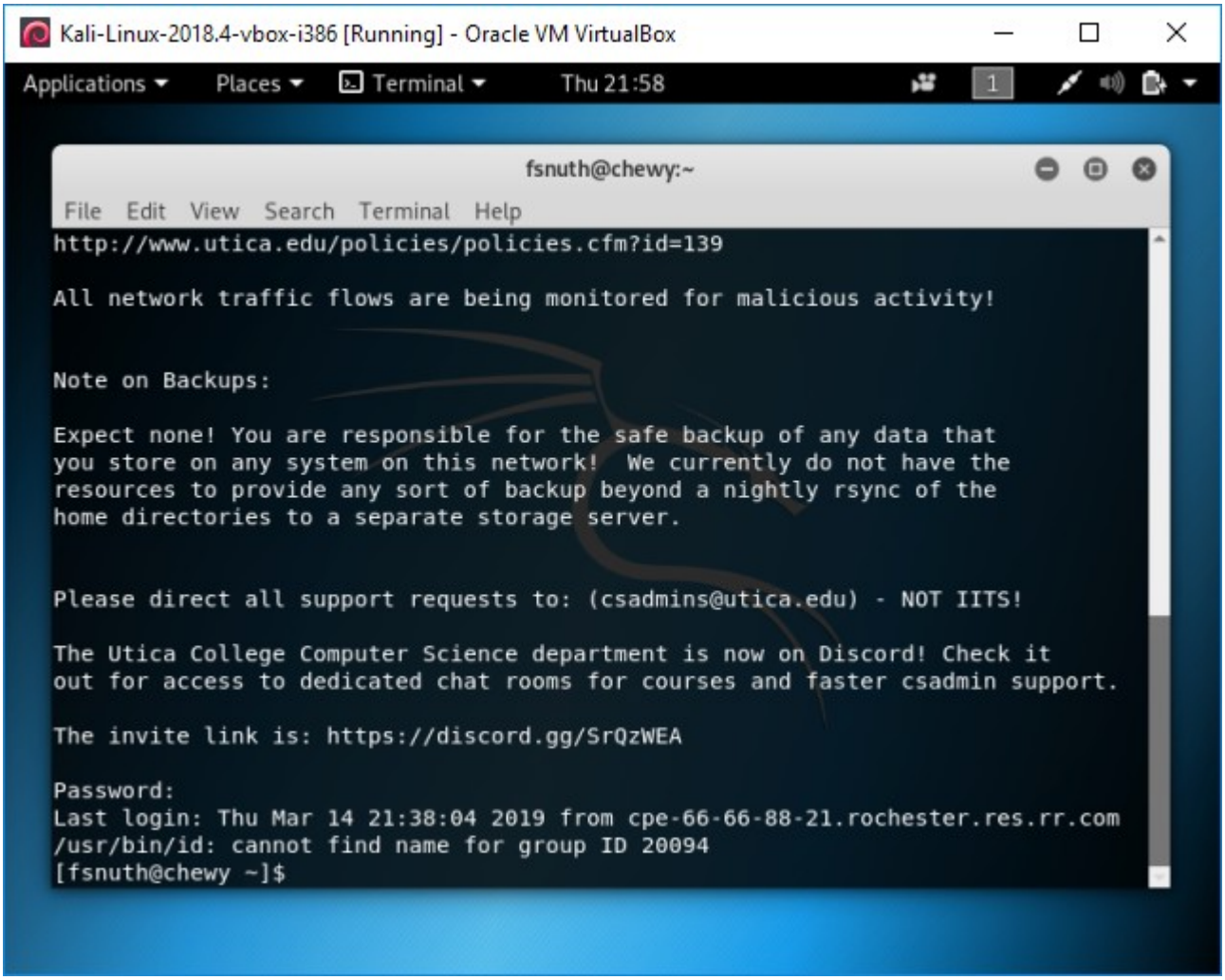
In this lab, I will be manipulating the firewall and network settings of my virtual Kali so that I can access my private network from anywhere at any time. I will be testing my ability to log in through Chewy normally and test if the SSH is working. I will also be using SSH to proxy my web traffic, which will allow me to listen in on traffic from specific ports. Then I will configure a file in my .ssh directory so that I can access Chewy without typing my username and the domain name, as well as generating an SSH key so I will have easier access to it. I will finish off by learning how to encrypt single files on my Kali Linux, as well as encrypting multiple files in a folder with GPG.

Introduction

I can access my Chewy account and my virtual network from anywhere in the world if I want to. With any computer, I can go to my router and web server for reconfiguring them to my needs. Not only that, I can access my Kali from ProxMox, and not through the command window because I made sure to be secure even with SSH port forwarding. The purpose of this lab will be to automate my access through Chewy and my network by eliminating the need for many certificates along the way. Doing this lab will give me faster access to my network and eliminates the need to remember any password for going to my network.

