Student Name:

Course: CSCE 5550

Semester: Fall 2021

Lab 4: Firewalls and IDS

In this lab, you will be using a special version of the **Ubuntu 20** VM.
Download the respective OVA file from the lab page and import it as usual.

The credentials are the same as before:

Username: sec-lab
Password: untccdc

Also, you will use the Kali 20 VM, which will be the same as in the previous labs.

The credentials are:

Username: osboxes
Password: osboxes.org

NOTE: Address all the questions (Q1, Q2, etc.) marked in bold.
When a screenshot is requested, try to fit all the results in one image.
If this is not possible, then attach multiple screenshots.

When a question is asked, e.g., "Who is an owner of the file?", type your answer, do not simply provide a screenshot.

IMPORTANT: Before running the lab, customize the command prompt in the Ubuntu VM to show your EUID – refer to the manual provided on the lab page.

Introduction

A firewall is a software- or hardware-based network security system that controls the incoming and outgoing network traffic based on applied set of rules.

Firewalls are classified into different types: Software firewall, hardware firewall, host-based, network-based, etc. Most of the operating systems have a software firewall to protect against network threats. Hardware firewall is a dedicated piece of hardware which protects one or more networks against threats originating from the Internet. Many routers that connect networks have some firewall features built into

them. Host-based firewalls are those that are present in a single host and that only manage the incoming and outgoing traffic for that host. A software firewall within an OS is a typical host-based firewall.

General-purpose computers such as a PC or server (or even a VM) with multiple network adapters can be turned into a router or firewall by setting up suitable software on it. For example, a Linux PC with two or more network adapters can be made a firewall by installing firewall software such as **iptables**.

Some other network related terms you need to understand to do this lab are the following:

Subnet: A subnet is a logical group of IP addresses or a subdivision of an IP network. The subnet mask, 32-bit values define the range of IP address in that subnet. For nodes in two different subnets to communicate, a router has to be present between them.

Port Forwarding: Port forwarding is a way of making a computer on your home or business network accessible to computers on the Internet, even though they are behind a router. It is commonly used in gaming, security camera setup, voice over IP, and file download. After you have forwarded a port you are said to have an open port. In this lab, we will make your VM accessible to your host computer, and to the computers that belong to your local network.

UFW (Uncomplicated Firewall): This is a default firewall configuration tool for Ubuntu Linux. Developed to ease iptables firewall configuration, UFW provides a user-friendly way to create an IPv4 or IPv6 host-based firewall. By default, UFW is disabled.

Configuration of port forwarding in the VM

Let us configure the VirtualBox VM network mode as “NAT”. Your host OS (i.e., the OS of your host computer running VirtualBox) will access to the VM through the port forwarding configured in the network setting of the VM.

1. Setup Ubuntu VM network settings as internal:

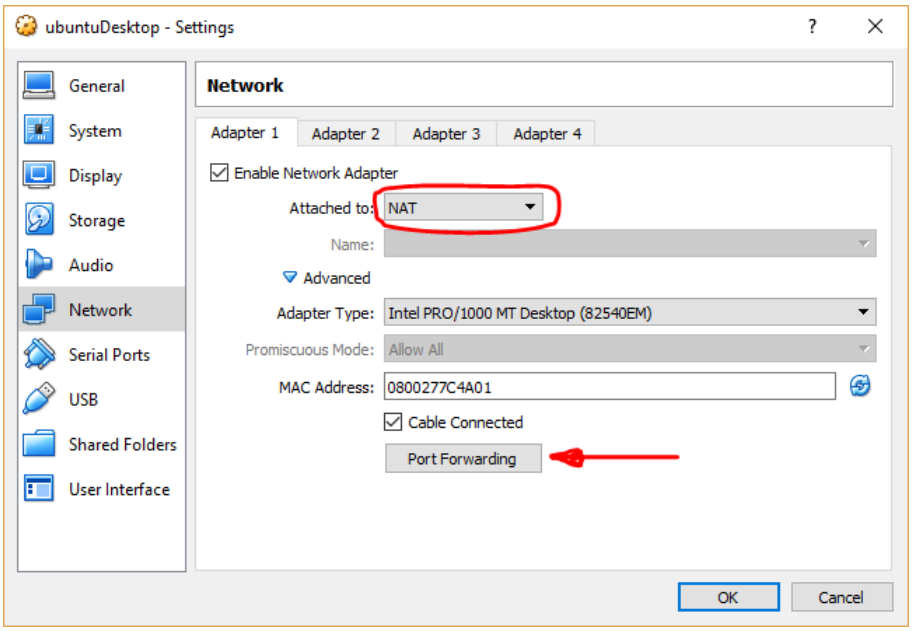
   In VirtualBox: Select the Ubuntu VM → Right-click → Settings → Network.
2. Under Adapter1, select “NAT” from the dropdown menu.
3. Click on Port Forwarding as shown on the below screenshot:

Student Name:

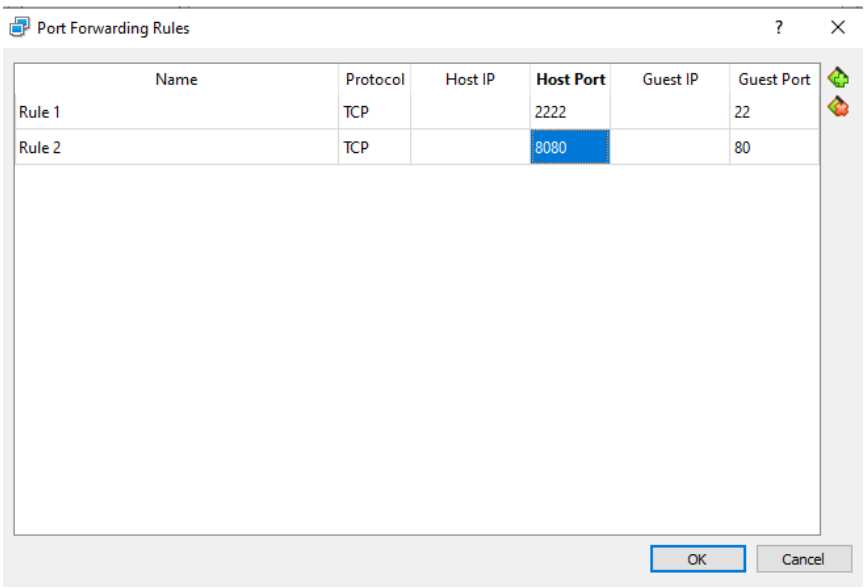
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



Then, enter the port forwarding rules as shown on the next screenshot:



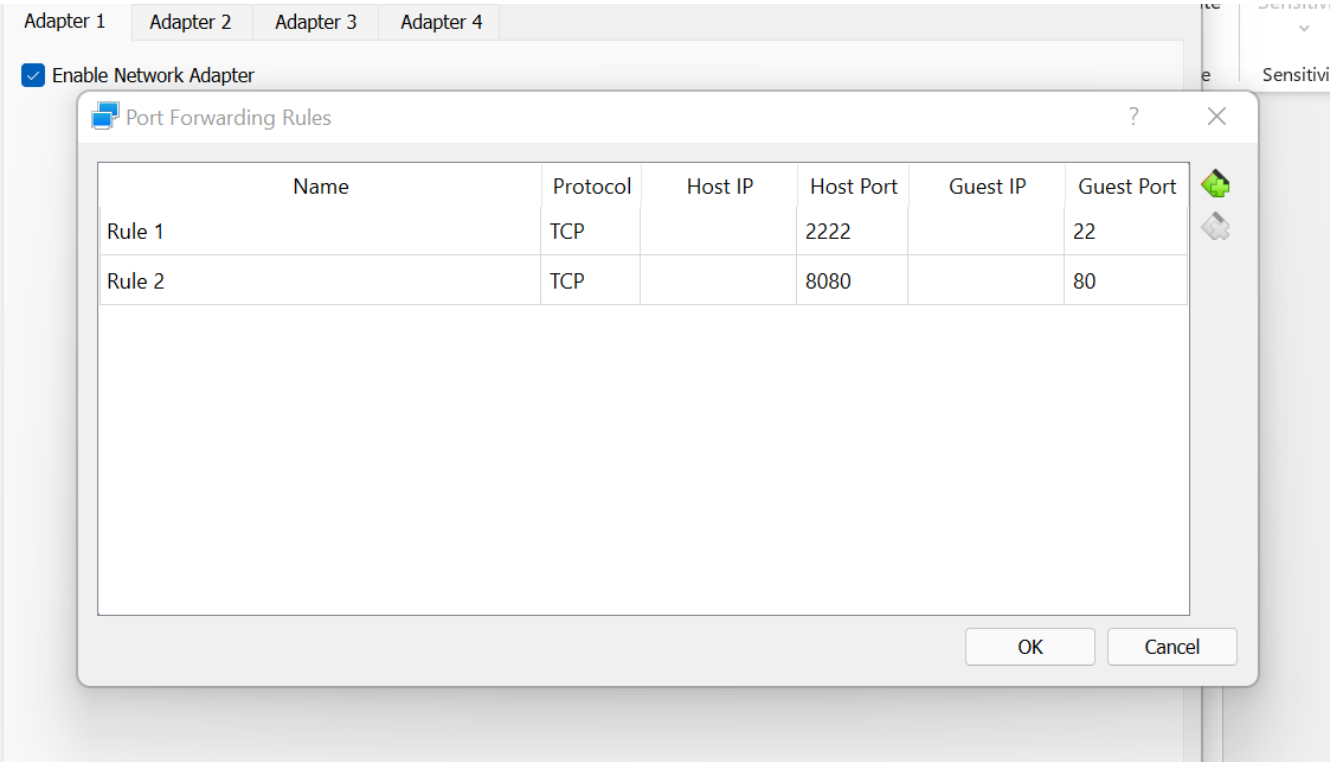
Q1: Attach a screenshot of your port forwarding rules. (It should look the same as above.)

Student Name:

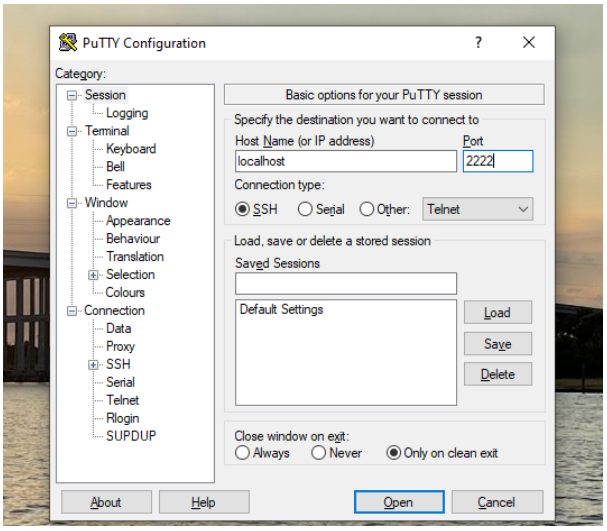
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



Now, let us confirm that the port forward works. If you use Windows open PuTTY on your host computer (an SSH/telnet client – install it, if you do not have it) and connect to localhost:2222.



If you use Linux/Mac OS, type “ssh sec-lab@localhost:2222” in the terminal.

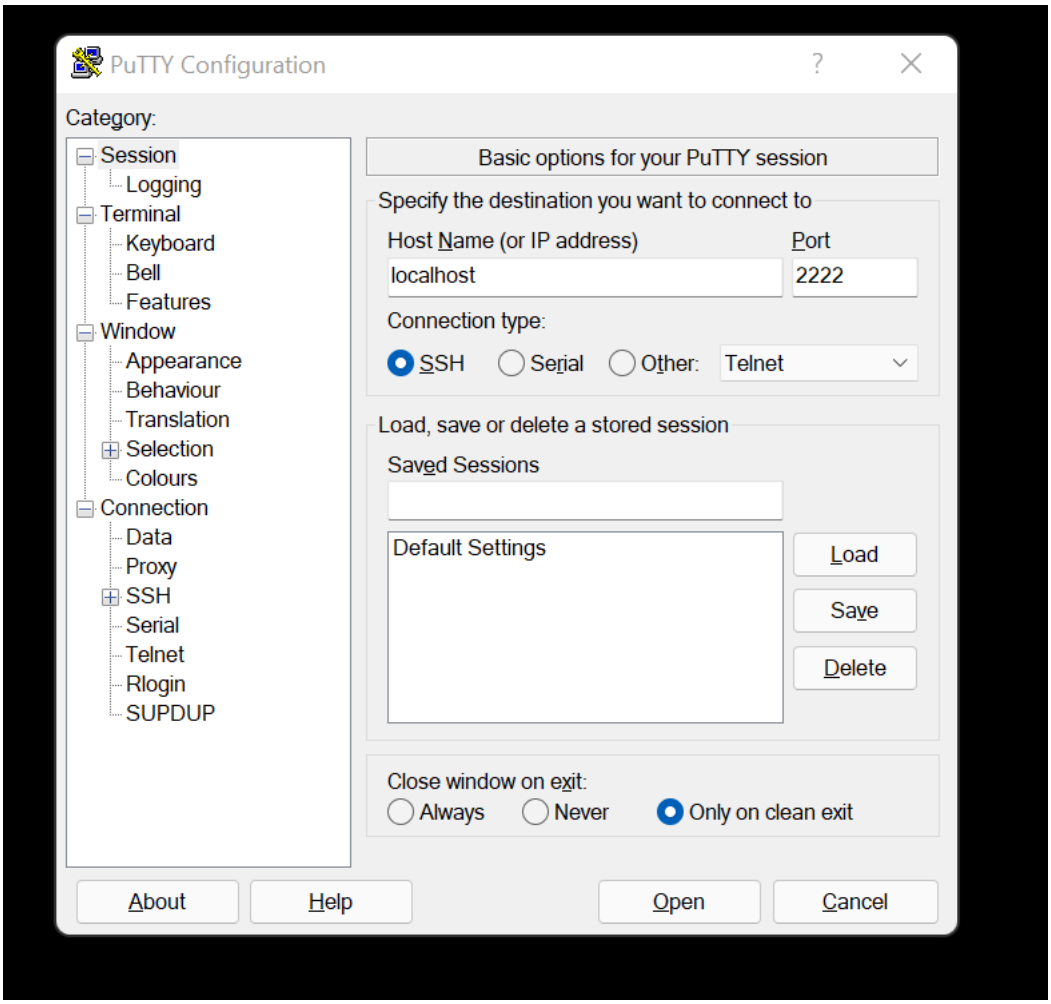
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

Q2: Attach a screenshot of the result.

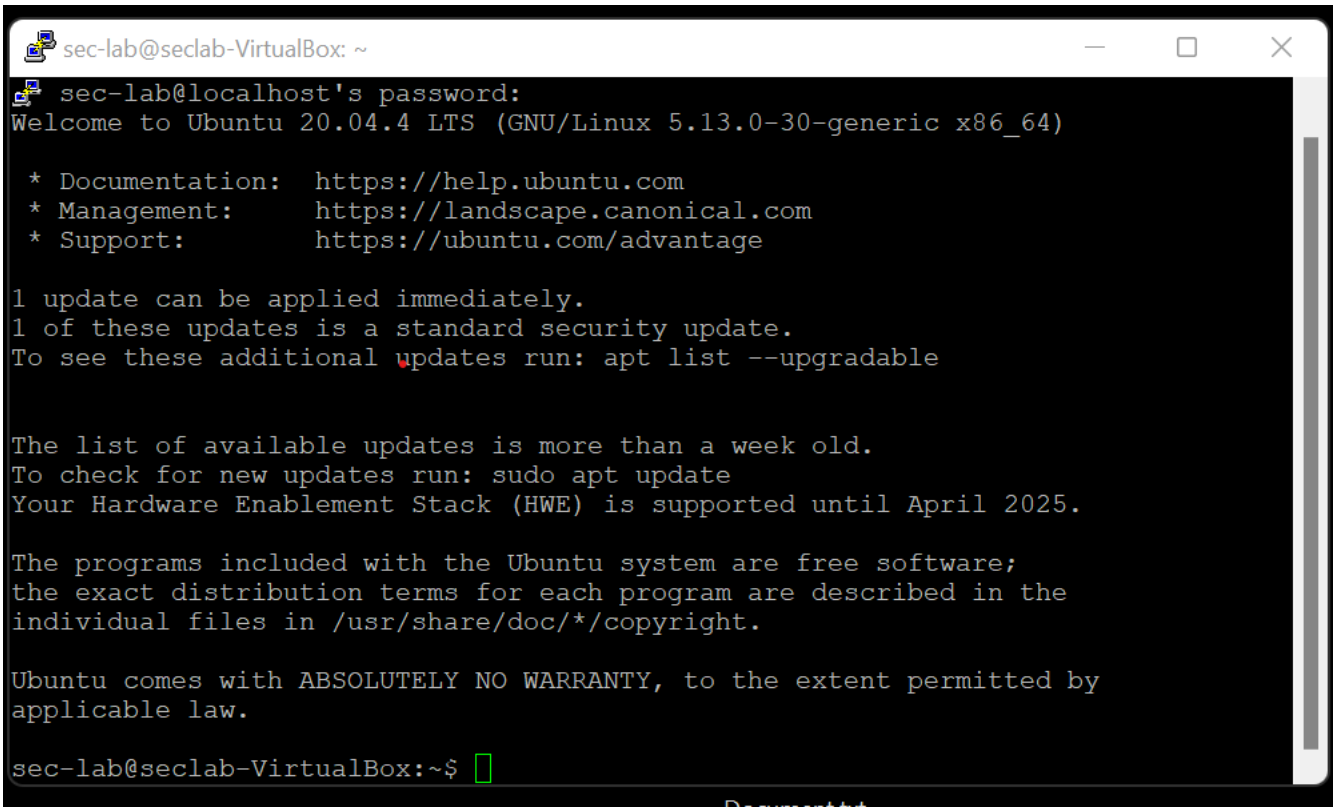


Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



On your host computer, open the web browser and access “localhost:8080”.

