SIMPLE WEB SCANNER

Prepared by: Noemi Mateo
NoemiM@genstudents.org

Generation Singapore Cloud Support & DevOps 2025 - Cohort 04

Project Summary

Schedule a Cron Job at regular intervals, such as every 10 minutes, to initiate an nmap scan of the network and save the results to a text file. Additionally, incorporate several enhancements and display the findings in a web browser.

TASK A

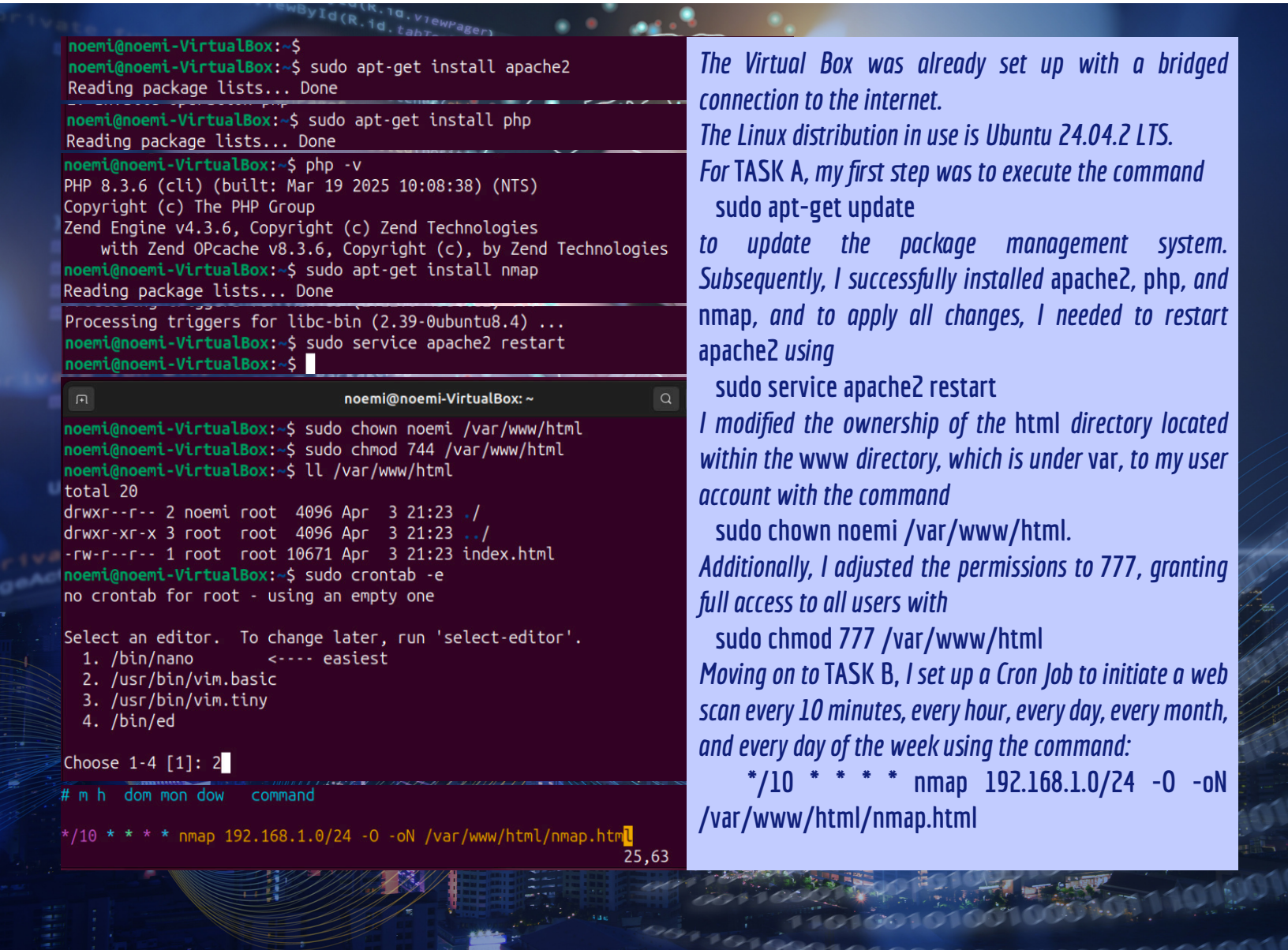
- Set up a Virtual Machine and install Ubuntu.
- Install Apache2, PHP, and nmap.
- Adjust Ownership and Permissions accordingly.

TASK B

- Configure the Cron Job.
- Develop the network.php file.

Learning Objectives

- Install and navigate a Linux distribution.
- Gain insight into the User Interface (UI) and Command Line Interface (CLI) of Linux (Ubuntu).
- Create Cron Jobs for scheduled tasks (crontab).
- Become familiar with NMAP.
- Understand basic file and folder permission changes and ownership (chmod, chown, members, groups).