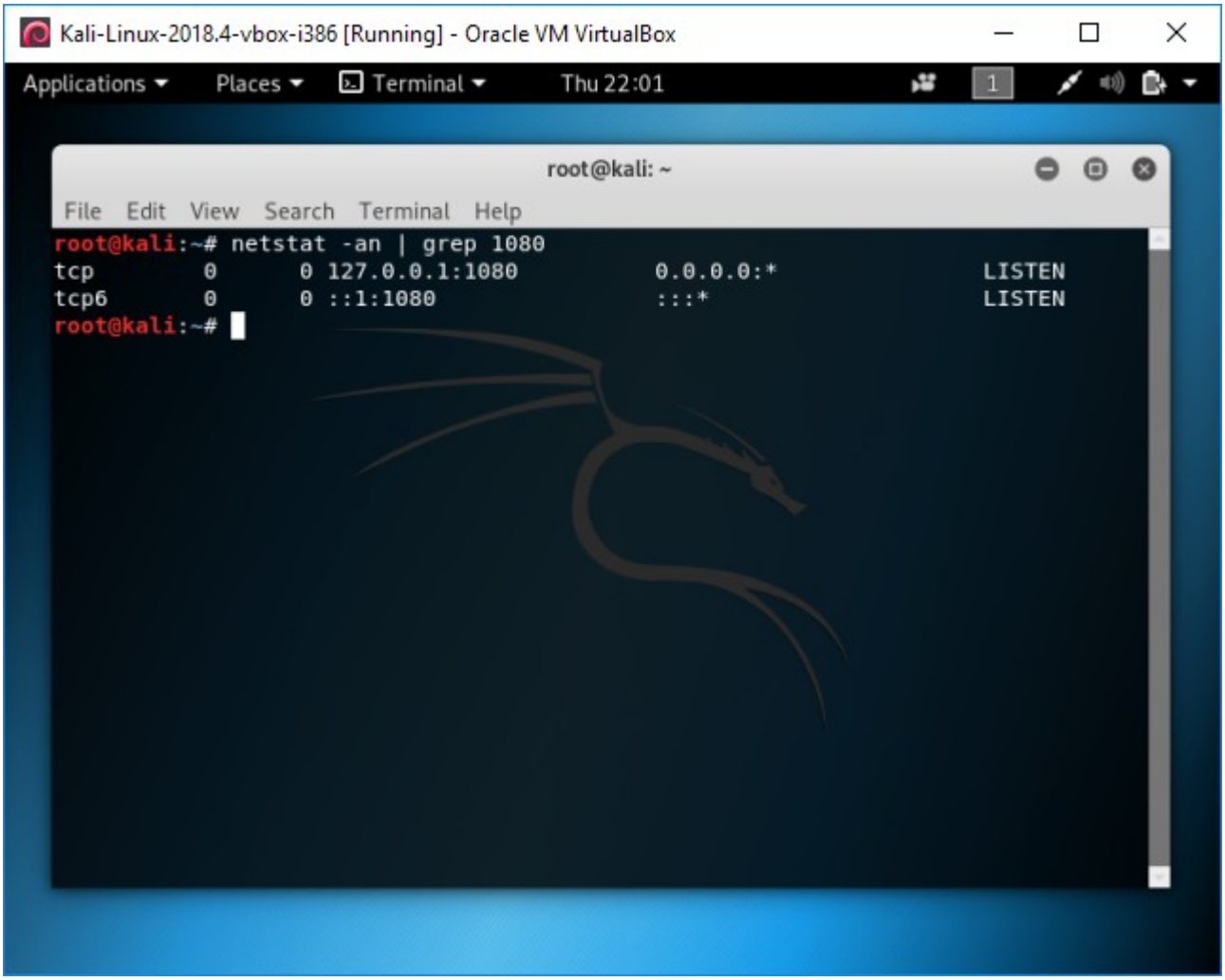
Processes & Screenshots

To start the lab, I need to see if I can access my Chewy account through any command window. I did this by opening a terminal on my personal Kali, and logged into my Chewy account with "ssh fsnuth@chewy.cs.utica.edu". This syntax will log into the account with the username "fsnuth" on Chewy, with the domain name "cs.utica.edu".



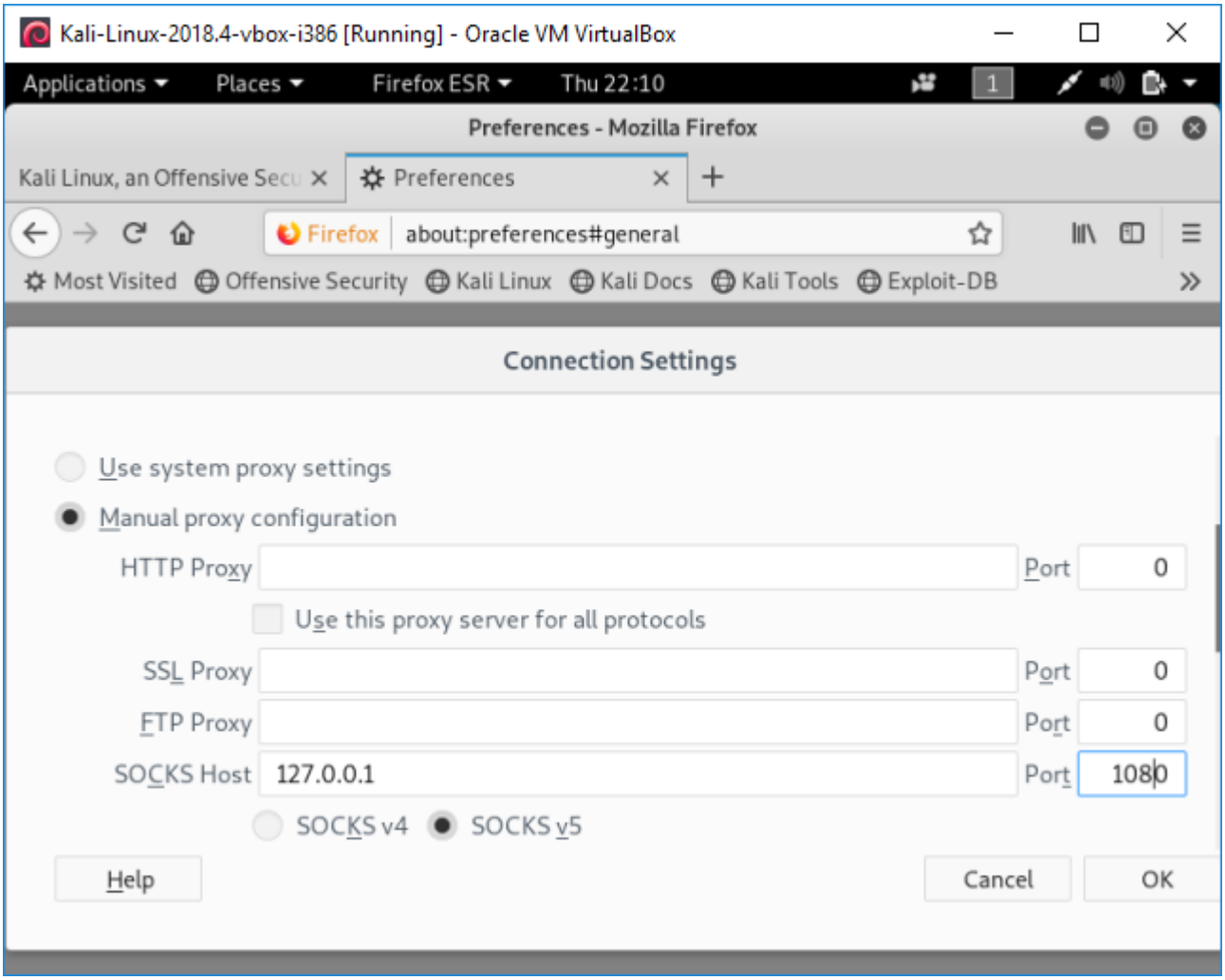
(Logging into my Chewy from my own Kali. I did this from my own Kali to see if I can truly access my network through SSH, and not just from college desktops or the Kali on my network.)