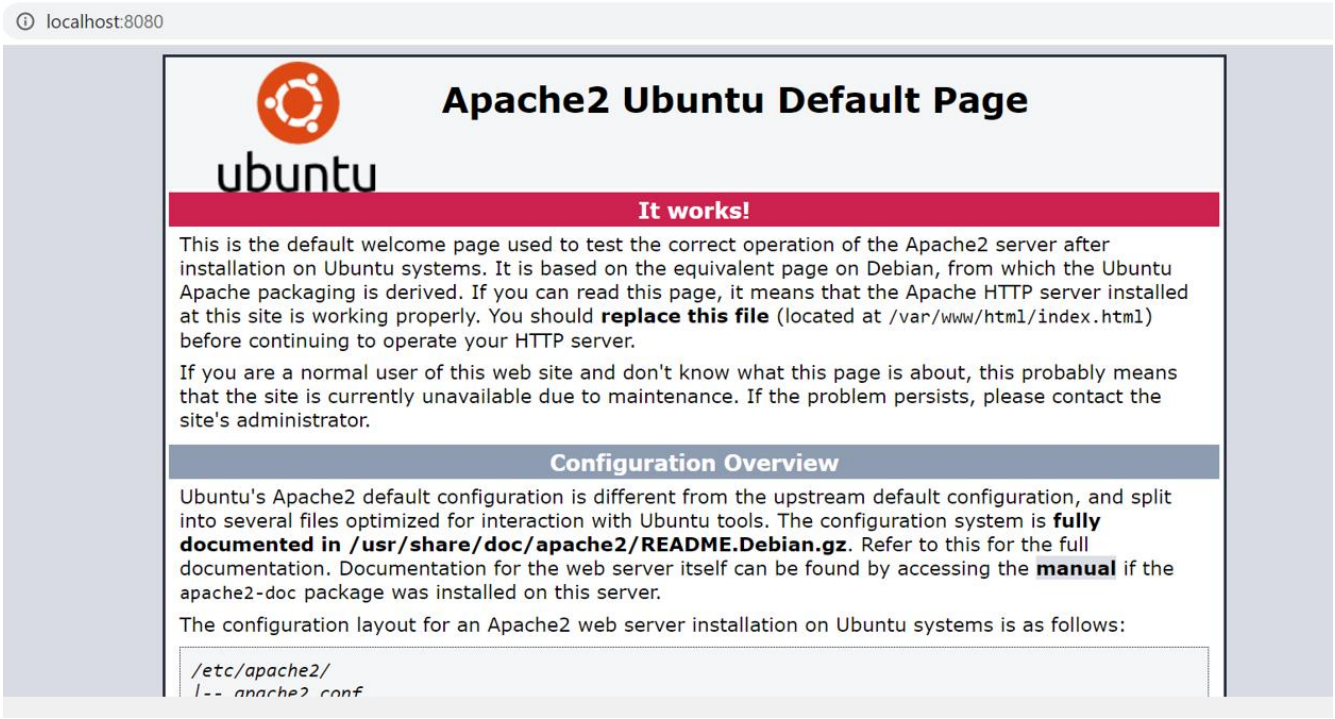
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

Q3: Attach a screenshot of the result.



Section 1: UFW - Uncomplicated Firewall

When UFW is activated, it uses a default set of rules (profile) that should be fine for the average home user. In short, they can be summarized as all 'incoming' is being denied, with some exceptions to make things easier for home users.

1. To turn UFW on with the default set of rules, type in the Linux terminal:

   sudo ufw enable

2. To check the status of UFW:

   sudo ufw status verbose

The output should resemble the following:

youruser@yourcomputer:~$ sudo ufw status verbose

[sudo] password for your user:

Status: active

Logging: on (low)

Default: deny (incoming), allow (outgoing)

Student Name:

Course: CSCE 5550

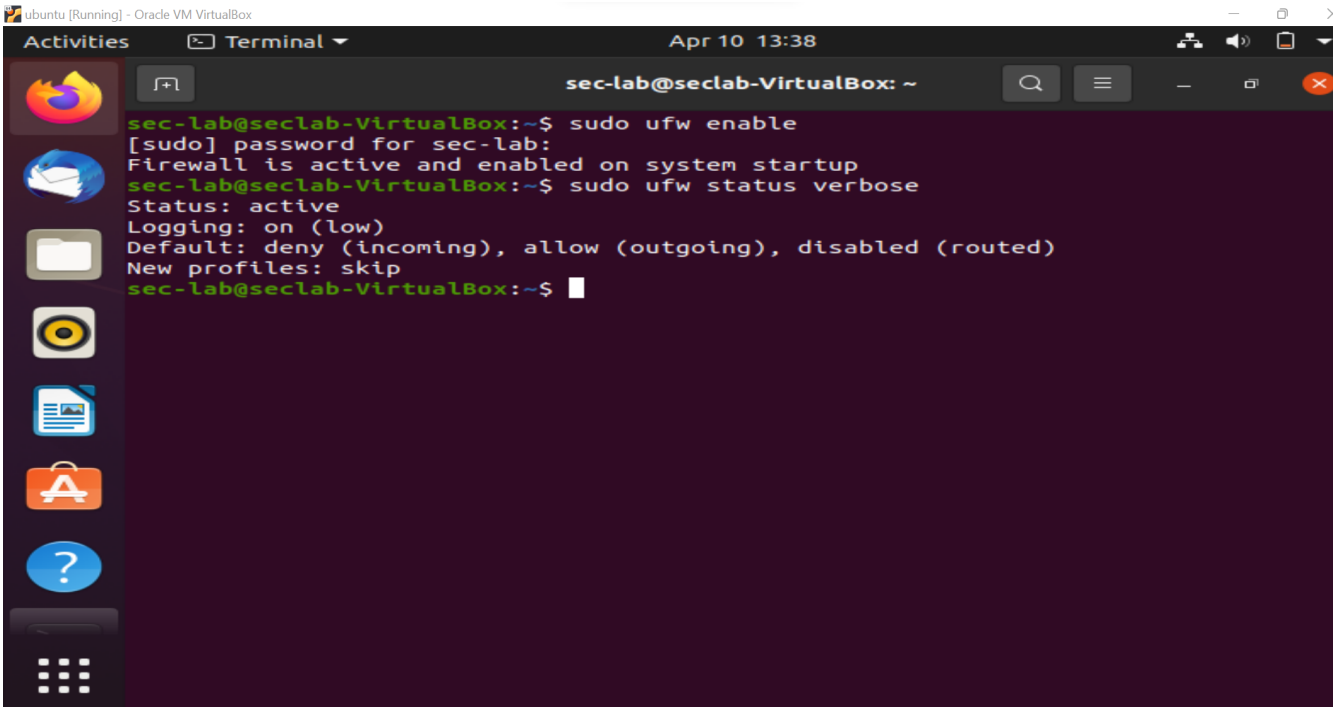
Semester: Fall 2021

Firewalls and IDS Lab

New profiles: skip

youruser@yourcomputer:~$

Q4: Attach a screenshot of the terminal showing the status of UFW.



3. Now, we will try to block incoming SSH packets by blocking the port using UFW.
   In the Ubuntu VM, type:

   sudo ufw deny ssh

4. From your host, try to establish an SSH connection again, as you did before Question 2.

5. Also try to connect on browser with “http://localhost:8080”.

Q5: Are you able to connect to your VM or get access to localhost website?
Attach a screenshot and explain what happened.

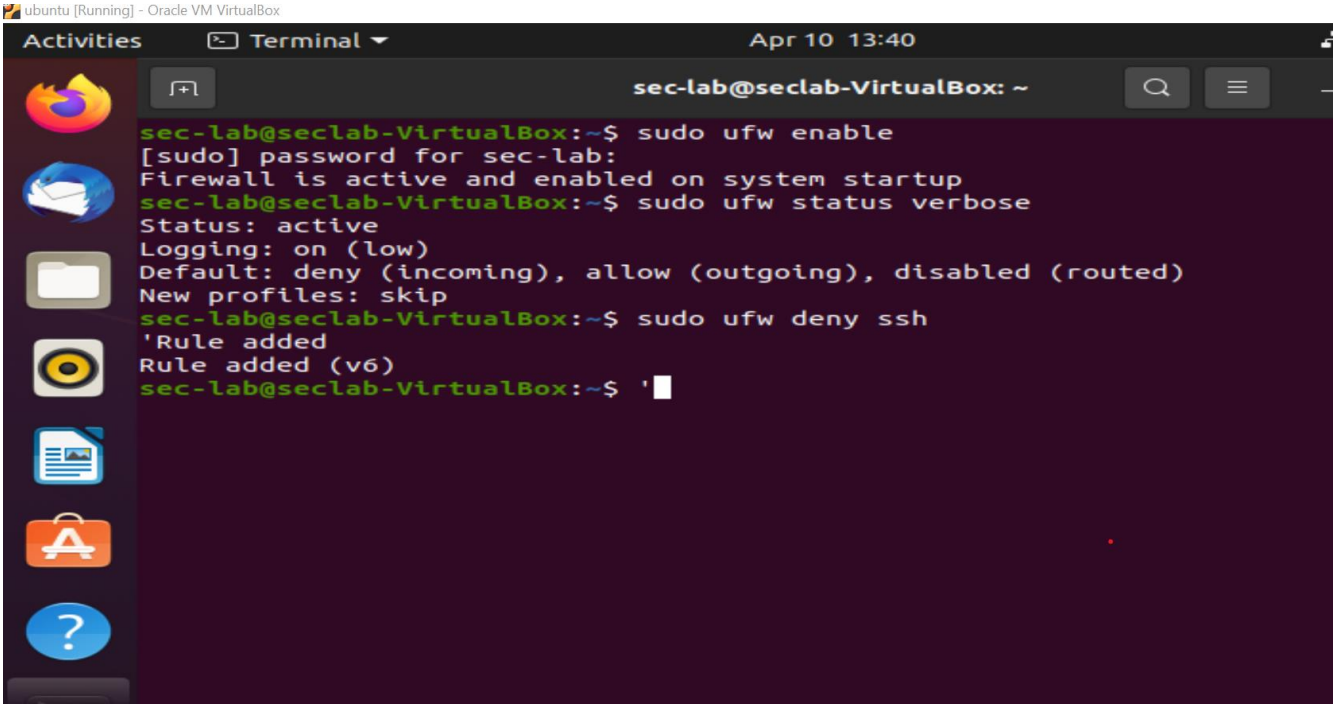
Student Name:

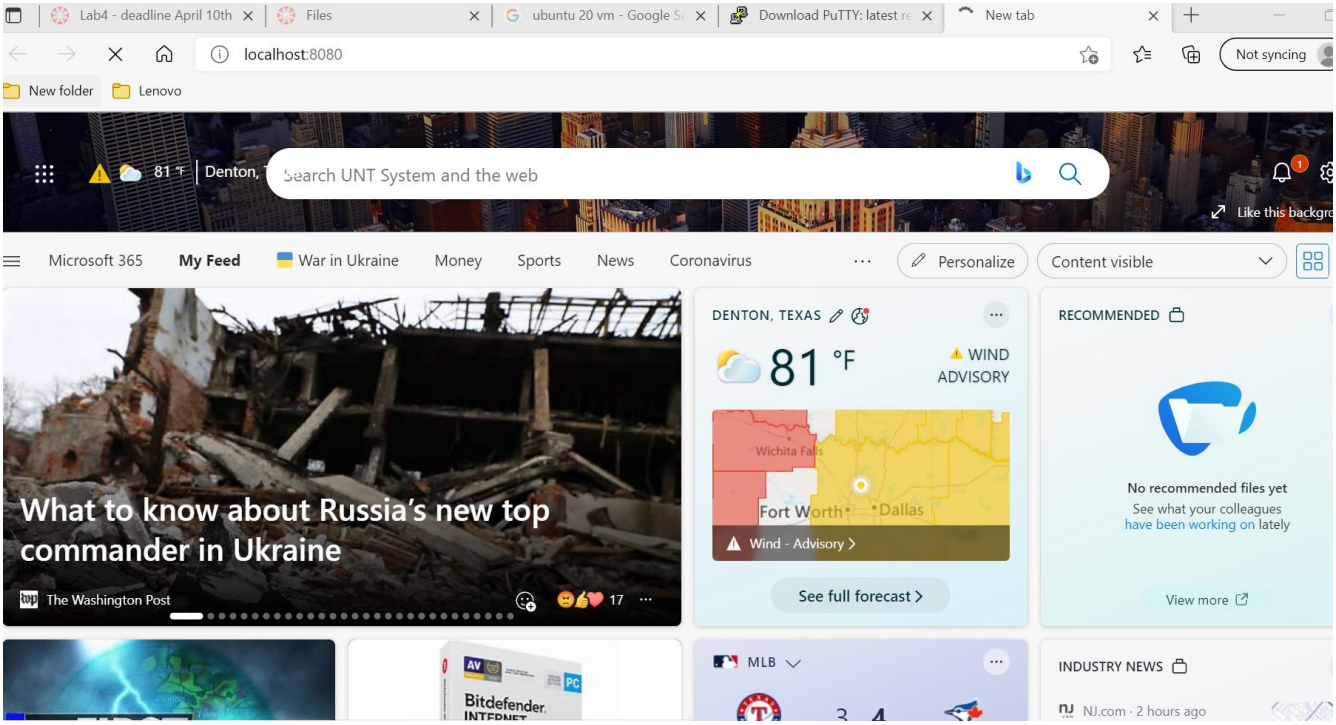
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

We have executed command “sudo ufw deny ssh” (denying the ssh connection). Hence, we cannot connect to localhost.





Student Name:

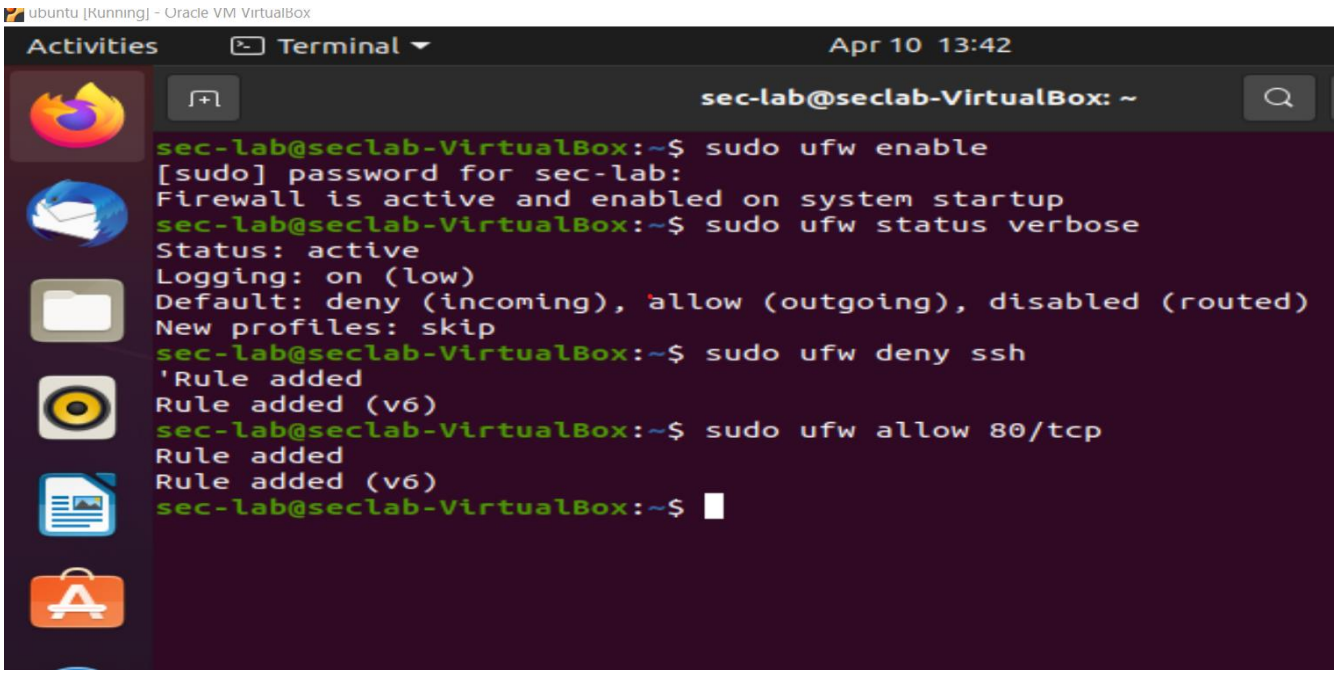
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

6. We are going to allow HTTP connections to the guest (Ubuntu VM acts now as a server). To allow incoming TCP packets on port 80 (http), type in the terminal of the Ubuntu VM:

   sudo ufw allow 80/tcp



Note: We allow port 80 because it is the port of the HTTP server installed in the Ubuntu VM. The port 8080 is not used in the Ubuntu VM, it is only used for access from the host and it is redirected to port 80 of the VM.