The Virtual Box was already set up with a bridged connection to the internet.
The Linux distribution in use is Ubuntu 24.04.2 LTS.
For TASK A, *my first step was to execute the command*

sudo apt-get update

to update the package management system. Subsequently, I successfully installed apache2, php, *and* nmap, *and to apply all changes, I needed to restart* apache2 *using*

sudo service apache2 restart

I modified the ownership of the html *directory located within the* www *directory, which is under* var, *to my user account with the command*

sudo chown noemi /var/www/html.

Additionally, I adjusted the permissions to 777, granting full access to all users with

sudo chmod 777 /var/www/html

Moving on to TASK B, *I set up a Cron Job to initiate a web scan every 10 minutes, every hour, every day, every month, and every day of the week using the command:*

*/10 * * * * nmap 192.168.1.0/24 -O -oN /var/www/html/nmap.html

The Cron Job was configured to examine the devices on the network, determine the operating system (-O) running on each device, along with all the additional information typically gathered by nmap, and to format this output as standard text (-oN) prior to saving it in nmap.html. This file would serve as the foundation for the website content displayed by the apache2 server when requests are made from the browser.

Before the content could be presented, I needed to produce a PHP file that would structure it in a way that is more user-friendly and visually appealing for those seeking the information. The network.php file was developed using a text editor and was subsequently saved in the /var/www/html/ directory.

Open network.php /var/www/html

```php
<?php

date_default_timezone_set("Asia/Singapore");

echo "Server Timestamp: ";
echo date("h:i:sa");

echo "<pre>";
include("nmap.html");
echo "</pre>";

?>
```

It essentially presents a timestamp followed by the scan details as specified in the crontab, including the scan type, target IP addresses, and the file designated to hold the scan output. Subsequently, the file displays all the scan results in an organized format. Clearly, the use of the <pre></pre> tags enhances the text's layout; without them, the information would appear as a continuous block. I included a line at the top to adjust the time to the Asia/Singapore timezone.