I exited the Chewy account and back into my Kali. After that, I typed "ssh –D 1080 fsnuth@chewy.cs.utica.edu". This will allow the local system to listen in on port 1080 traffic, which we will manipulate later with Firefox. I then typed "netstat –an | grep 1080" to see if any traffic is being picked up from that port.



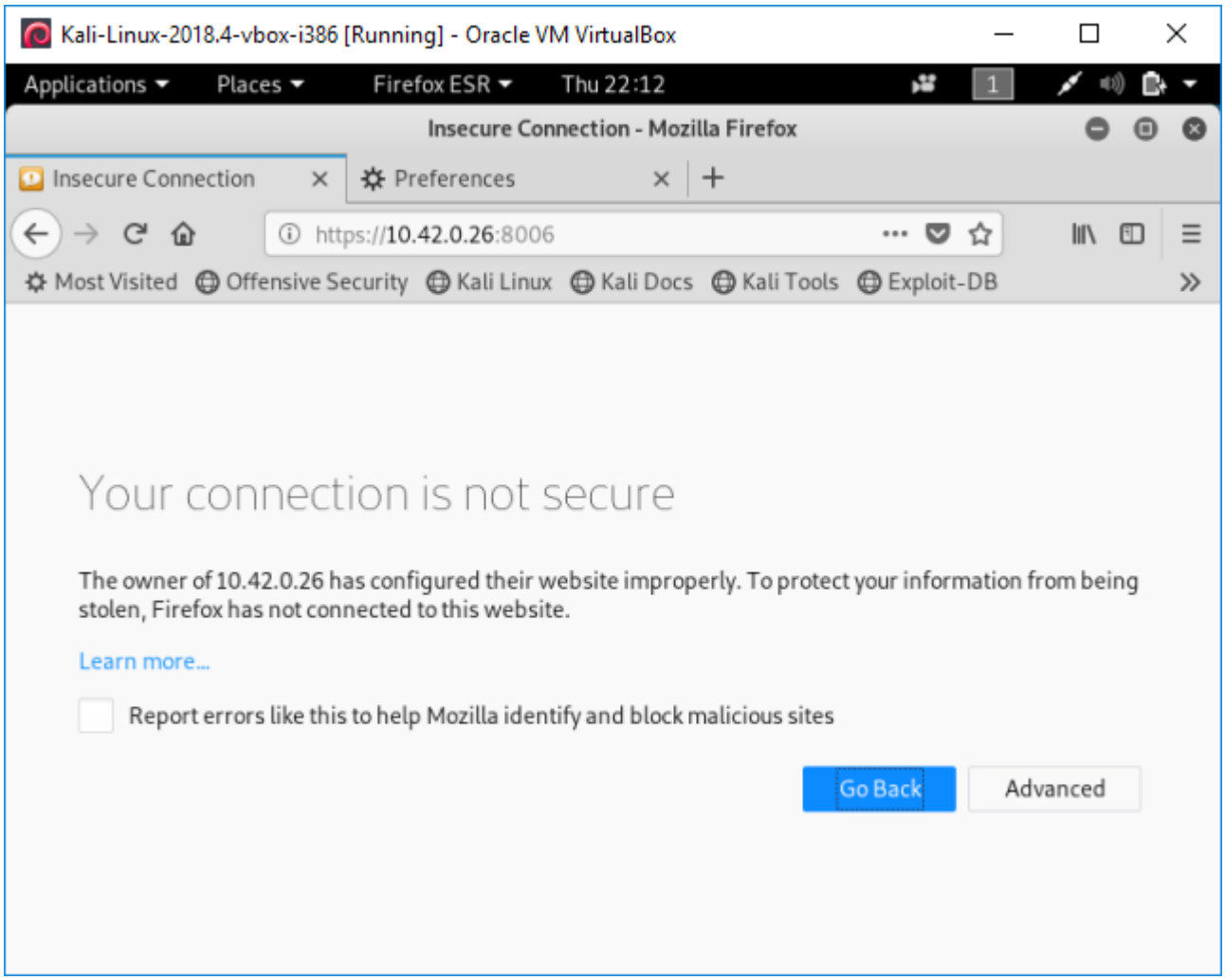
(The result of typing "netstat –an | grep 1080". The most interesting thing to note here is that 1080 port traffic is being listened to on 127.0.0.1, which is most likely the IP of my Chewy account.)

After confirming that my local system is listening for port 1080 traffic, I then move on to feed it web traffic through Kali's Firefox browser. The lab says to use IceWeasel, but that browser is not in my Kali, and the network settings menus are similar to the point where I have no reason to worry.