From your host, try to access “localhost:8080” in the browser.

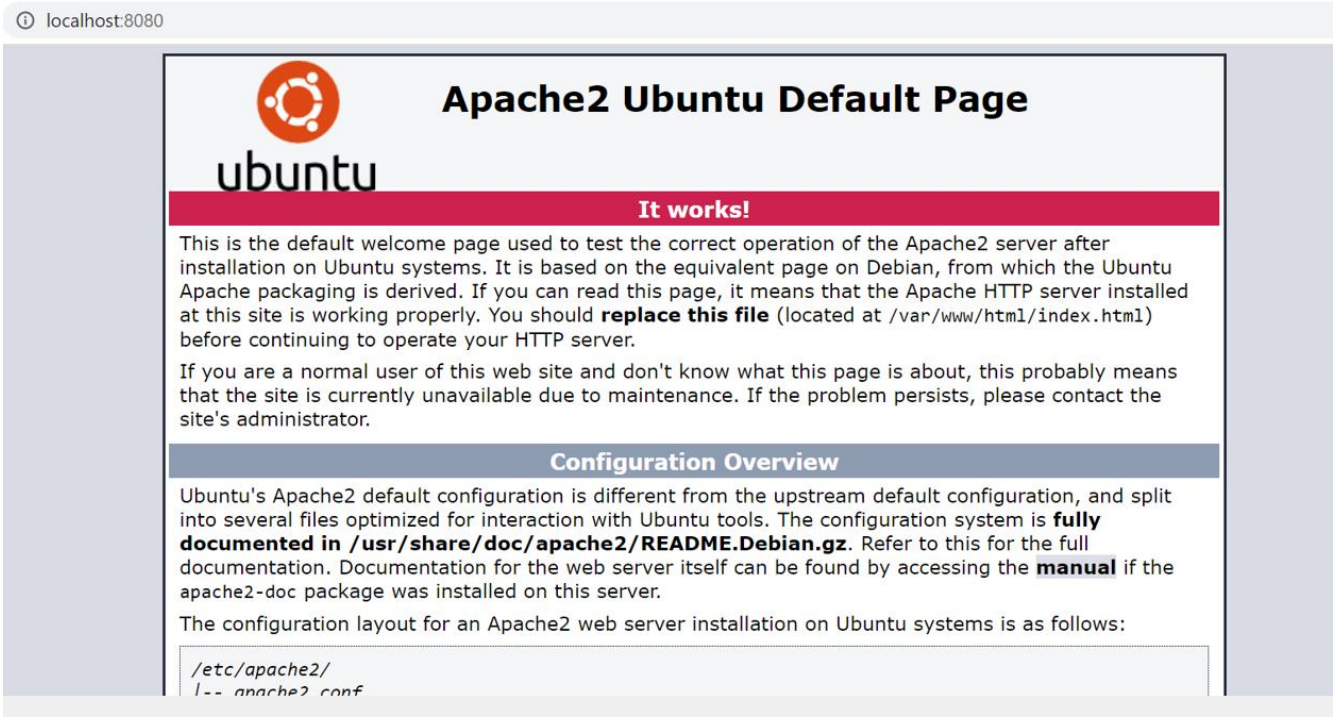
Q6: Attach a screenshot of the resulting browser window.

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



Note: In the rest of the lab, we are going to deny outgoing connections from the Ubuntu VM.

7. Using UFW to block outgoing traffic to website “www.cnn.com”. The command syntax is as follows:

   sudo ufw deny out from any to _ipaddress_

   To write the above command, we need to know the IP address of the server www.cnn.com.

8. Type:

   host -t a www.cnn.com

   You may expect the output similar to the following:

   www.cnn.com is an alias for turner-tls.map.fastly.net.

   turner-tls.map.fastly.net has address 151.101.1.67

   turner-tls.map.fastly.net has address 151.101.65.67

   turner-tls.map.fastly.net has address 151.101.129.67

   turner-tls.map.fastly.net has address 151.101.193.67

Student Name:

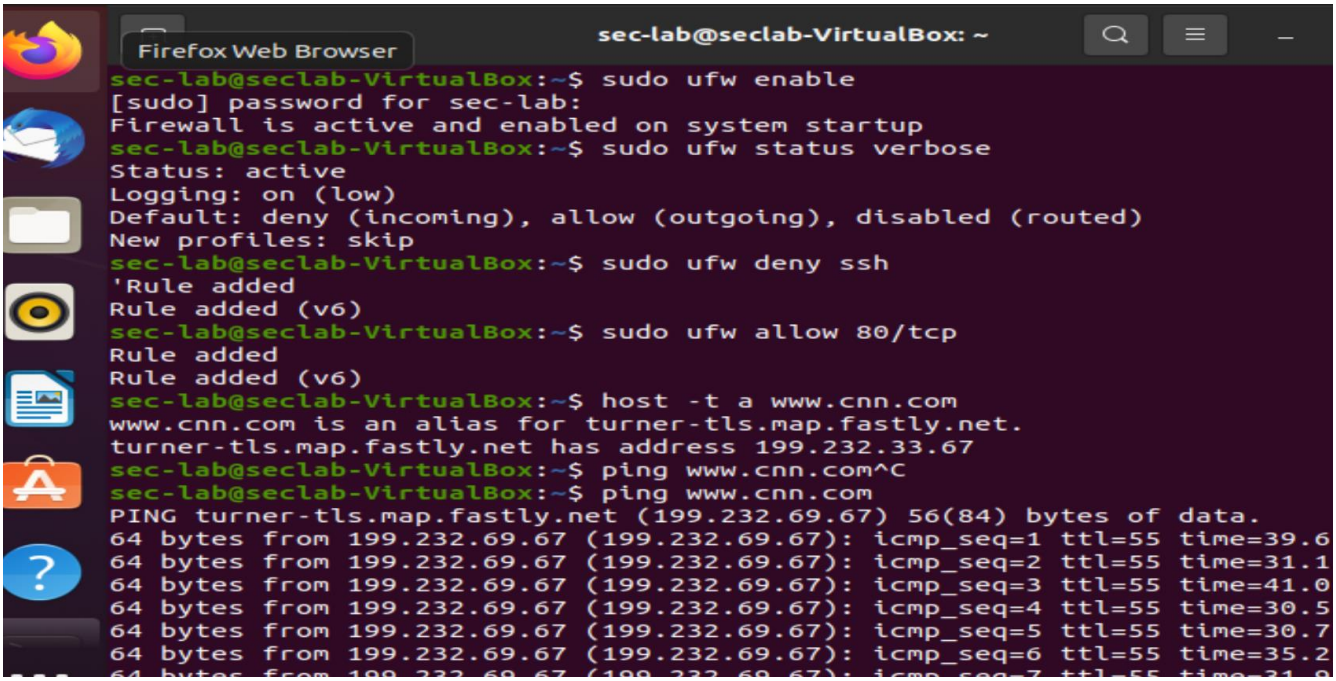
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

The main IP address of www.cnn.com is 151.101.1.67, but the server has alternative IP addresses 151.101.65.67, 151.101.129.67, and 151.101.193.67.

Q7: Attach a screenshot of the result.



Note: Alternatively, you can use the ping tool, but in this case you will only get the main IP address. Type “ping www.cnn.com” and press “Control + C”, the output will be similar to the following:

PING turner-tls.map.fastly.net (151.101.1.67) 56(84) bytes of data.

64 bytes from 151.101.49.67: icmp_req=1 ttl=52 time=98.8 ms

64 bytes from 151.101.49.67: icmp_req=2 ttl=52 time=28.8 ms

64 bytes from 151.101.49.67: icmp_req=3 ttl=52 time=79.2 ms

Test in the guest OS (Ubuntu) browser, that you are able to access the www.cnn.com website.
(Note: All tests in this lab are done in the browser of the Ubuntu VM.)

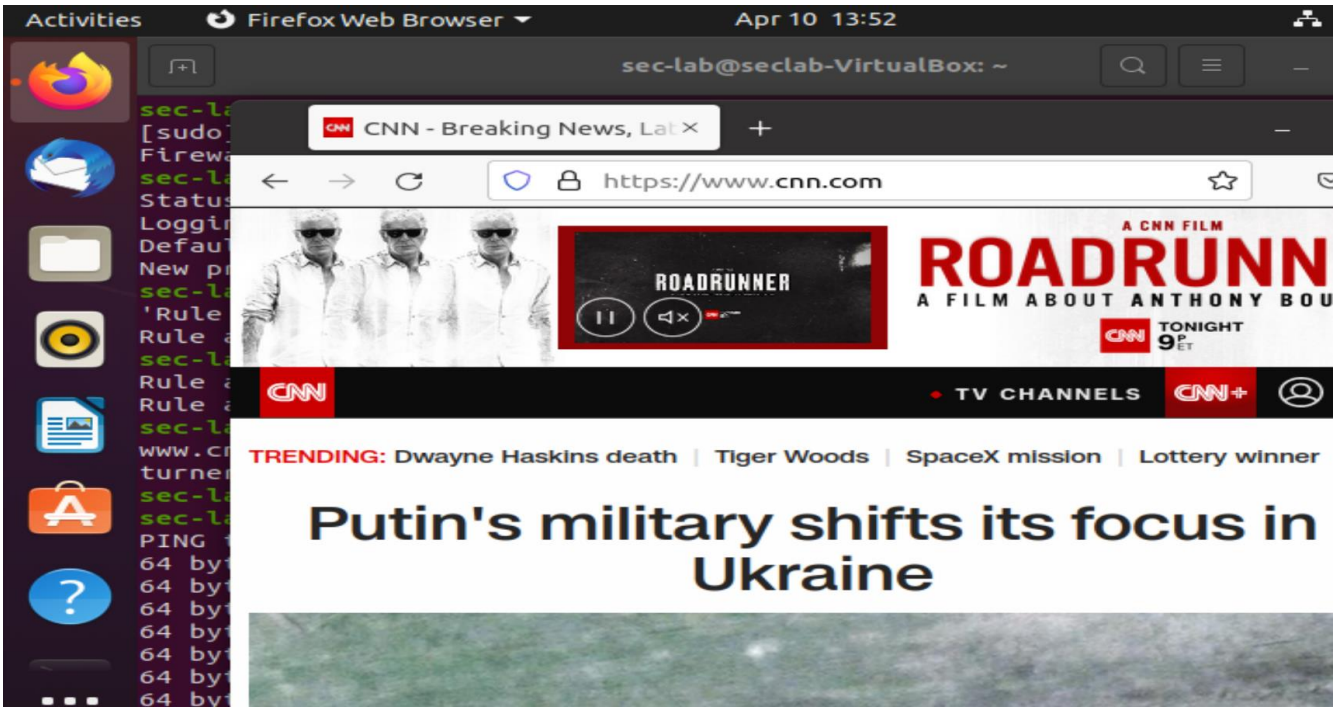
Q8: Attach a screenshot of the resulting browser window.

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



Now, we will run the following command:

sudo ufw deny out from any to 151.101.1.67

After that, you may not be able to access to the www.cnn.com webpage.

9. Try to access www.cnn.com in the browser.
10. If you still are able to connect, follow the steps mentioned in the note below.

Note: In order to completely deny access to the domain www.cnn.com, we need to cover all the IP addresses obtained using “host -t a www.cnn.com” along with alternative ip addresses mentioned earlier.

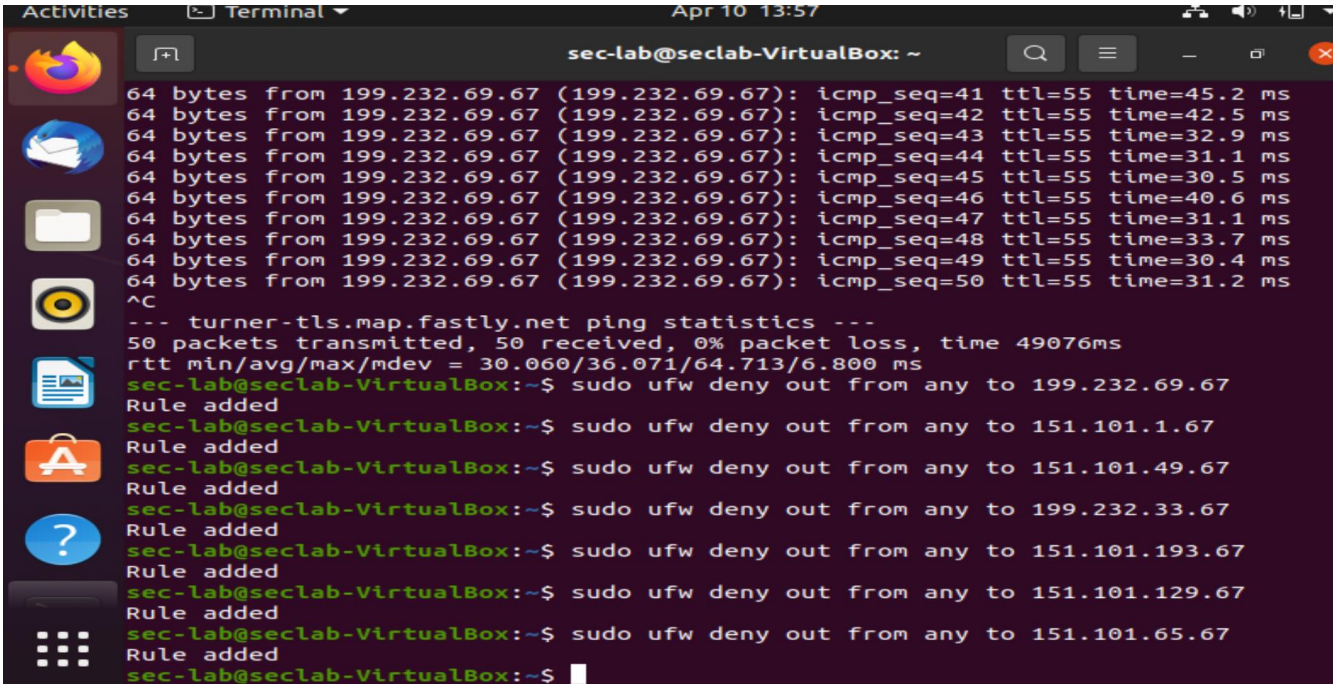
Q9: Attach a screenshot of the resulting browser window.