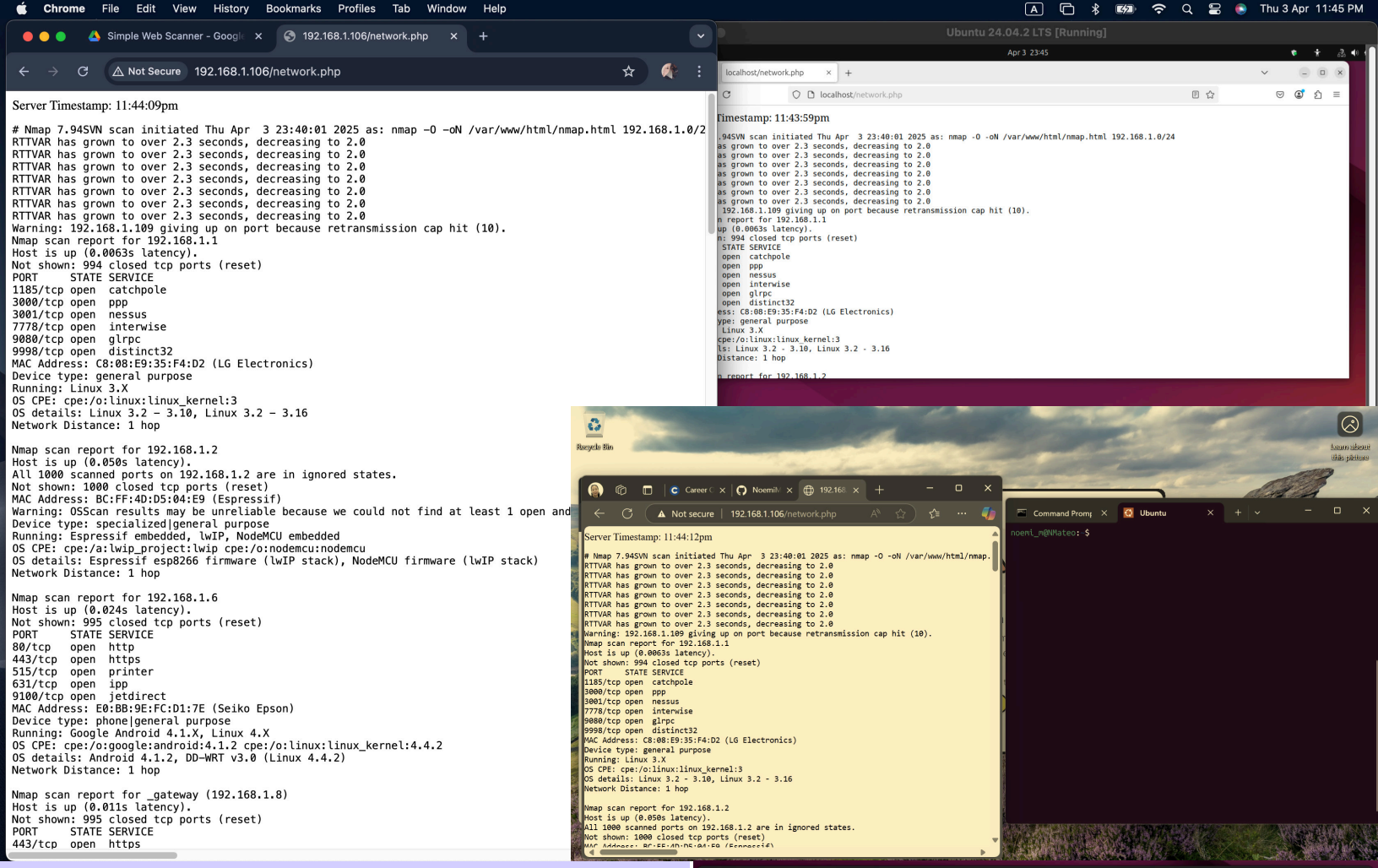
Bonus Task A

As previously stated, I initially assigned 777 permissions to the html directory.

This configuration granted me root permissions, along with identical access to any users in the same group and everyone else.

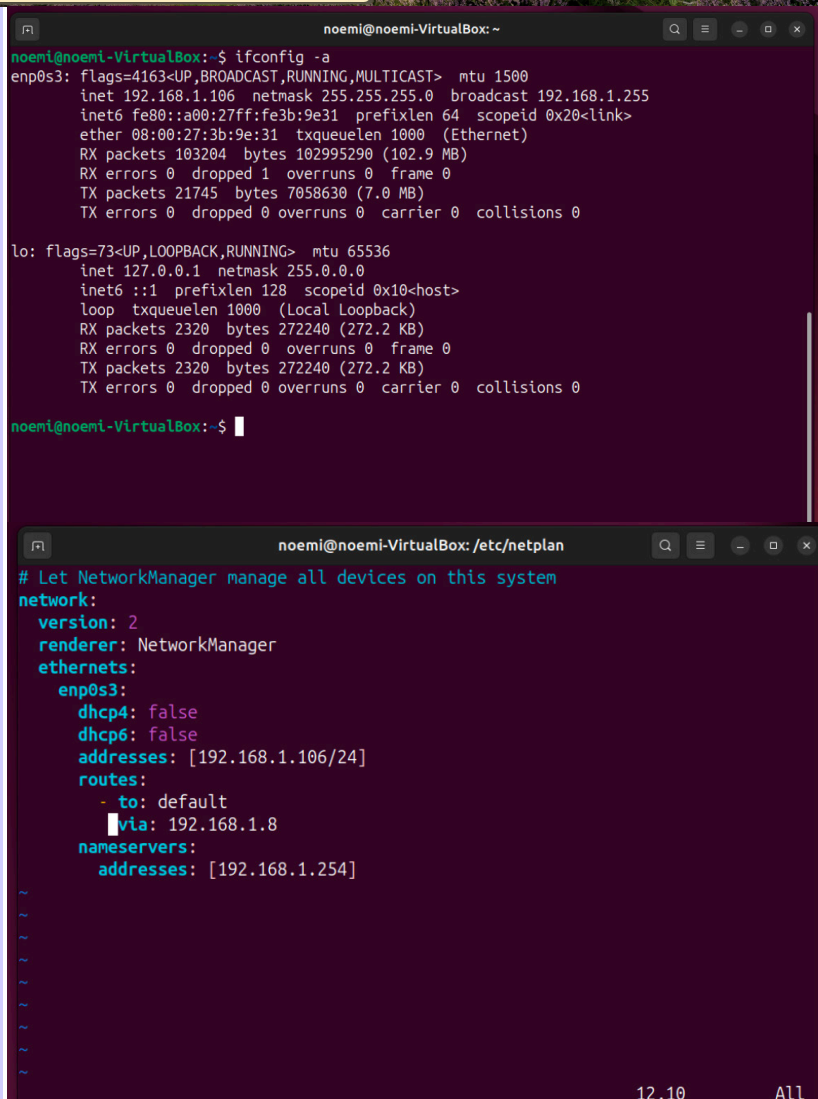
In terms of security, this permission configuration stands as the most harmful choice available. To protect your data and systems effectively, it is essential to grant users only the minimum privileges required to complete their tasks.

The value 7 corresponds to read (4), write (2), and execute (1) permissions. Since directories require executable permissions, I opted to modify the setting to 755, as other permission combinations would restrict my access to the nmap output, despite my attempts with various values.



Bonus Task B

After conducting research and learning how to modify the /etc/netplan/01-network-manager-all.yaml file with the new configuration in vim, I confirmed that my Ubuntu was already connected via a bridged connection. I then attempted to access the network.php file on Ubuntu using the Safari browser on my host system, which was a MacBook Pro. Additionally, I was able to view the scan results from my Windows laptop, and the open WSL window serves as evidence that the connection was successfully established through the Edge browser on the Windows device.



Recommendations

- *Opt for permissions other than 777. Create users with minimal privileges, and add to the sudoers group for elevated access if needed.*
- *Utilize tools like nmap solely within your personal network and avoid their use on secure systems to prevent alerts. Refrain from using such tools at work to avoid triggering alarms or logs of suspicious activity.*

Note: snapshot of nmap's impact on Windows laptop.