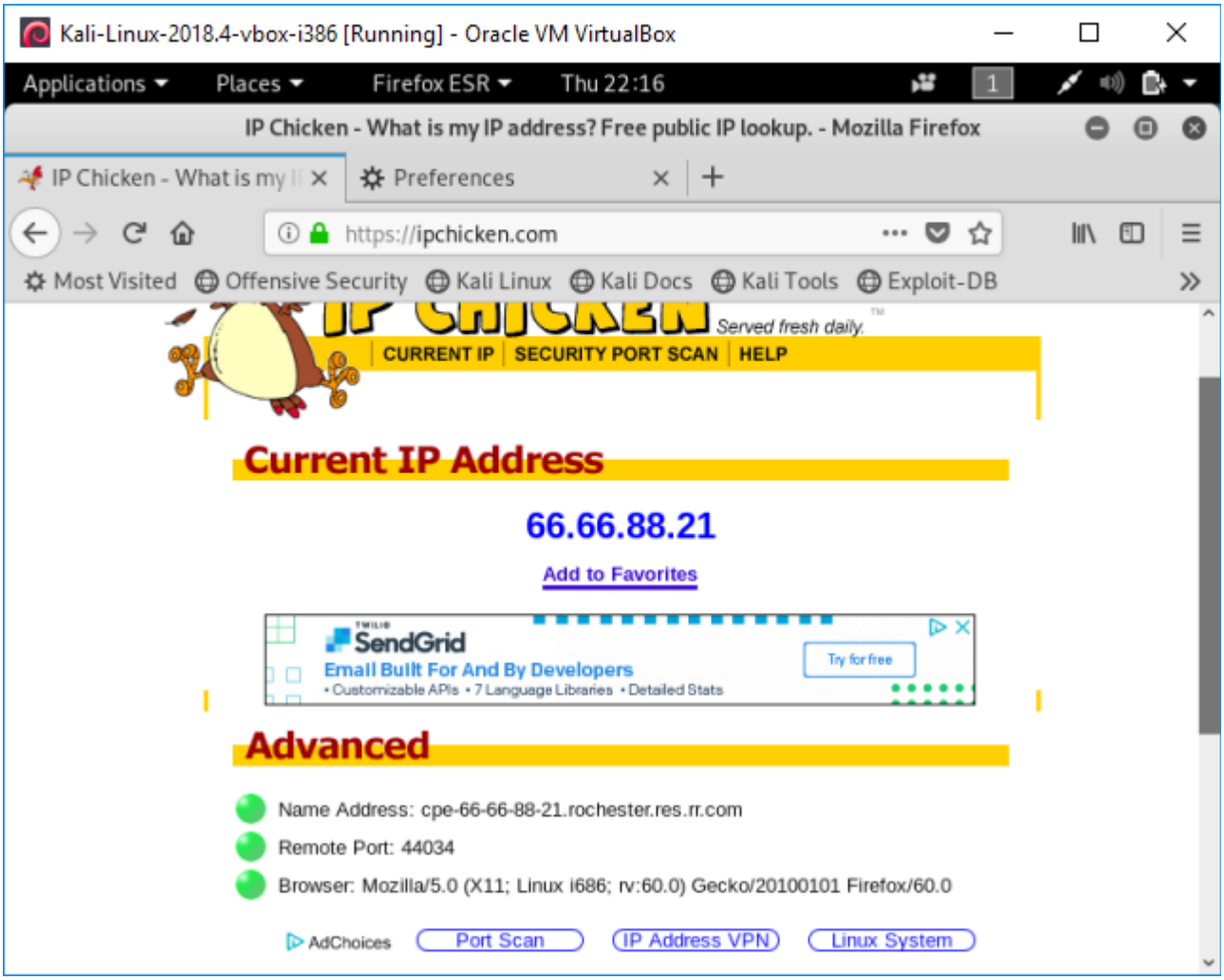


(Configuring a manual proxy on my Kali's Firefox network settings. I notice that the IP used for the SOCKS Host and the Port is the same information we found when listening with netstat.)

I typed "https://10.42.0.26:8006" into the Firefox browser, the private address for the VLE web portal. I have been denied, but the lab says this is part of the lab, so I moved on to the next step of immediately checking my IP address first with my manual settings, and the settings that DHCP gives me.



(Trying to get in the VLE web portal. This error message is significant in that it points to the owner not configuring the website properly rather than connection error.)