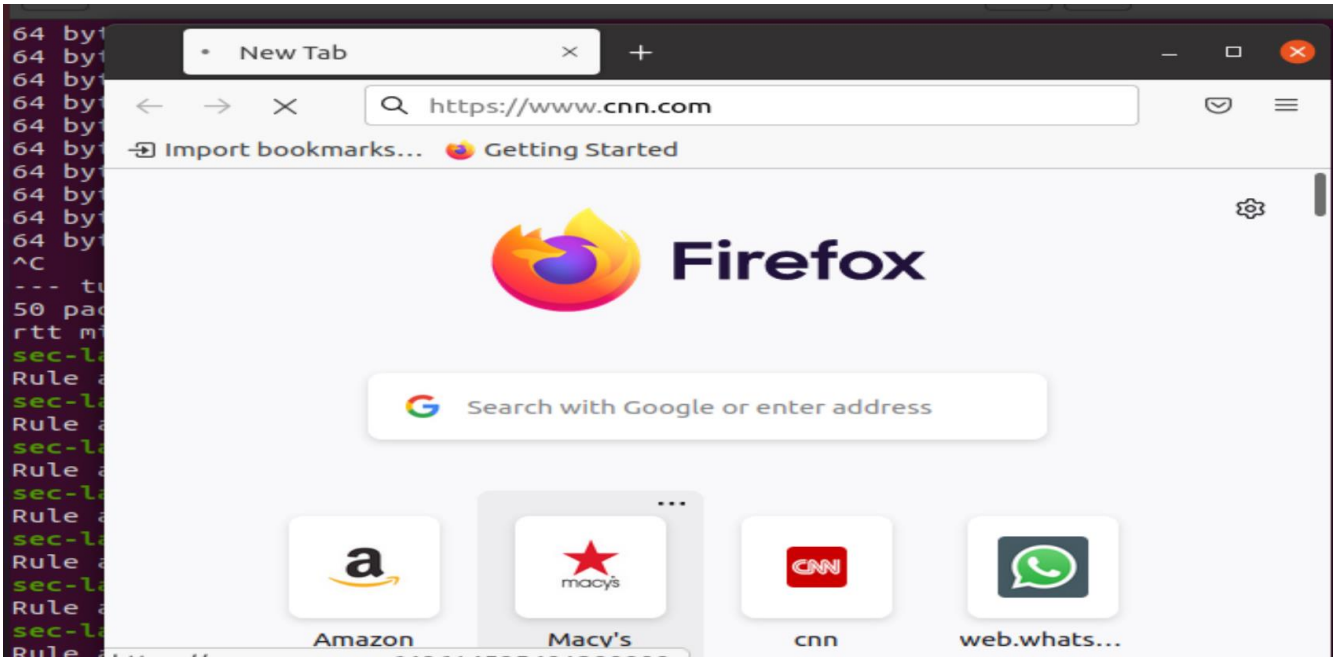
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab





11. Check the UFW status again:

    sudo ufw status verbose

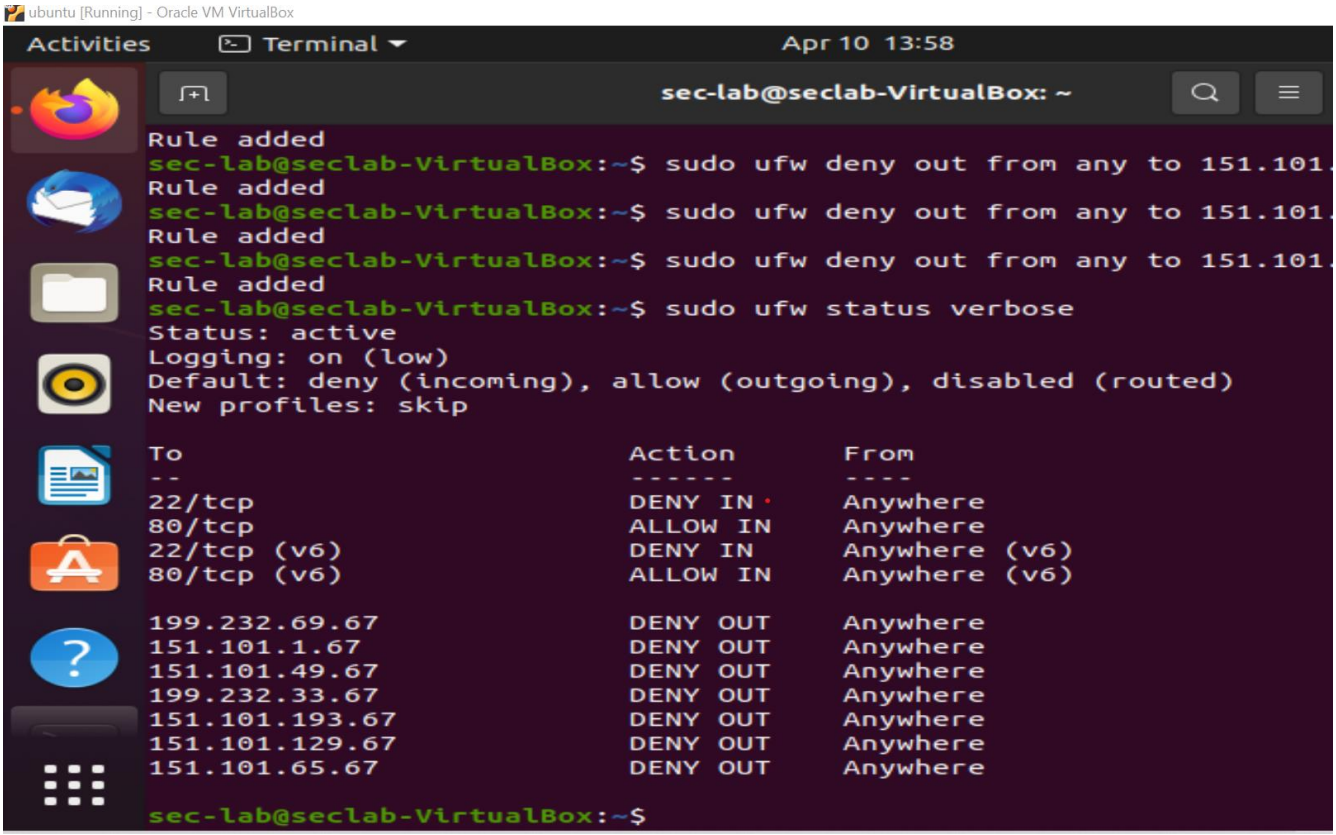
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

Q10: Attach a screenshot of the result.



12. Delete the last rule using the following command:

    sudo ufw delete deny out from any to 151.101.1.67

    Note: If you have added more than one rule for restricting access to www.cnn.com, then delete all of the rules related to accessing this website.

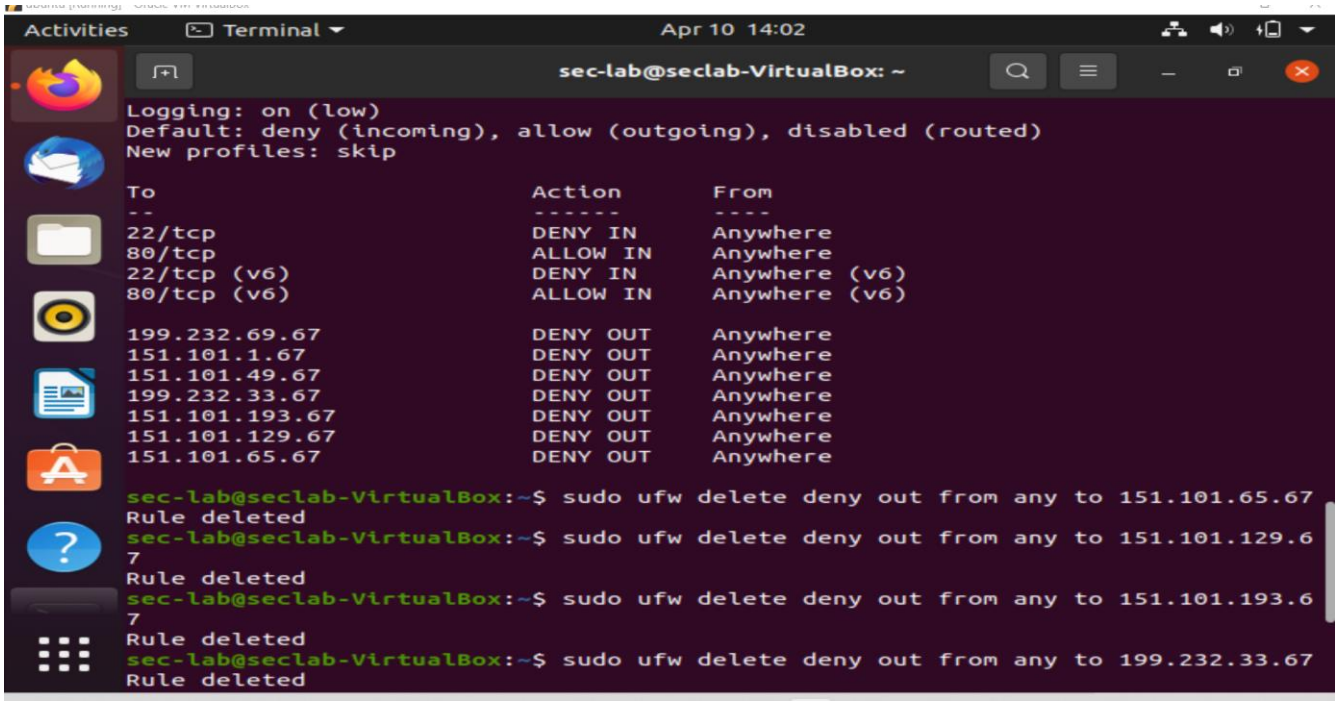
    Confirm that the access to the www.cnn.com page has been restored.

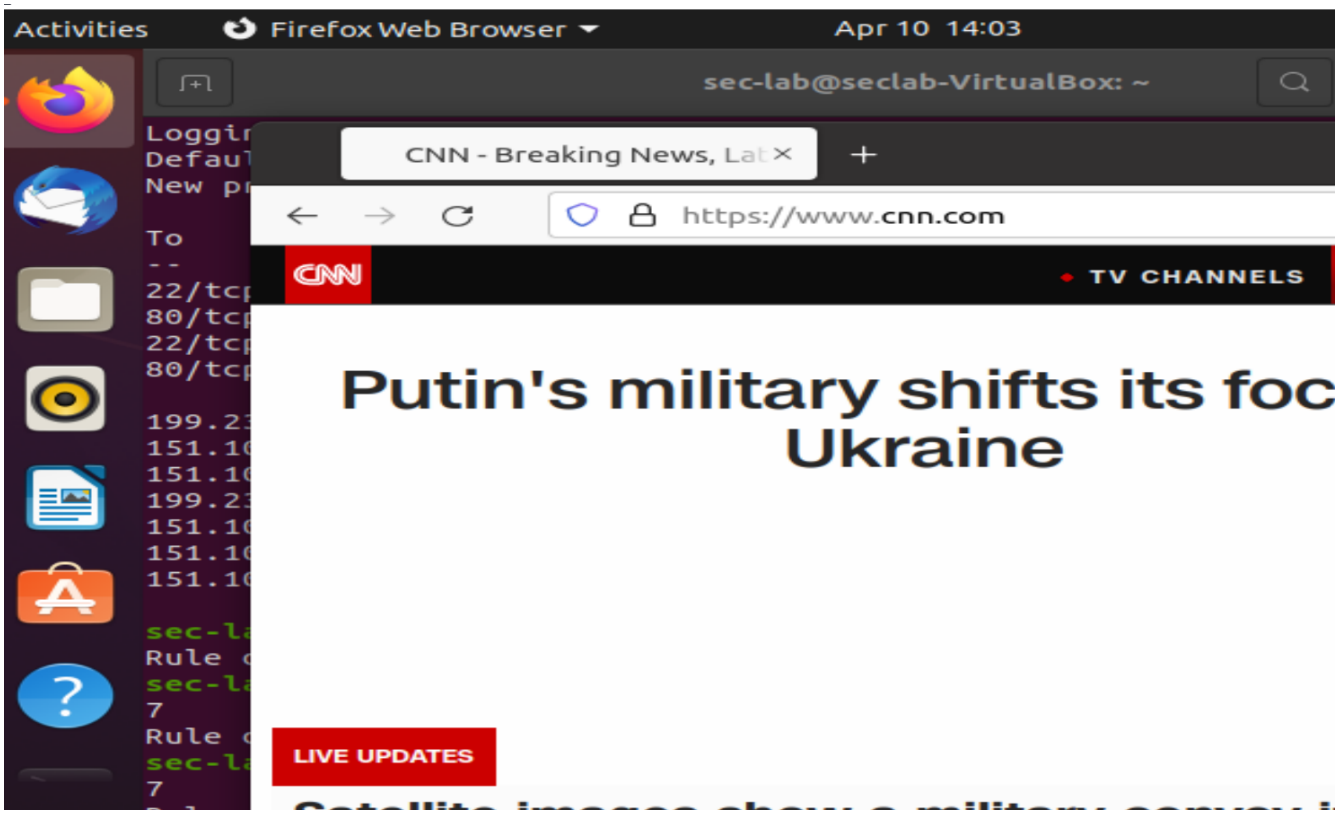
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab





Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

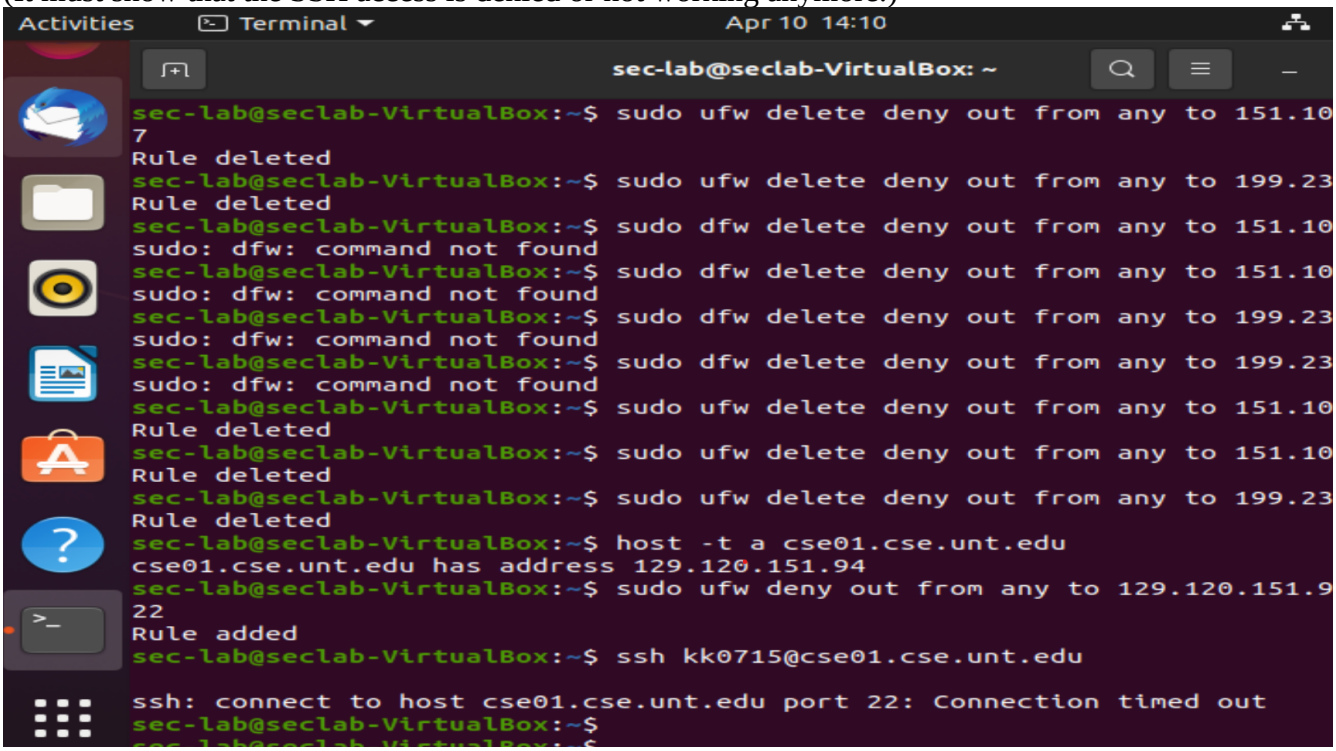
13. Now, we will block the access to the cse01.cse.unt.edu machine on the port 22 (SSH service). The command for adding this rule can be typed in the Ubuntu VM terminal as follows (you need to fill in the IP address and port number):

    sudo ufw deny out from any to _ipaddress_ port _port-number_

    To test whether the rule is working, in a terminal, try to establish a connection to cse01.cse.unt.edu server at port 22 (default SSH port). Remember that you need to use your EUID as a username for this connection.

    The corresponding command is: "ssh ***euid***@cse01.cse.unt.edu".

Q11: Attach a screenshot of the terminal showing that you cannot create a SSH connection.
(It must show that the SSH access is denied or not working anymore.)



