IMPORTANT NOTE:
Students found guilty of tampering with the computers’ configurations will receive an ‘F’ grade in the course. Only run the commands which are necessary for completing the lab session.

**NMap Network Scanning & Firewall Rules**

*Maximum Points = 14*

**Setup for this lab:**
Ubuntu workstations, Terminal, Nmap, ufw, ssh

*Login info: Username – cse321, password – c$e321*
*Hotkey to open Terminal: Ctrl+Alt+t*
*To select an area for screenshot on Ubuntu: Shift+prtscn*

1. In this lab, you will work with a partner and have one of your machines target the other. We will refer to the two machines as A and B. A will attack B and B will stop the attack.

   A will run NMap against the target B in order to scan for vulnerabilities. B will use ufw commands to setup a firewall that stops the attack.

2. Nmap ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. By using scanners such as Nmap, the attacker is able to sweep networks and look for vulnerable targets.

3. UFW stands for Uncomplicated Firewall. It simplifies the process of setting up firewall rules by providing a frontend interface for iptables.

**SUBMISSION:**
- Mention the ip addresses of A and B
- The document should contain screenshots for each task with screenshots for B followed by screenshots for A. No explanations are needed.

Submission will be done through Blackboard, deadline is tonight 11:59 pm.
PART 1: Familiarity with Nmap and ufw commands

Before starting, run the following commands on both A and B.
- In the terminal, get root access. `sudo bash` and enter password: c$e321
- Reset the firewall to default settings.
  `root@laptop0-0:~# ufw reset`
- Check the status of the firewall. It should show Inactive.
  `root@laptop0-0:~# ufw status`
- Get the IP address of A and B. These will be required when writing commands.
  `root@laptop0-0:~# ifconfig`
  Choose the ‘inet addr’ under ‘ens5’

On B, check if ssh service is up and running,
- `root@laptop0-0:~# service ssh status`
  If the status is inactive then restart ssh service on B
  `root@laptop0-0:~# service ssh restart`

**Task 1: Checking for Open Connections**

[1pt] B will check for active connections by running.
- `root@laptop0-0:~# netstat -ntlp | grep LISTEN`

[1pt] Meanwhile, launch Nmap on A and check which ports are open on B.
- `root@laptop0-0:~# nmap <IP_of_B>`

We used `netstat` to scan the ports of B locally. We used `nmap` to scan the ports of B remotely from A. As the firewall is inactive, A should be able to see all open ports on B.

-- Get the Screenshots of commands and output of A and B for Task 1 --

**Task 2: Enable the Firewall**

[1pt] Enable the firewall on B.
- `root@laptop0-0:~# ufw enable`

Check the status of the firewall. It should show Active.
- `root@laptop0-0:~# ufw status`

[1pt] Launch Nmap on A and check which ports are open B.
- `root@laptop0-0:~# nmap <IP_of_B>`

As the firewall is now active, A should not be able to see any open ports on B.

-- Get the Screenshots of commands and output of A and B for Task 2 --
Before you proceed to task 3., reset and enable the firewall on B.
- root@laptop0-0:~# ufw reset
- root@laptop0-0:~# ufw enable

**Task 3: Firewall allow rule for specific IP and port**

**[1pt]** Write a firewall rule on B to allow only A to see the port 22.
- root@laptop0-0:~# ufw allow from <IP_of_A> to any port 22

View the added rule.
- root@laptop0-0:~# ufw status

**[1pt]** A will run an NMap to scan available ports on B by entering the following command.

- root@laptop0-0:~# nmap <IP_of_B>

--Get the Screenshots of commands and output of A and B for Task 3 --
PART 2: Attacking and Stopping the attack

The last firewall rule in Part 1 is to allow A to access the port 22. Now A will exploit this fact and get access to a file on B’s desktop.

- Create a text file `example.txt` containing a simple message ‘Hello World’ on the desktop of B.

**Task 1: A attacks B**

[1pt] A will check for vulnerabilities on B and will find port 22 open.
- `root@laptop0-0:~# nmap <IP_of_B>`

[1pt] A will login into B using SSH.
- `root@laptop0-0:~# exit`
- From A, login into B with ssh
  - `cse321@laptop0-0:~$ ssh <IP_of_B>`
  - Enter the enter password: `cse321`

[1pt] A gains access to the files on B.
- A views the contents of file `example.txt`
  - `cse321@laptop0-0:~$ cat Desktop/example.txt`

--Get the Screenshot of the commands and output of A for Task 1--

**Task 2: B stops attack from A**

[1pt] B kills the current ssh connection from A
- `ps -ax|grep sshd`
- `root@laptop0-0:~# kill -1 <process_id from the line containing @pts/>`

[1pt] B updates firewall to prevent future attacks.
- `root@laptop0-0:~# ufw deny from <IP_of_A> to any port 22`
  - View the added rule.
  - `root@laptop0-0:~# ufw status`

[1pt] A is now denied access to B
- `root@laptop0-0:~# ssh <IP_of_B>`
  - As B is now denied access, the ssh request will not display any request and eventually the connection gets timed out.

-- Get the Screenshot of the commands and output of A and B for Task 2 --
Task 3: Reset Firewall to the Default

[2pt] This is the end of the lab, reset and enable the firewall on A and B.

- root@laptop0-0:~#ufw reset
- root@laptop0-0:~#ufw enable

- root@laptop0-0:~#ufw status
  Status should print active with no rules displayed.

-- Get the Screenshot of the commands and output of A and B for Task 3 --

SUBMISSION:

You will name the document as Teams_<team no.>_<team no.>

- Mention Team IDs
- Mention the ip addresses of A and B
- The document should contain screenshots for each task with screenshots for B followed by screenshots for A. No explanations are needed.

Submission will be done through Blackboard, deadline is tonight 11:59 pm.