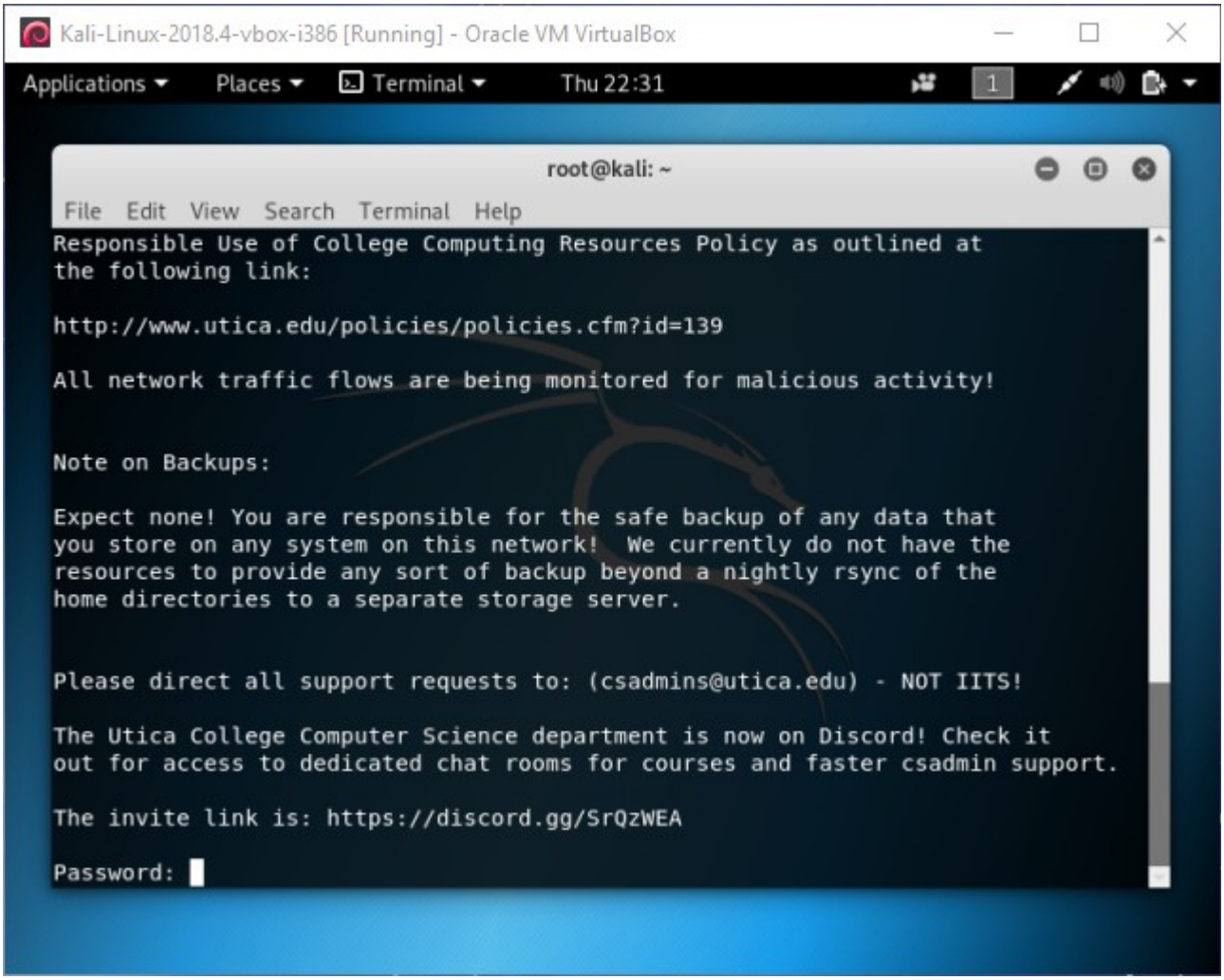


(My IP address after changing my network settings from manual to automatic, which was previously 4.26.24.234. I noticed that the IP address changed when I change the network settings. I think this sudden switching of IP address could be used for changing the flow of web traffic.)

After this, I created a config file in the .ssh directory of my Kali, and configured it with the following code:

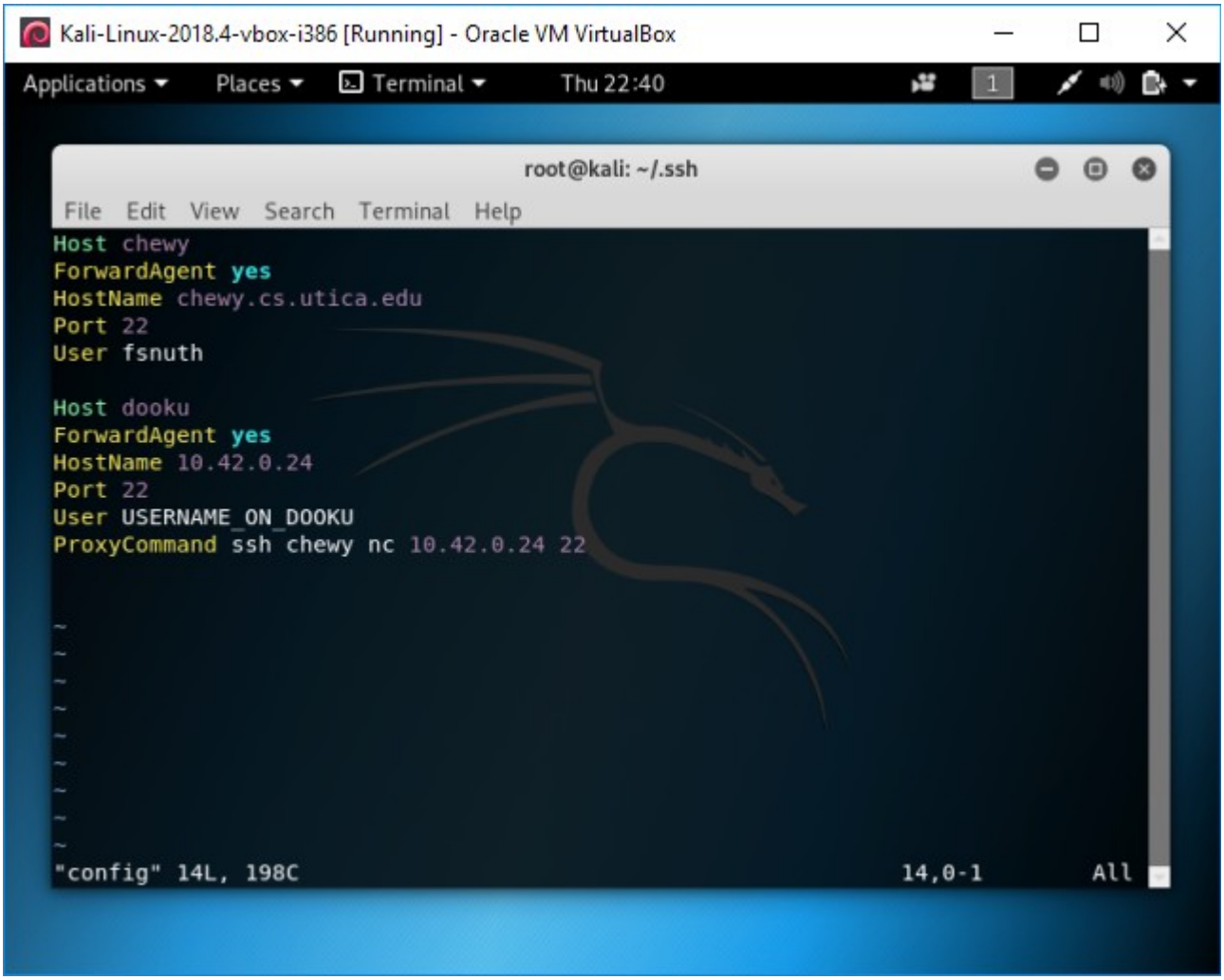
Host chewy
ForwardAgent yes
HostName chewy.cs.utica.edu
Port 22
User fsnuth

After putting the above code into the config file, I attempted to SSH back into Chewy to see if I have to typed "fsnuth@chewy.cs.utica.edu". It turns out I don't have to, reducing the things I have to typed to "ssh chewy".