14. Delete the above rule by typing:

    sudo ufw delete deny out from any to _ipaddress_ _sshport_

15. Now, try to establish an SSH connection to cse01.cse.unt.edu again.

Student Name:

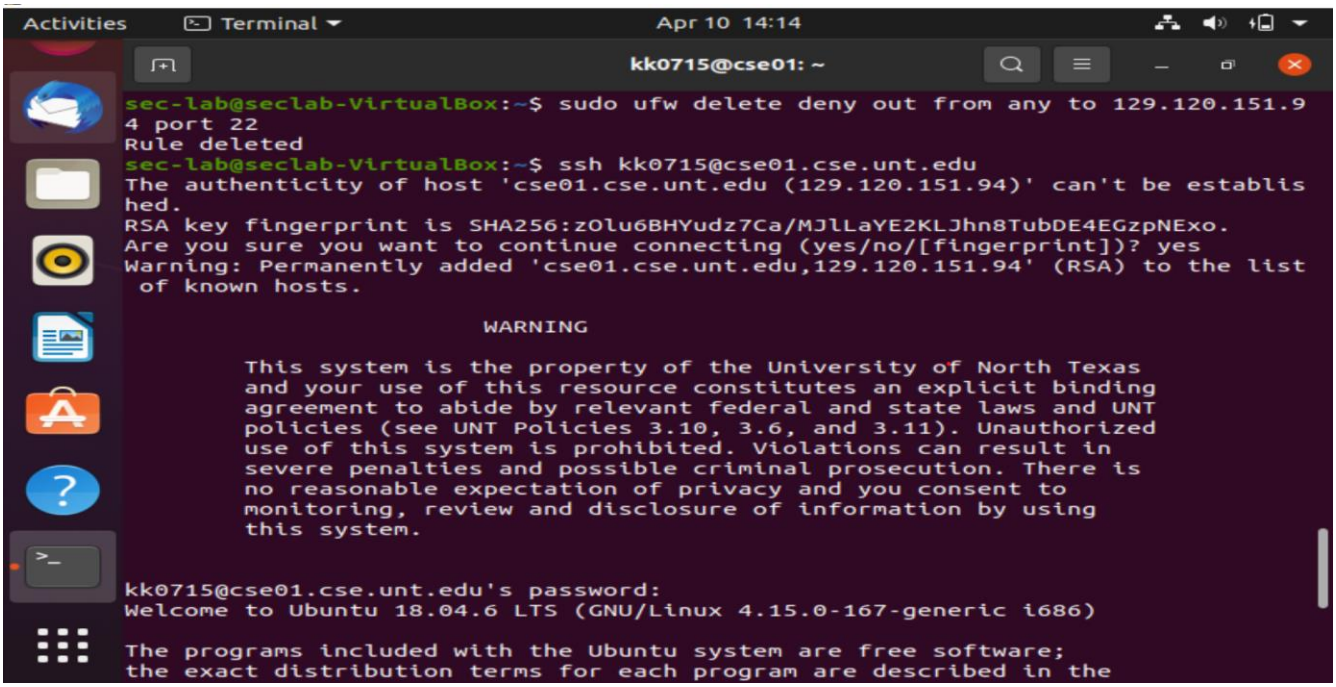
Course: CSCE 5550

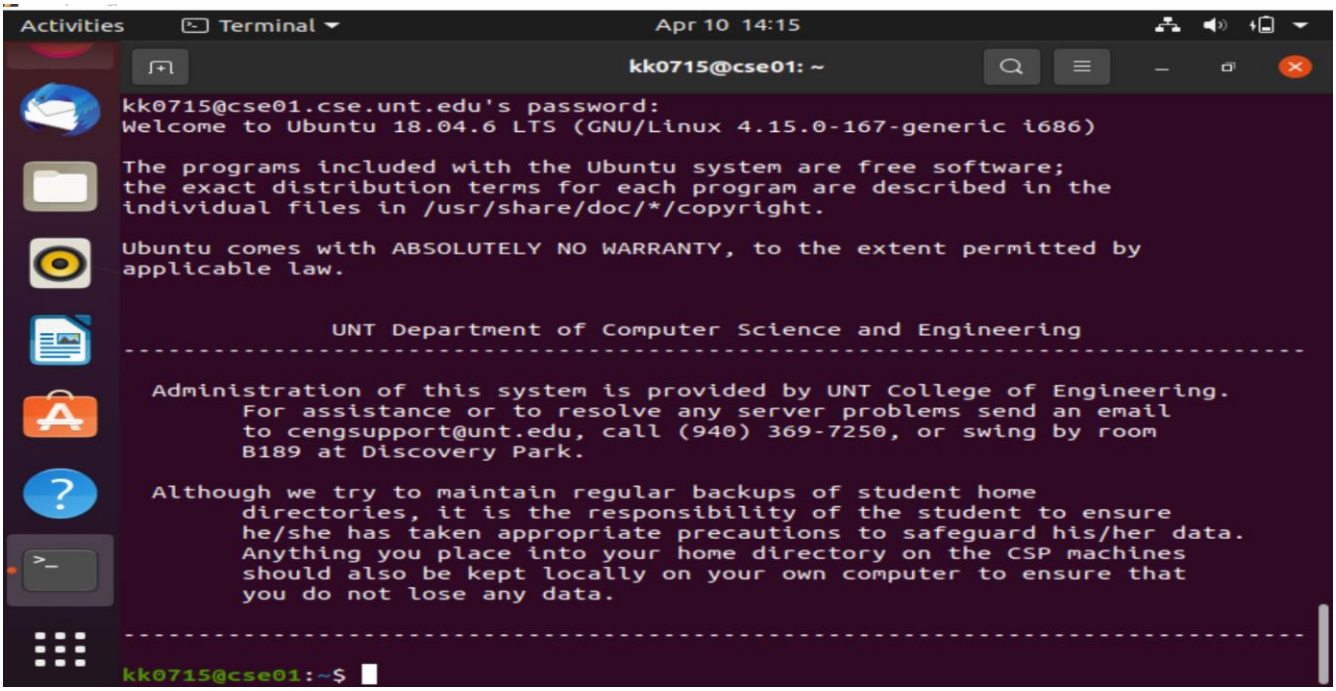
Semester: Fall 2021

Firewalls and IDS Lab

Q12: Were you able to establish a connection? Attach a screenshot of the result.

Yes, we were able to establish a connection correctly.





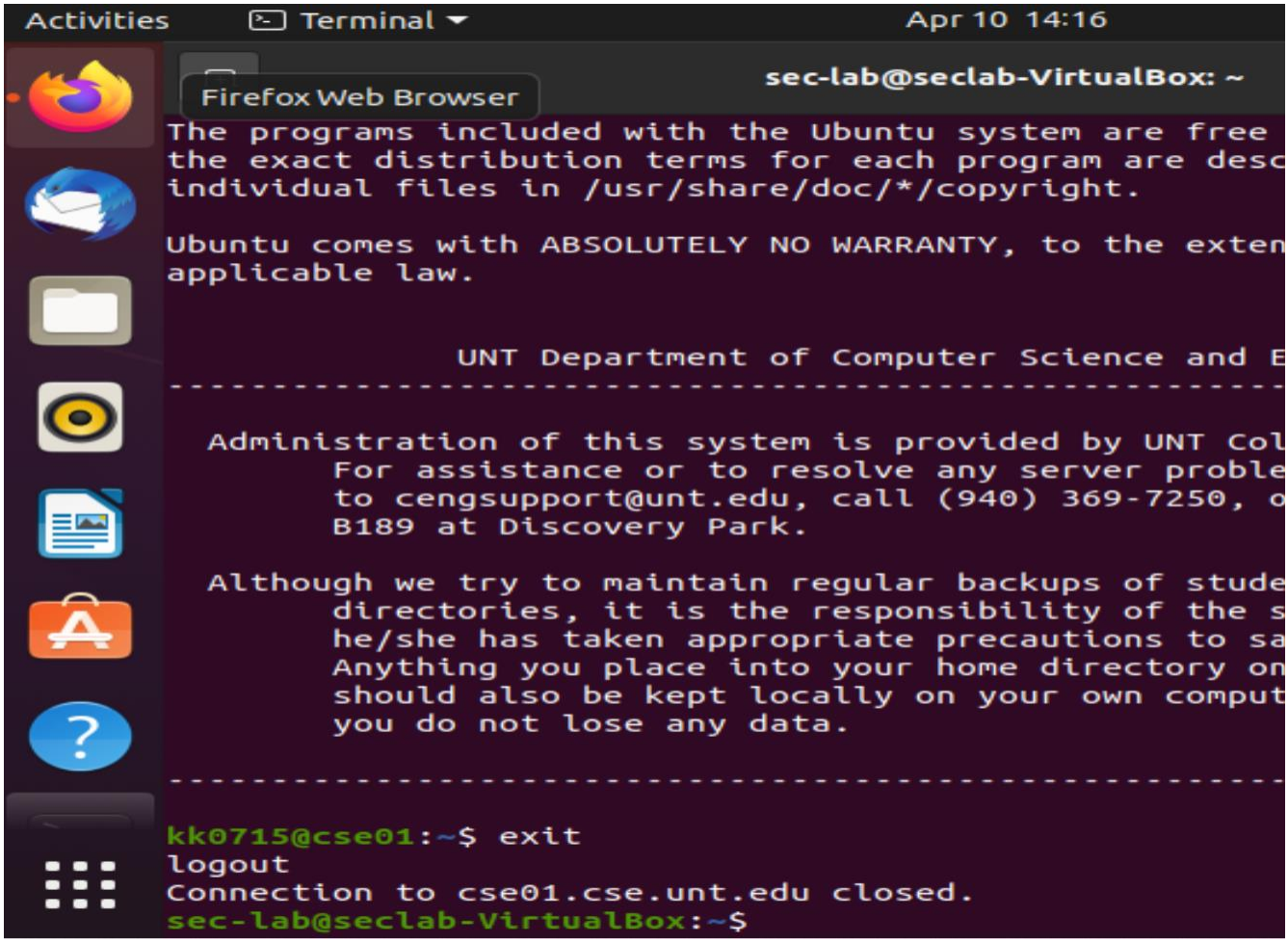
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

16. Terminate your session on CSE machine using the command “exit”, in case if you are logged in.



17. Reset UFW on your Ubuntu20 VM and disable it:

    sudo ufw reset

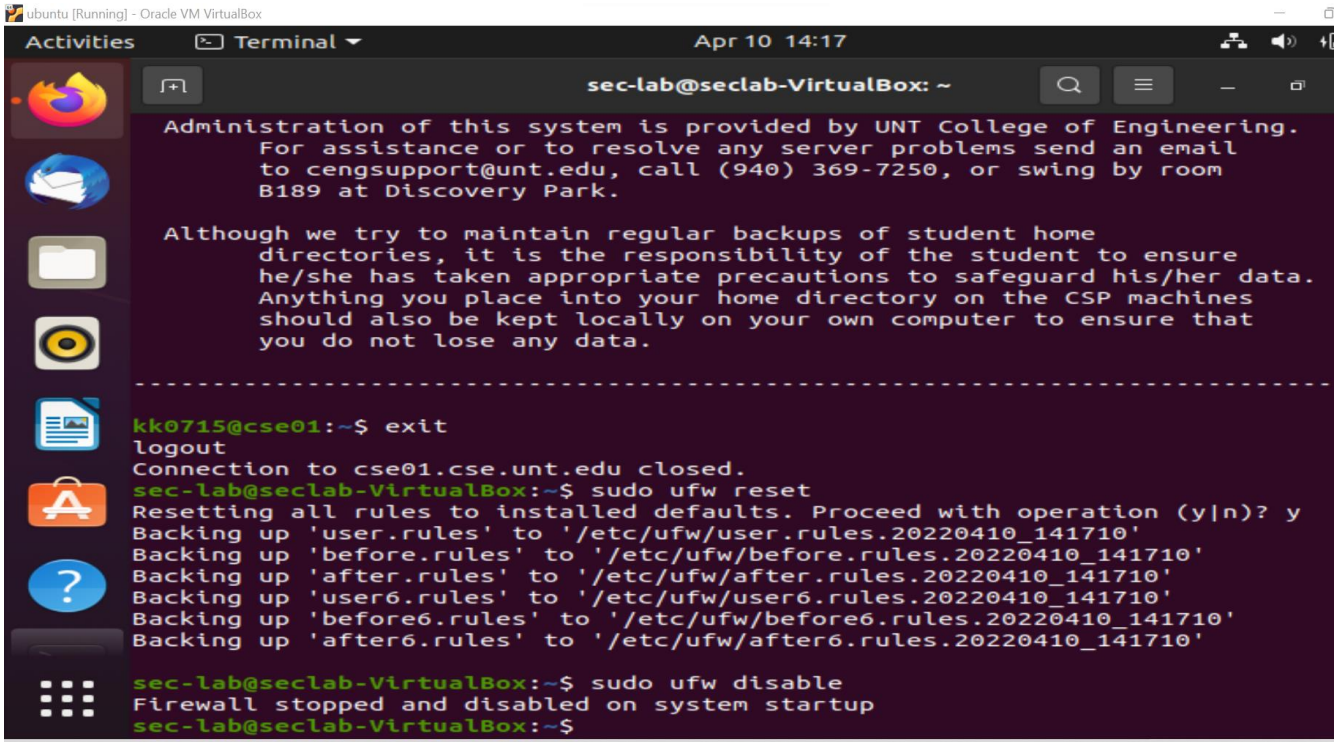
    sudo ufw disable

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



18. Power off the Ubuntu VM.

Section 2: Denial of Service (DoS) Attack – TCP SYN Flood

In this section, we will run the TCP SYN flooding attack against the Ubuntu VM (target) from the Kali VM (attacker). We will use the hping3 utility (www.hping.org), which a TCP/IP packet assembler/analyzer – it is commonly used for network testing and port scanning.

Note: The Kali 20 VM will be the same as in the previous labs.

First, let us configure the VMs so that they can communicate with each other. Note that the VMs must be powered off during this configuration process.

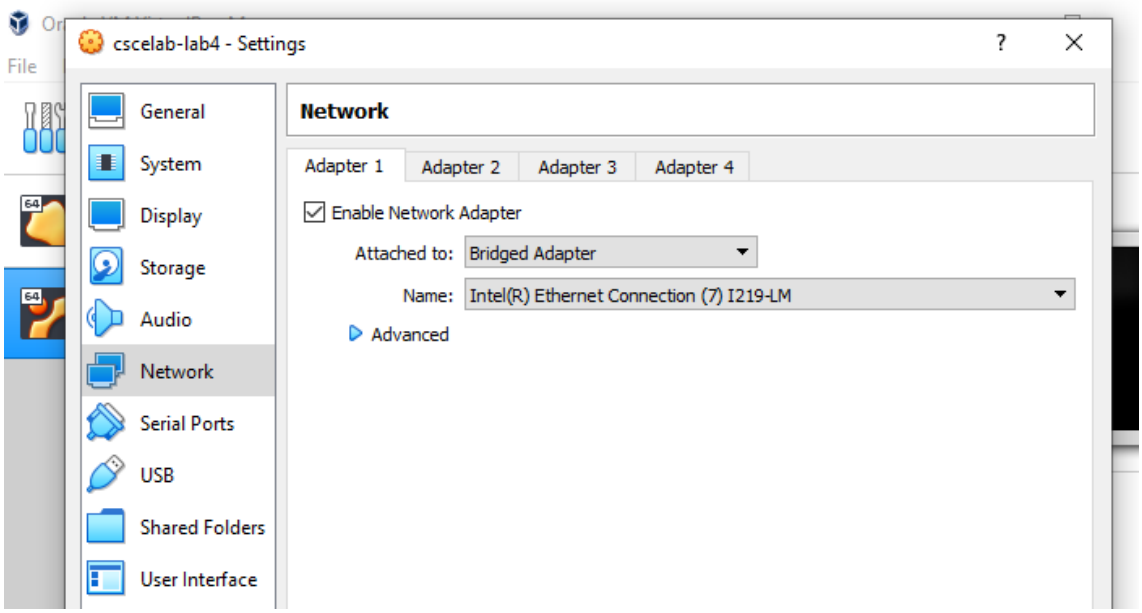
1. Use the following Bridged Adapter settings for the Ubuntu VM:

Student Name:

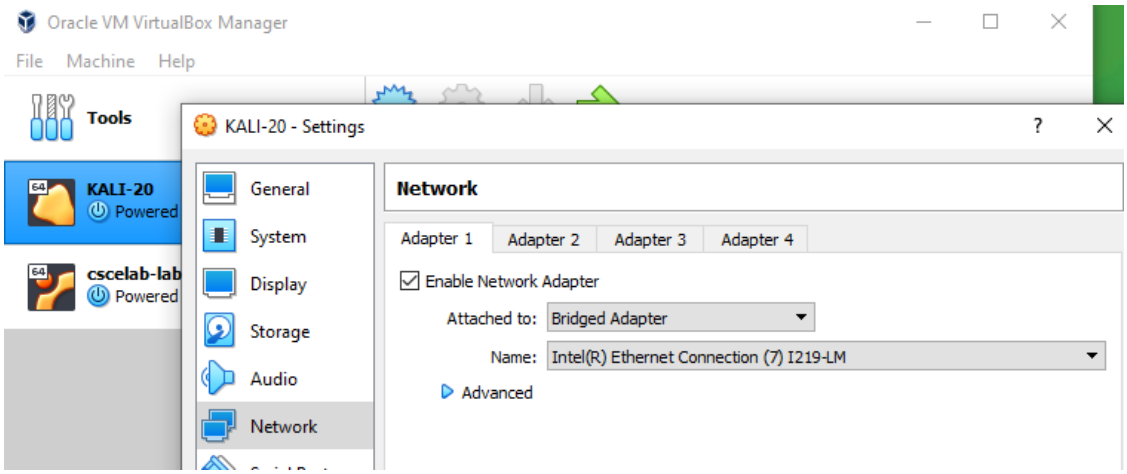
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



2. We recommend to "refresh" your Kali VM, i.e., to import it again from the respective OVA file. (Note that your previous work on that will be lost.)
3. Apply the following network adapter settings for the Kali VM:



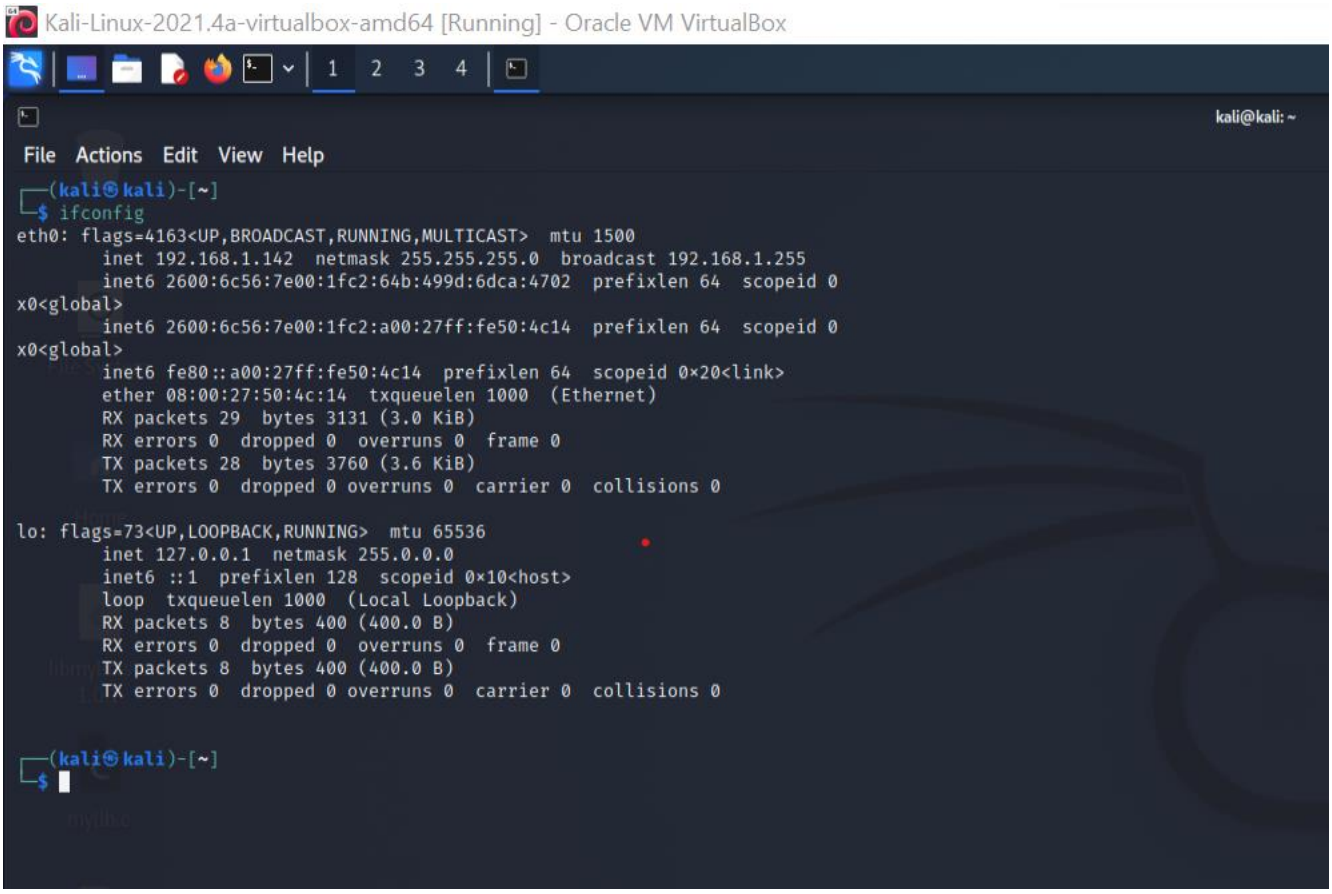
4. Start both the VMs and login with the respective credentials (Kali: osboxes/osboxes.org, Ubuntu: sec-lab/untccdc).
5. On Kali VM, start the Terminal and type "ifconfig". Notice the IP address and interface name(It may be different in your scenario):

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



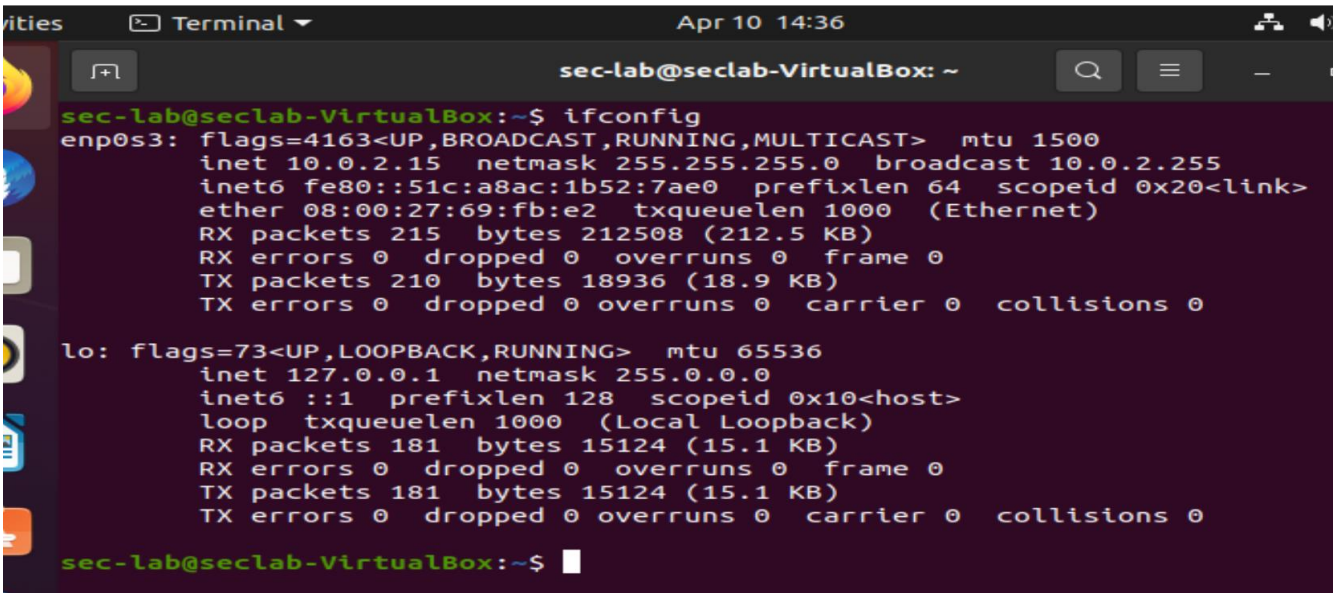
6. On Ubuntu VM, open the Terminal and type “ifconfig”. Notice the IP address and interface name (it may be different in your experiment):

Student Name:

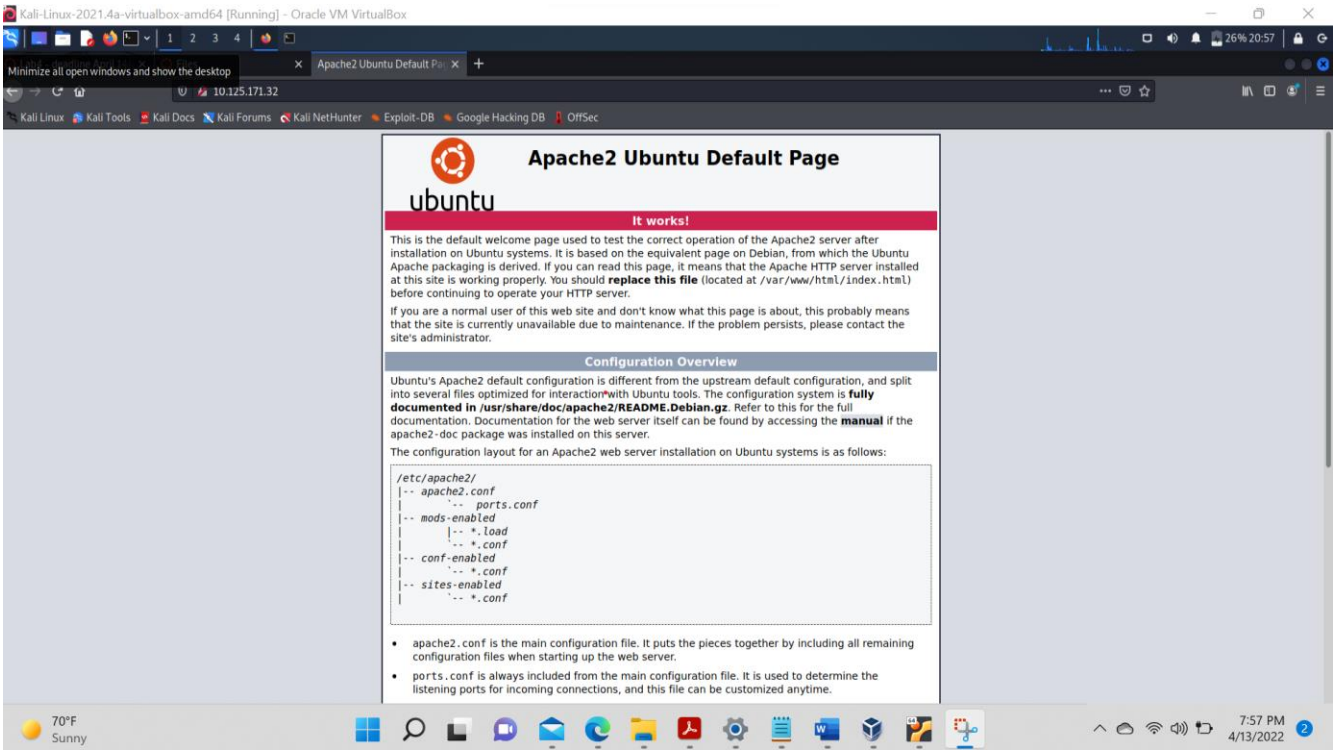
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



7. Next, we will access the Ubuntu VM from Kali VM via HTTP.
   Open the Firefox browser in Kali and type http://<ip address> as shown below.
   You may expect to see the page as shown below.



Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

8. From Kali VM, let us run an attack against the web server as follows:

   Open terminal and type below command

   sudo hping3 <web server ip address> -p 80 -S --flood --rand-source

Let us briefly discuss the parameters of the above command.

- -p 80 : since we would like to attack the web server on the Ubuntu VM, we set the HTTP port 80.
- -S : specifies SYN packets.
- --flood : the packets are send as fast as possible and replies are ignored.
- --rand-source : a random IP source address is set.

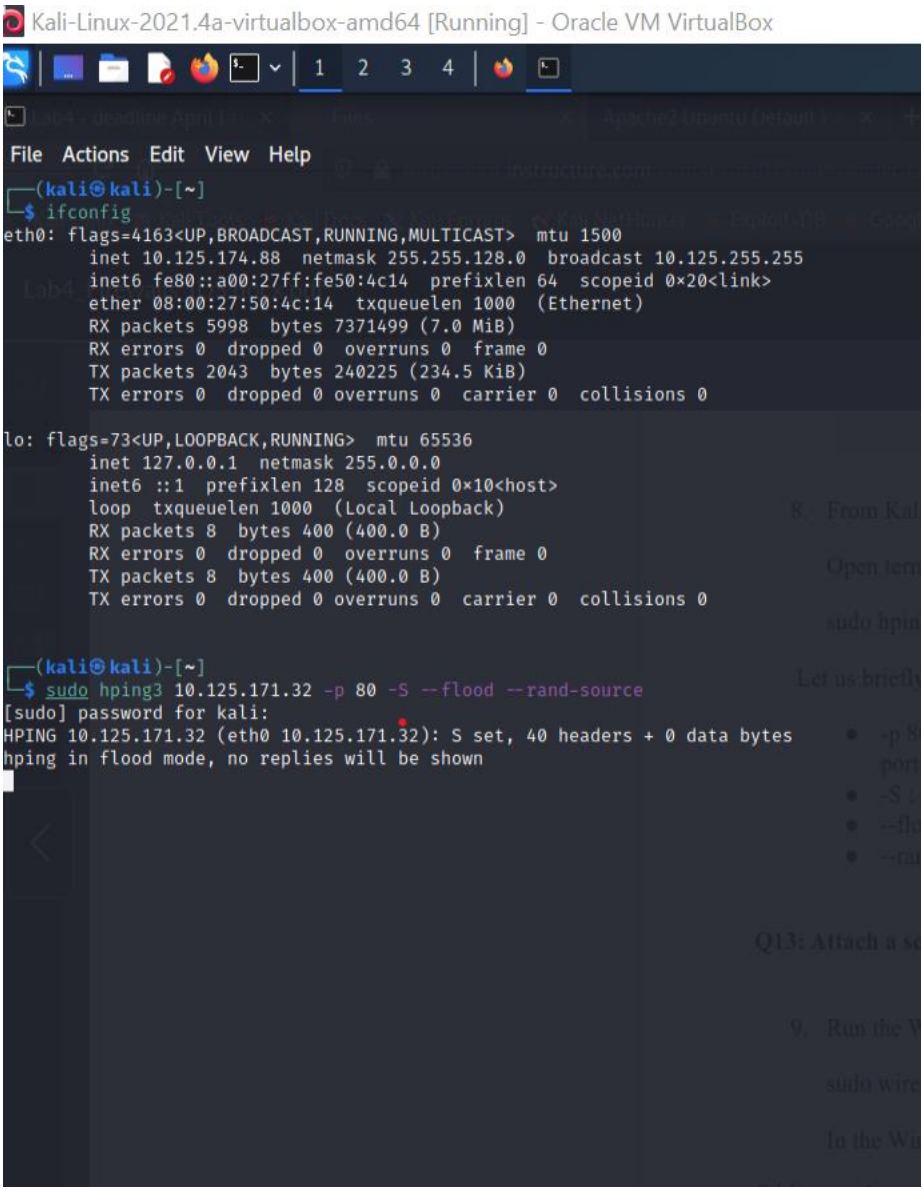
Q13: Attach a screenshot of the result.

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



9. Run the Wireshark on the Ubuntu VM:

   sudo wireshark

   In the Wireshark, start capturing packets.

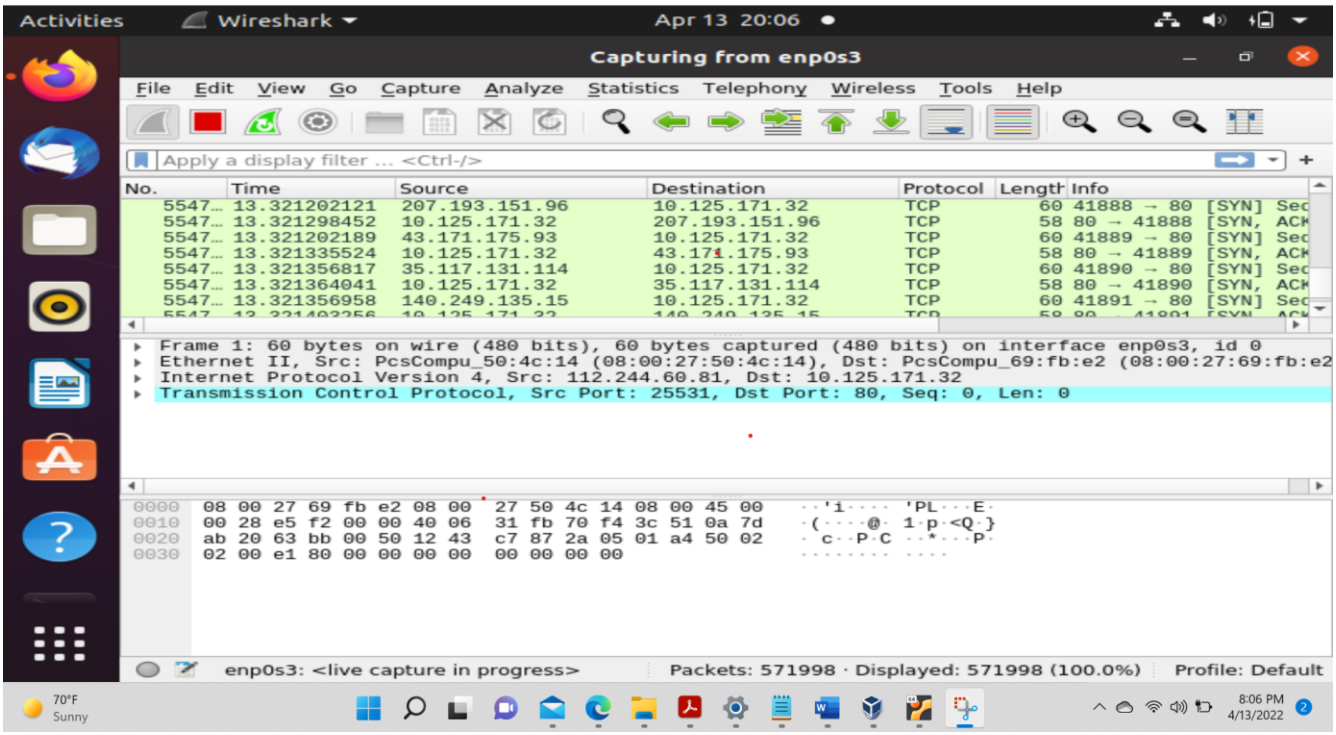
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

Q14: Attach a screenshot of the result.



10. Access the Ubuntu VM via browser with address (http://<Ubuntu VM IP address>) from your Kali VM – use a new private window in the browser. Then, refresh your browser window several times. Probably, you can still access the Ubuntu VM, which shows that a single instance of hping have not yet overloaded the web server. If this is the case, then run another instance of the hping utility (use a new terminal window) – for this, repeat the steps described in Item 8.
11. If you are still able to access the web server, keep adding the hping instances until the access to the Ubuntu VM via browser becomes very slow (or stops completely). Now, this would indicate that the VM is overloaded by your DoS attack.

    Note: Keep adding the hping instances until you have 10 of them running simultaneously. After that, proceed to the next step.

Q15: Report your observations from the above steps. In particular, how many *hping* instances did you have to run? Did it slow down the access to Ubuntu VM via browser?