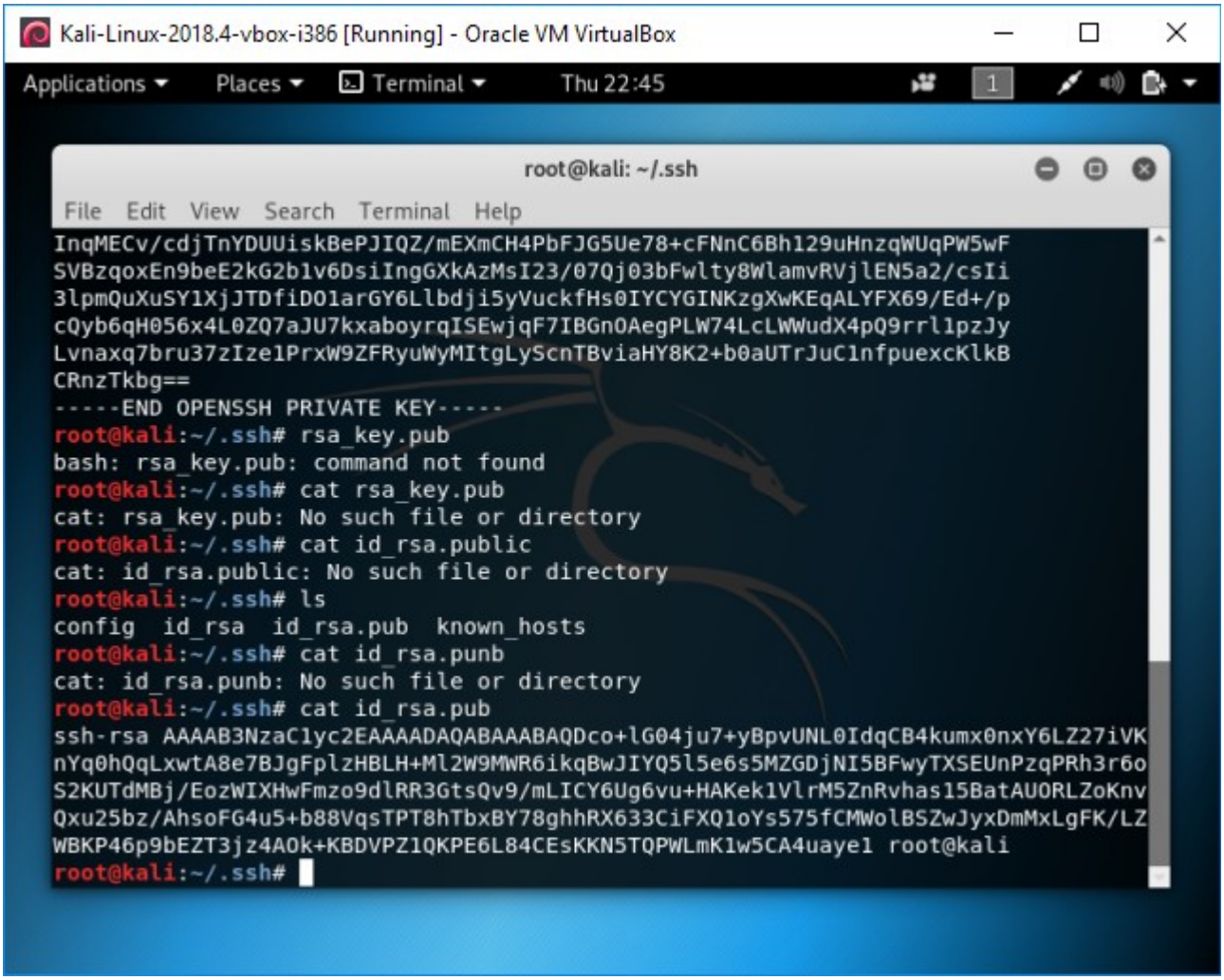
(SSH'ing into my Chewy after editing the config file.)

I also editited the file afterwards to configure my SSH config file for the host named dooku.