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

We had run 10 hping instances at the same time. Yes, it did slow down the access to Ubuntu VM via browser, taking 4 to 5 seconds to load the page. The Ubuntu VM and the virtual box-hosting system is also not working efficiently. I unable to access wireshark or make any changes, not even on the screen.

12. Stop all the hping instances on Kali VM (press Ctrl+C for exiting an hping process). Make sure that you stop/kill all the hping instances. At this moment, your VM might be still overloaded, and you may not have access to it. If this is the case, rebooting it should solve the issue. Close Wireshark on Ubuntu.

Section 3: Intrusion Detection Systems (IDS) – Snort

IDS is software/hardware system which monitors network traffic for attacks or policy violations. The policies are defined as sets of rules. When their violation is detected, IDS typically alert administrators and/or automatically deploy certain pre-set countermeasures. IDS are typically used inside the local network as the next line of defense after firewalls.

In this section, we will use Snort IDS (https://www.snort.org) to detect the TCP SYN flood attack. Again, the attacker will use the Kali VM and the Ubuntu VM will be a target.

13. Let us first install and configure Snort on the Ubuntu VM. (Note that Snort comes with some default rulesets.) Run the following command:

    sudo apt-get install snort

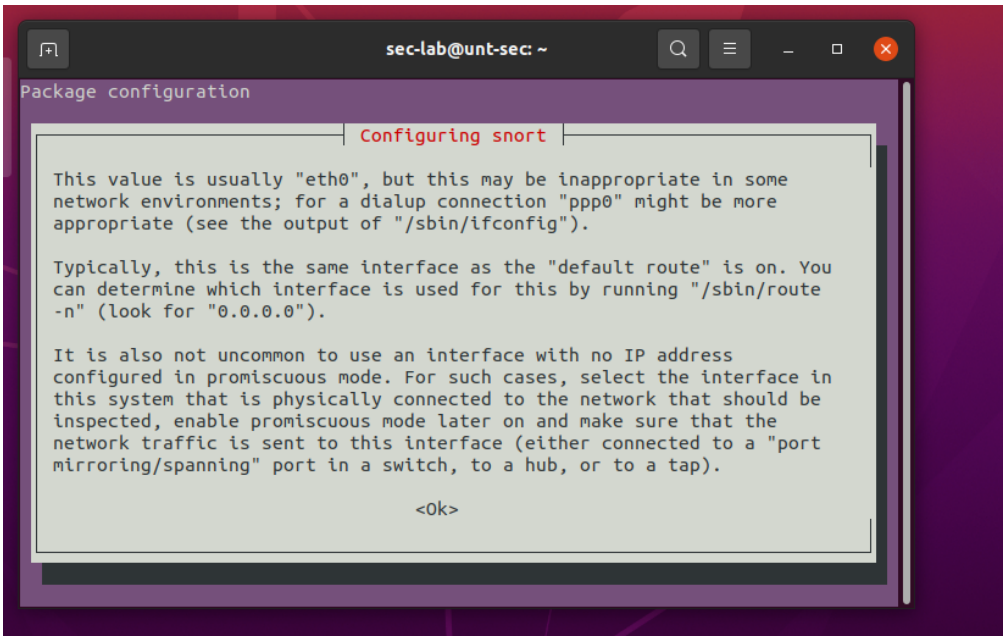
    You may expect to see notification similar that shown below.

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



14. Press Tab on the keyboard to move your selection to <Ok> and press Enter.

    During the installation, you will need to specify your network information. Input your interface name as shown below. In our case, it was “enp0s3”. You can get your network information using the command “ifconfig”, as shown earlier.

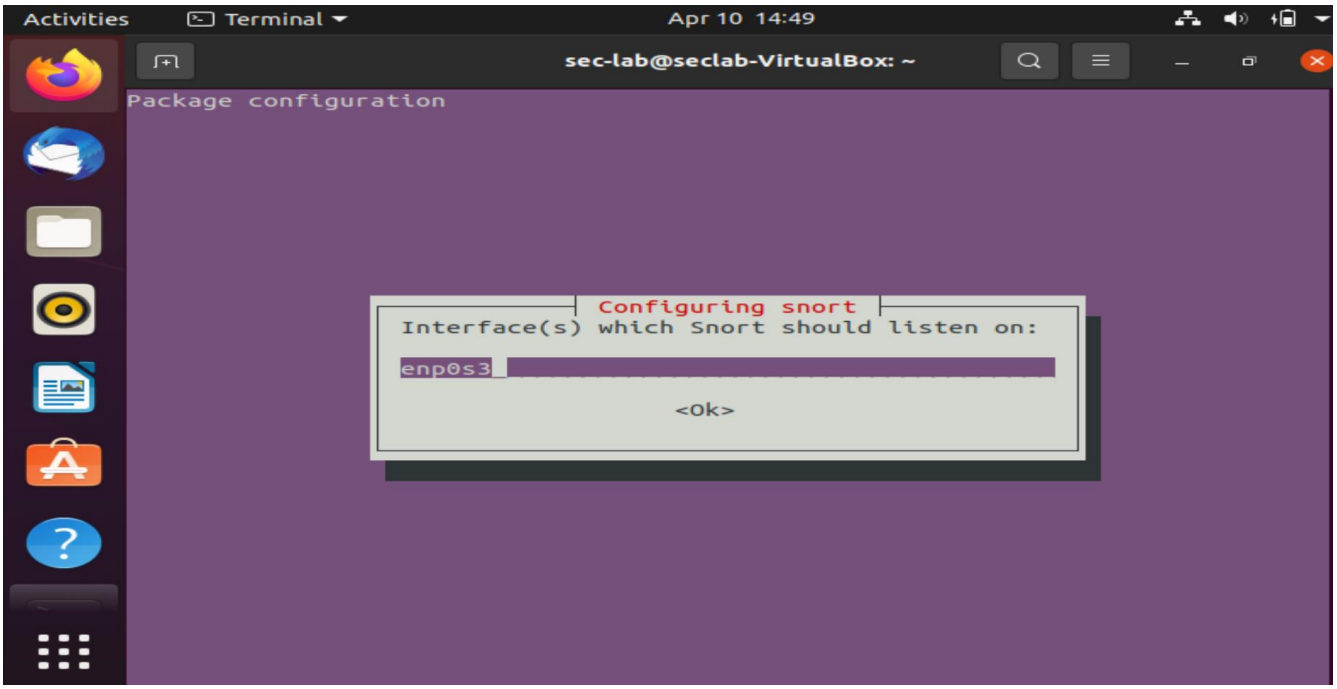
    Note: If the terminal shows that ifconfig command is missing, then install the package *net-tools* using the command “sudo apt-get install net-tools”.

Student Name:

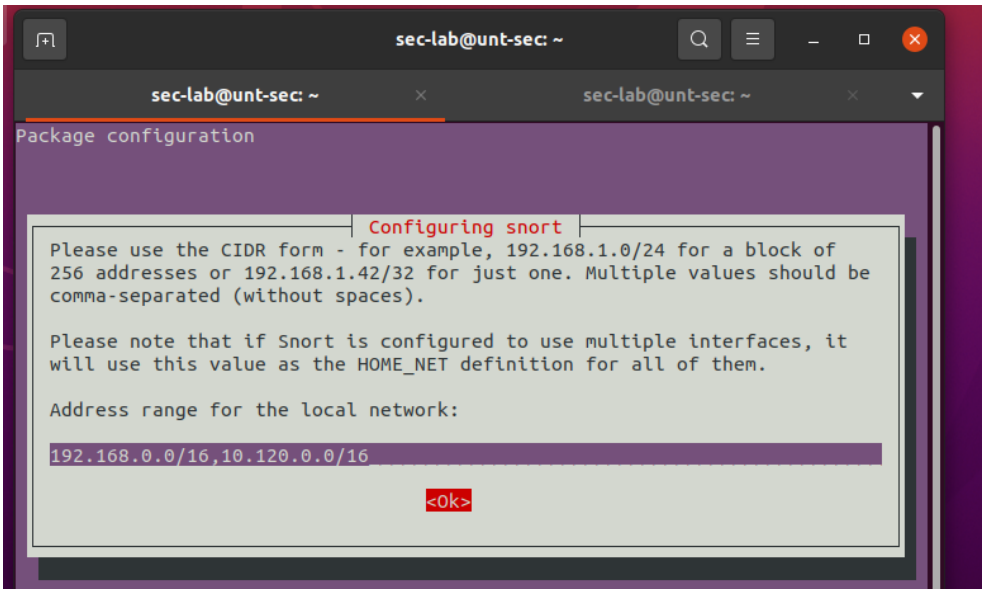
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



15. Add network 10.120.0.0/16 to the next page with a comma as shown below.



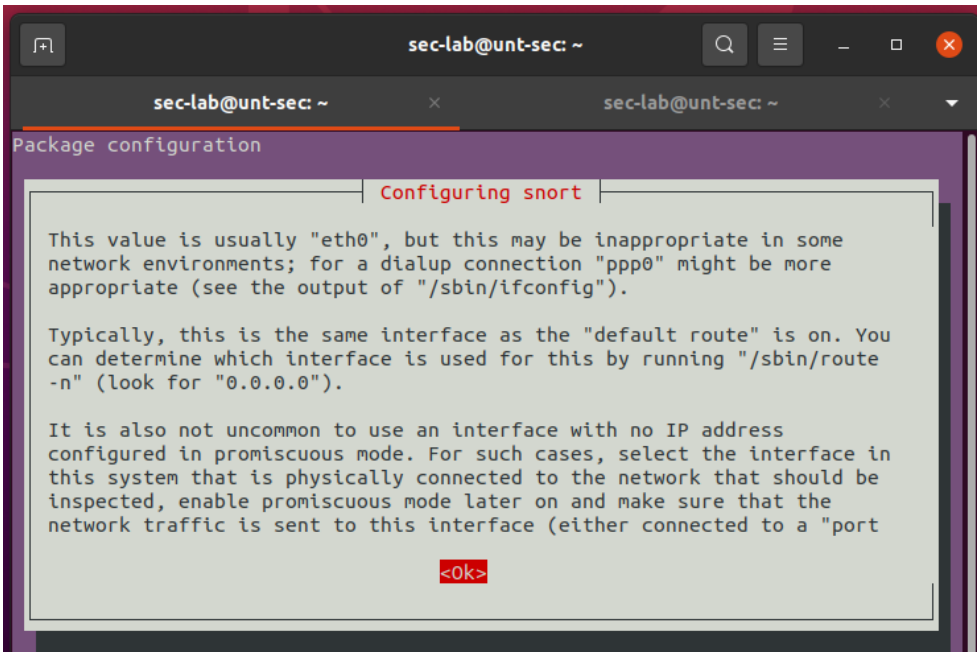
Press Tab and then press Enter in the next screen.

Student Name:

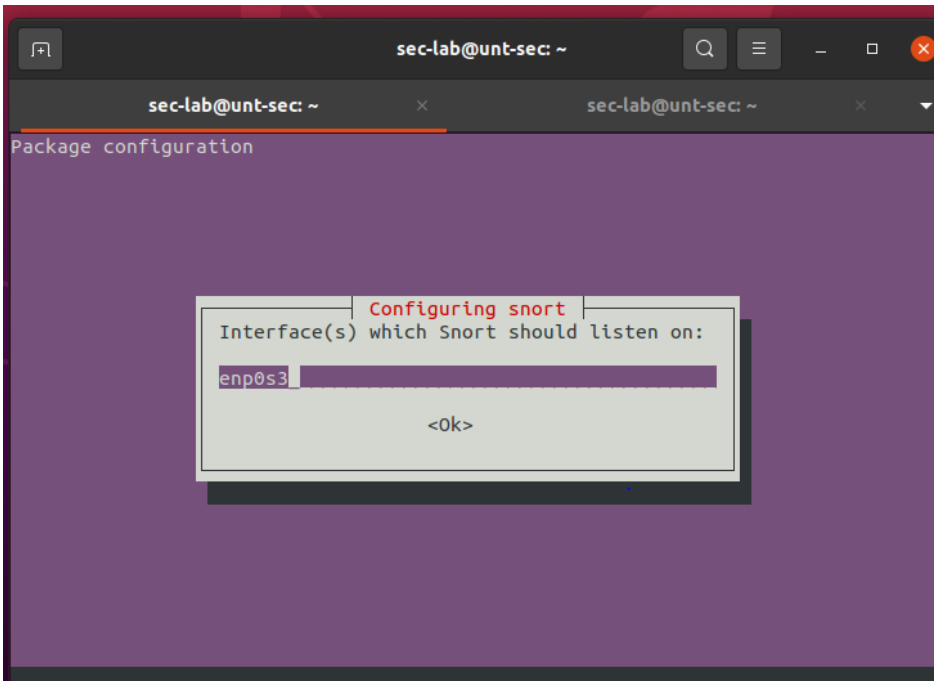
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



16. On the next page, select <Ok> by pressing Tab and then Enter.



After that, your installation will be complete.

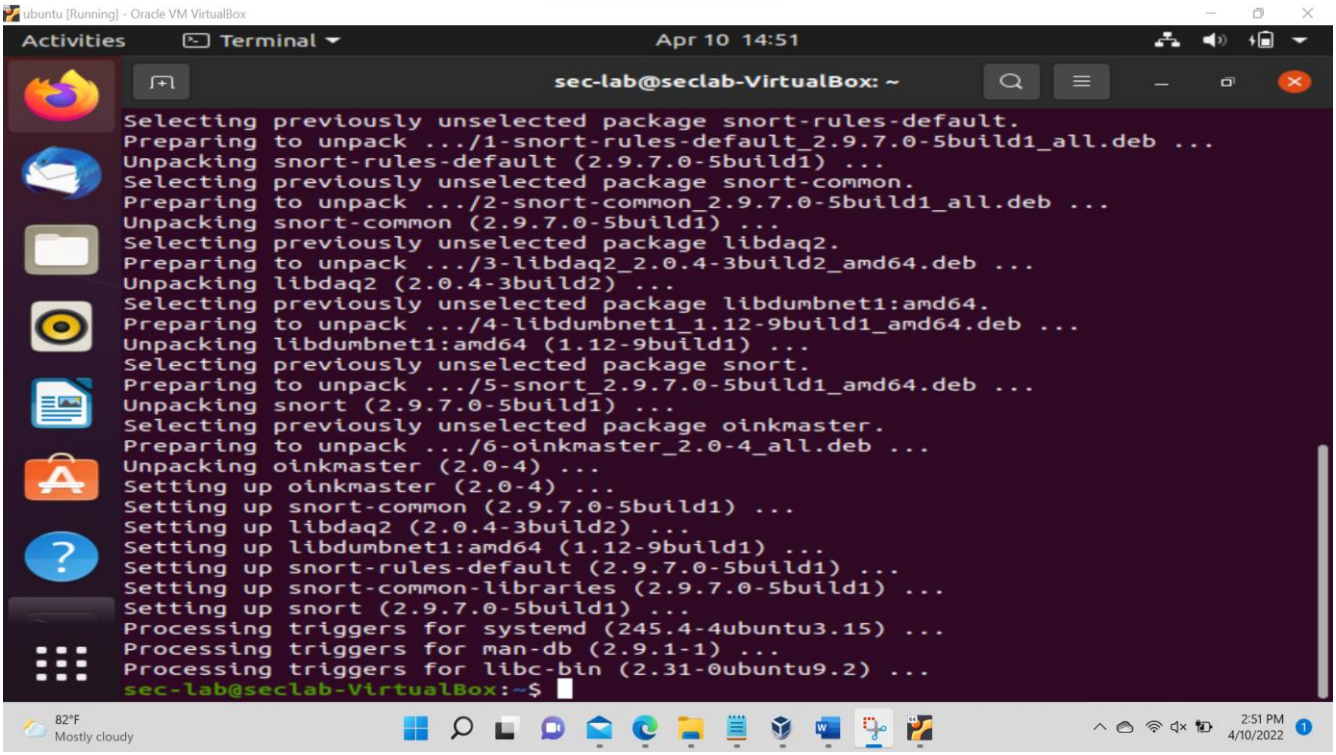
Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

Q16: Attach a screenshot of the result. (Only one screenshot showing the terminal after the installation has been completed is enough.)



17. Next, we will slightly change the default output format of Snort to make it convenient for the purpose of this lab.

    Edit the file */etc/snort/snort.debian.conf* with the root privilege (use "sudo") and update the line starting with *DEBIAN_SNORT_OPTIONS* with the following:

    DEBIAN_SNORT_OPTIONS=" -A full "

18. Now, let us add a custom rule for the TCP SYN flood detection. For that, we will add the following rule in the file */etc/snort/rules/dos.rules* (as a new line at the end) using the root privilege:

    alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible TCP SYN flood"; detection_filter: track by_dst, count 50, seconds 10; sid:1000001;)

    - The first six parameters of the rule require to generate an alert for a TCP connection from any source IP address and port to any IP address in the protected network, port 80 (http).

Student Name:

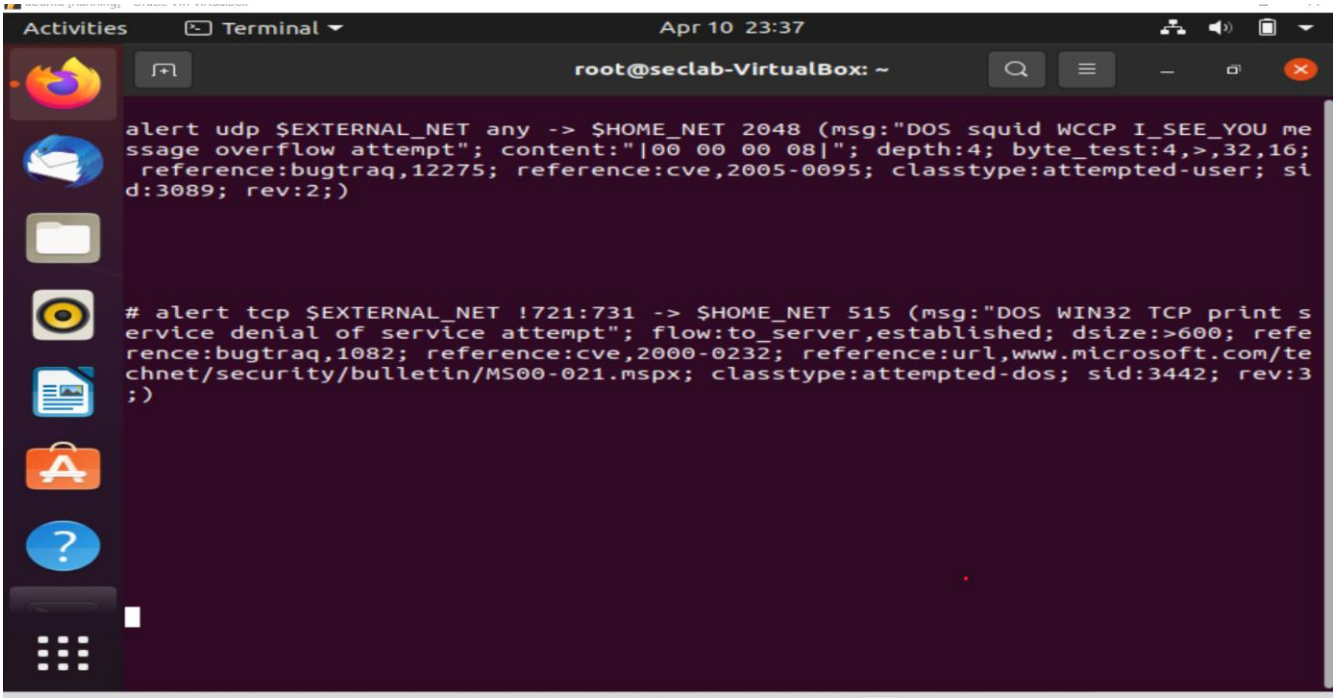
Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab

- The following conditions, which are specified in the brackets, need to be met: the packet must be SYN, and the following rate must be exceeded (counted by the destination IP): within each 10-second frame, after 50 packets which satisfy the conditions, each next packet will trigger an alert. Note that these condition define what frequency of packets will be considered as attack – this frequency must be substantially higher compared to what you normally expect for your network.
- The last parameter is a Snort identifier of your new rule. (We use this number because the SID's before 1000000 are reserved by Snort.)

Q17: Attach a screenshot of the updated *dos.rules* file.

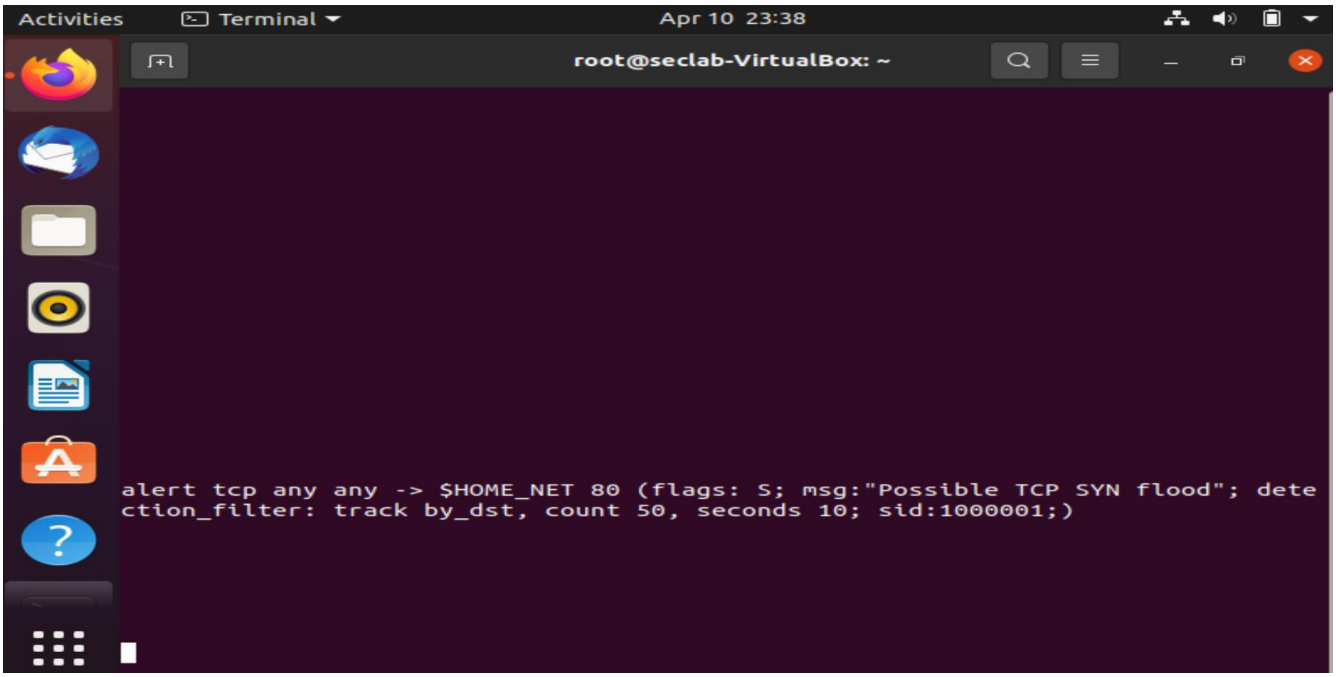


Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



19. After you updated the *dos.rules* file, restart Snort by typing:

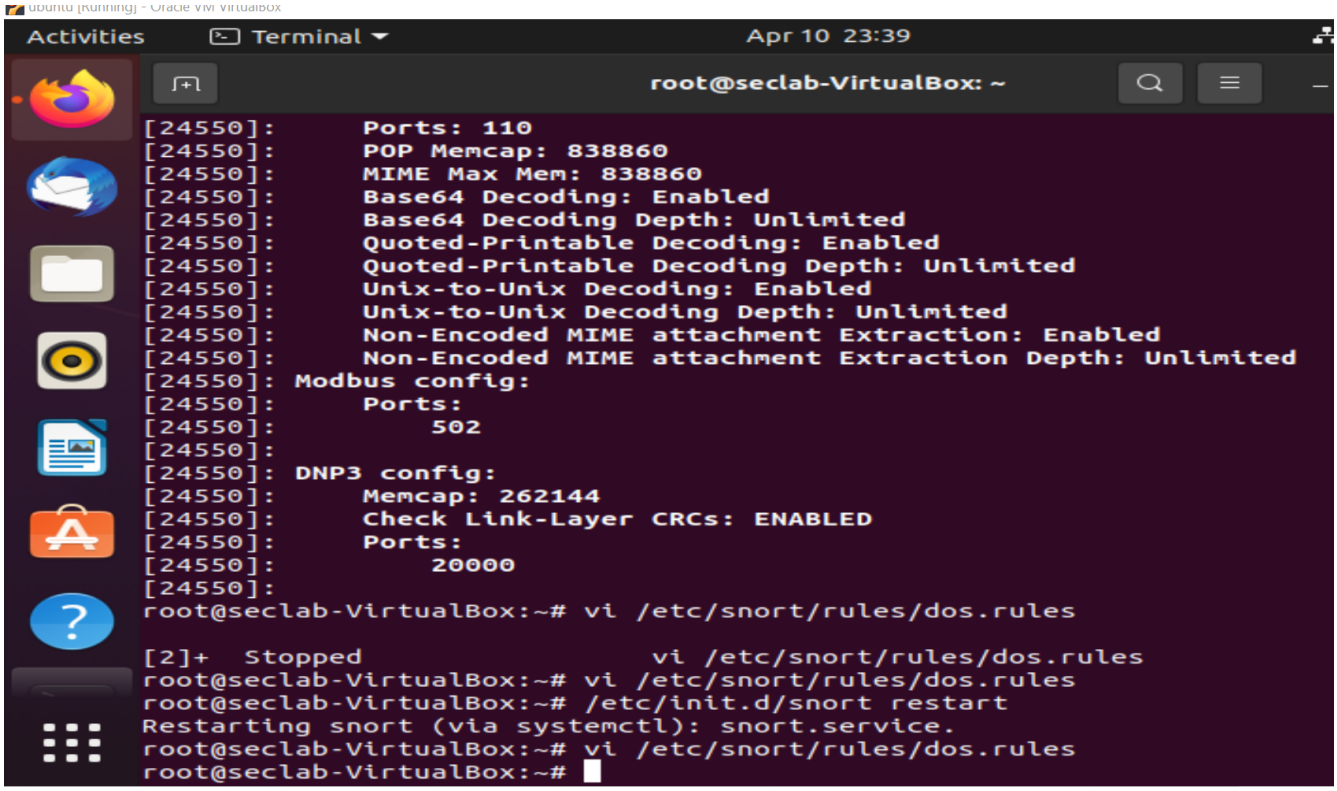
    /etc/init.d/snort restart

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



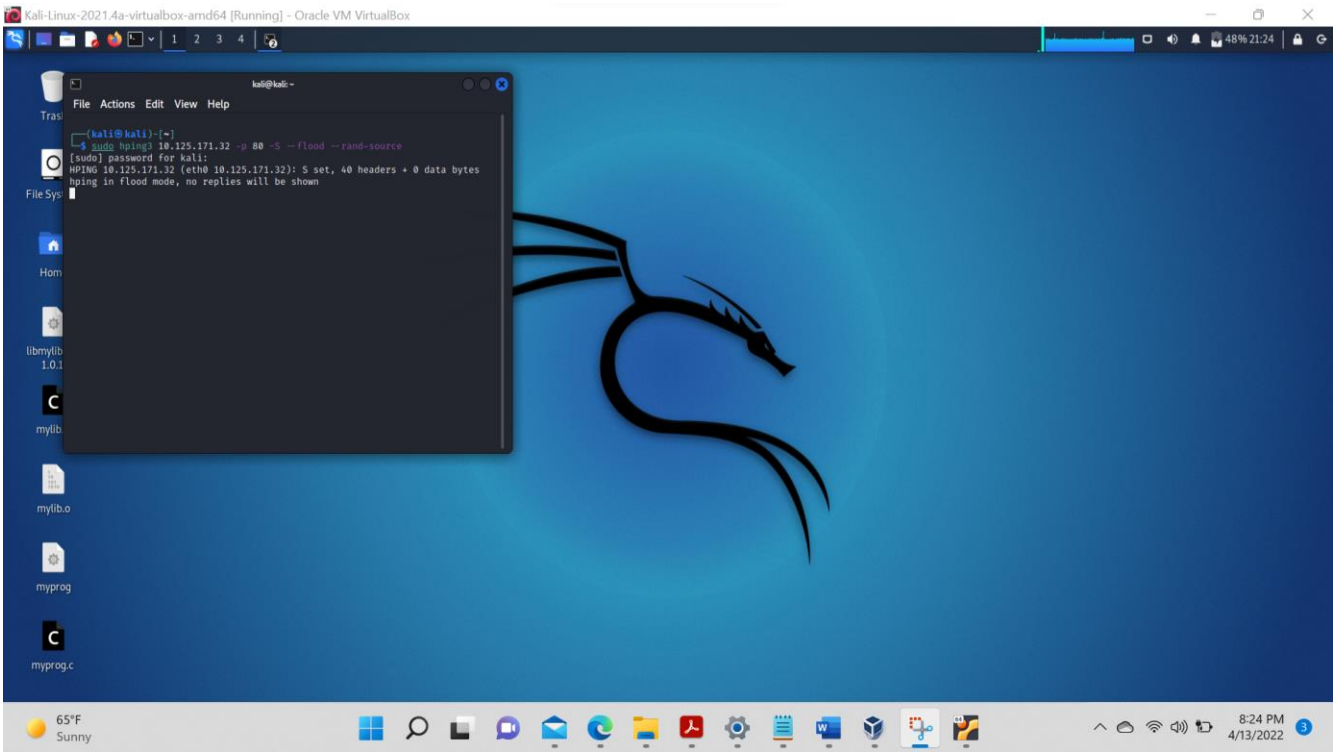
20. Next, run the attack described in Section 2 again from Kali VM (a single hping instance is sufficient).

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



21. Monitor the Snort log file, using the following command:

    tail -f /var/log/snort/alert

    You should be able to see the output with your own rule's sid (1000001).

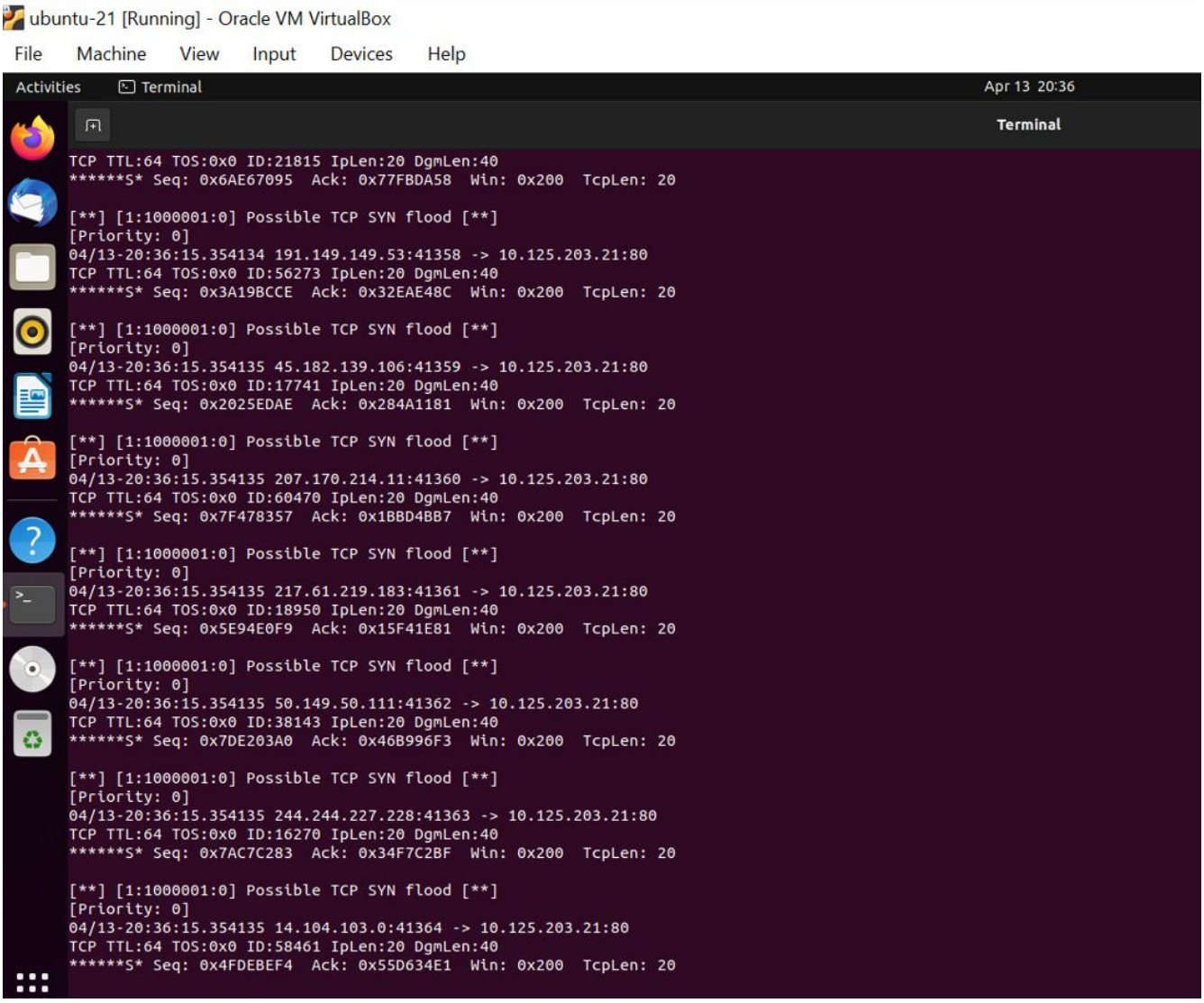
Q18: Attach a screenshot of the result. Explain what you see.

Student Name:

Course: CSCE 5550

Semester: Fall 2021

Firewalls and IDS Lab



A rule was set to allow snort to detect flooding requests from any server. Hence, we can see TCP flooding requests in the snort alerts file of Kali VM.