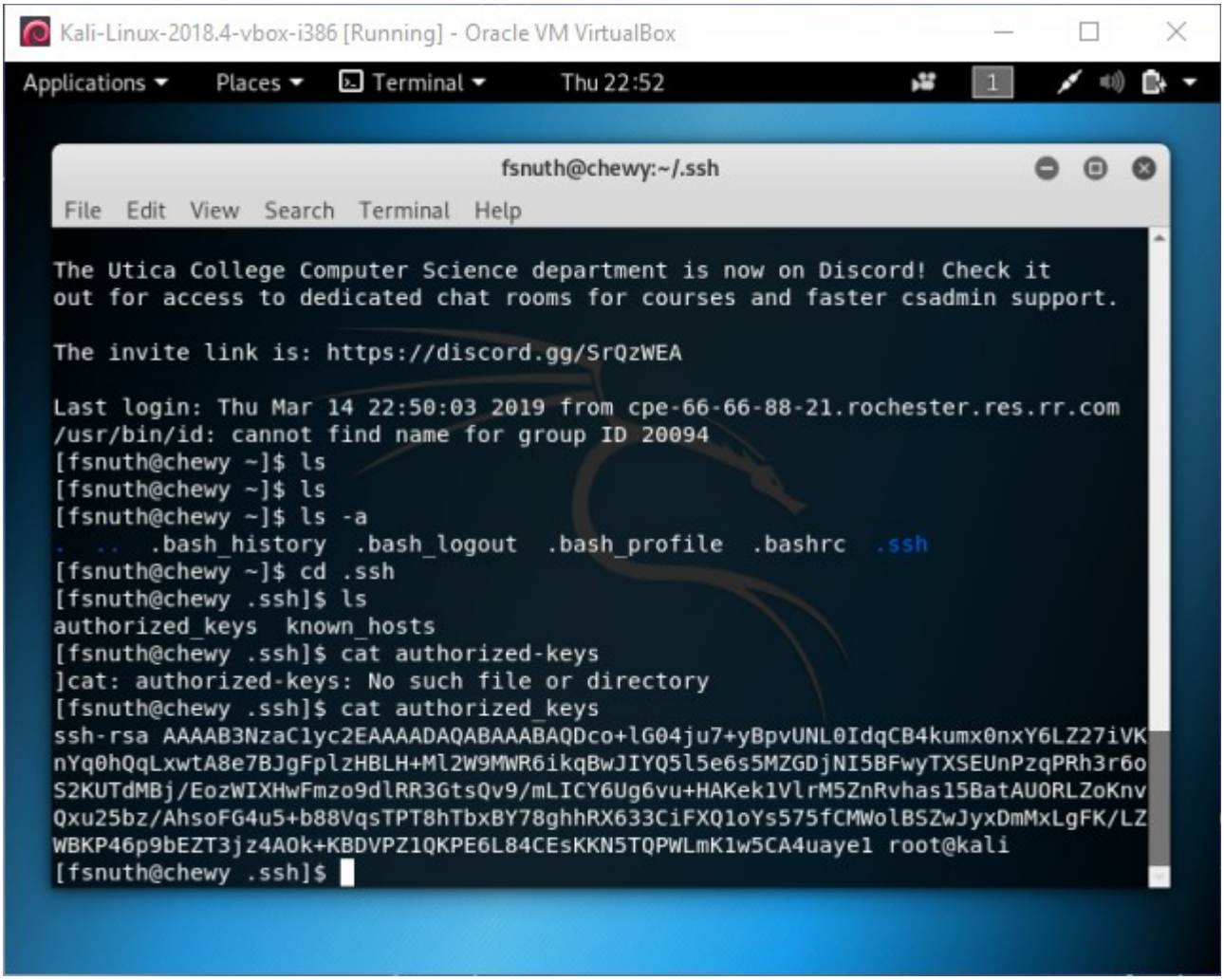
(Configuring the SSH file for the host named dooku.)

I also need to generate two SSH keys; one public and one private. These keys will not only be encrypted with RSA, but will also be used when I SSH into Chewy so that my password will only need to be typed in once.



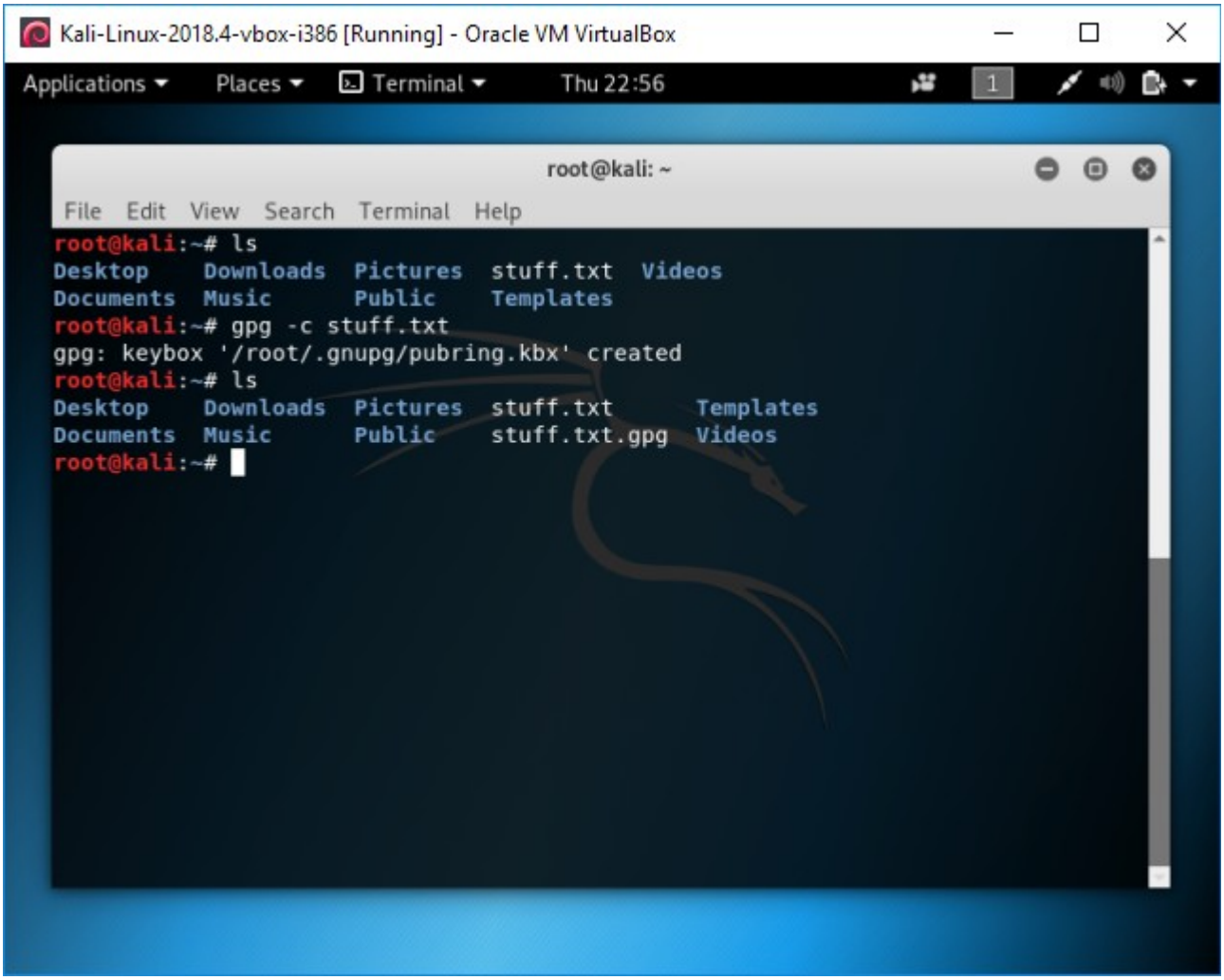
(Generating the SSH keys with "ssh-keygen".)

I went to my Chewy and configured the files in my .ssh directory. I checked the file "authorized_keys", which will show me the public and private keys that I generated earlier.



(Checking the RSA encrypted key in the "authorized_keys" file on Chewy's .ssh directory.)

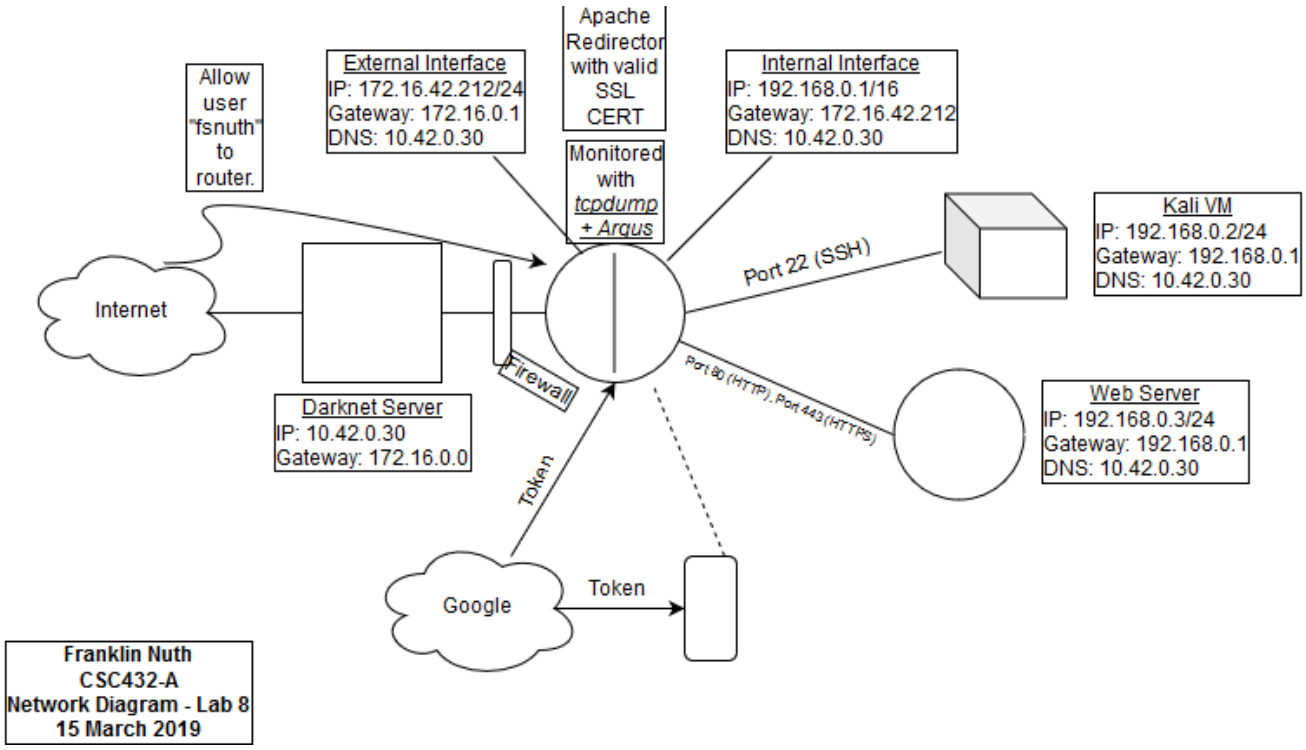
Finally, I will attempt to use the GNU Privacy Guard to encrypt my files in Kali Linux. First, will encrypt single text files with "gpg -c <filename>" and encrypted multipled files by putting them in a zip folder and encrypting said zip folder with the same syntax. If I ever need to decrypt any file or zip folder, I just need the password and I am set to decrypt with GPG.



(Encrypting a single file with GPG. I have a normal file called "stuff.txt" and an encrypted file with the same name and a .gpg file extension.)

```
Zuhairs-MacBook-Pro:Desktop zuhairhallak$ gpg --decrypt franklinEncryptedFile.tx
t.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
"I am a pointless line of text XD."
Zuhairs-MacBook-Pro:Desktop zuhairhallak$
```

(The result of my partner, Zuahir, receiving my encrypted text file and decrypting it with the passphrase I emailed him.)



(My network diagram after doing Lab 8. With my Kali, I can easily access the router since it recognizes me by my SSH public and private keys.)

Issues & Resolutions

Surprisingly, I did not run into any issues upon doing this lab.

Conclusion

In this lab, I have manipulated the firewall and network settings of my Kali so I can access my network from any computer with SSH. I have tested out how my IP address works under manual and automatic settings, as well as listening in on certain IP addresses. With SSH key generation, I can use public and private keys so that I only need to enter my password once to access my router. This lab taught me that accessing my router does not always have to be a hassle, and that there are ways to circumvent that. I might also have a bit of fun encrypted my files with GPG in the near future.

References

Bayden, Asmali. 18 November 2018. *How To Zip, Unzip Files In Linux?* Retrieved from: http://www.easybib.com/reference/guide/apa/website

Hoelting, Joel. Hoelting Joel. 2016 October 10. *Video: CLC #1 - GPG Simple File/Folder Encryption.* [Video File]. Retrieved from: https://www.youtube.com/watch?v=C0l3Oekix2M