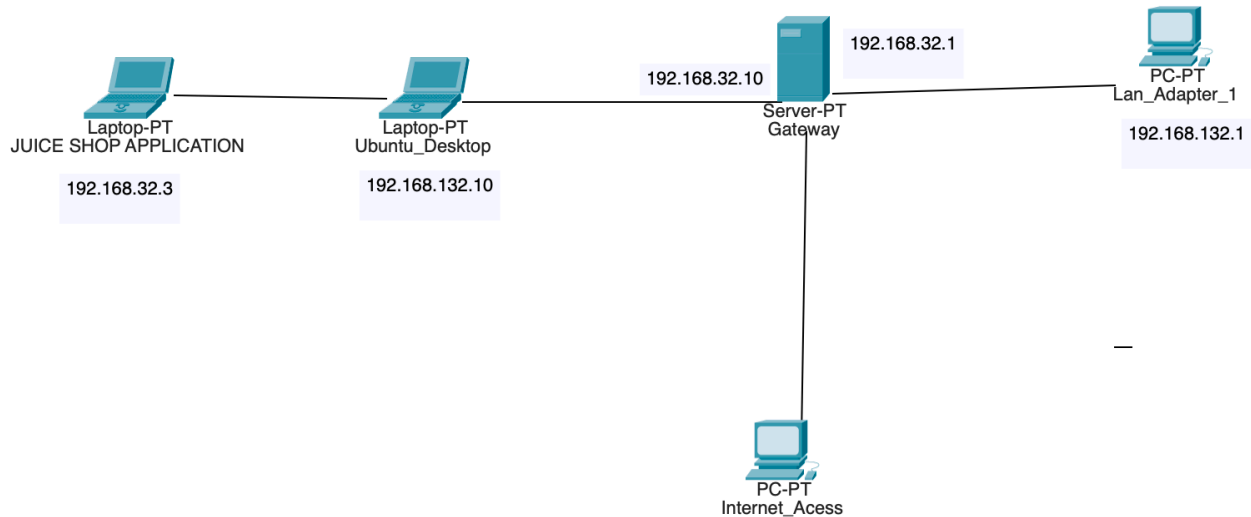


Networking Diagram



Ip table -

Device	Role	IP Address	Subnet Mask
Desktop VM	Management	192.168.32.10	255.255.255.0
Gateway router VM (enp0s3)	Subnet 01 - Internal Network	192.168.32.1	255.255.255.0
Gateway router VM (enp0s8)	Subnet 02 - Internal Network	192.168.132.1	255.255.255.0
Gateway router VM (enp0s9)	Internet Access	192.168.132.10	255.255.255.0
Application Server VM	Server	192.168.32.3	255.255.255.0

Configuration report

The steps to configure the different virtual machines weren't so simple just for the reasoning In the beginning, I first configured an Ubuntu server as a router instead of using the Ubuntu gateway server, so there was a bit of time wasted when configuring the virtual machines, but along the way I had personally learnt some new command lines in which can be used as an alternative to some of the other command lines. An example is that instead of going through `sudo nano` and entering the file name, I was shown a code along the lines of `"sudo ifconfig enp0s8 ip..... "` and this assigned the IP address to the Network interface I wanted to assign it to.

When setting static IP addresses to each virtual machine, there were no issues, as I had previously done this in an undergraduate class, so I had some knowledge of what to do. My only

problem was when I checked that the virtual machines had internet connectivity, and it wasn't working. This would be due to not having the network settings set on an internal connection, so it wasn't a major issue. But the main issue of having no network connection was on the Ubuntu desktop where even though the IP address had been set, the internet was running as should, and the result of this problem was that in the DNS section of the IPV4 setting, there was no ping 8.8.8.8 and there was no Mac connection.

Juice shop configuration on Ubuntu Desktop

- Downloading node.js on terminal
- Cloning git hub respiratory
- Multiple issues
- No files were found in the directory

Configuring two application servers -

For an additional factor for this coursework, I am going to be reviewing the factor of what steps would need to be configured if there was going to be an additional application server being run on the other internal network, which in this case would be subnet 02 as subnet 01 is running juice shop application server.

This is an example of what the IP address table would look like when another application server is running on subnet 02.

Device	Role	IP Address	Subnet Mask
Desktop VM	Management	192.168.32.10	255.255.255.0
Gateway router VM (enp0s3)	Subnet 01 - Internal Network	192.168.32.1	255.255.255.0
Gateway router VM (enp0s8)	Subnet 02 - Internal Network	192.168.132.1	255.255.255.0
Gateway router VM (enp0s9)	Internet Access	192.168.132.10	255.255.255.0
Application Server VM	Server	192.168.32.3	255.255.255.0
Desktop VM 2	Server		255.255.255.0

Steps to configure the application server on Ubuntu Desktop.

The first step which is needed to configure an application server onto the Ubuntu server is to make sure the server is up to date using the command line:

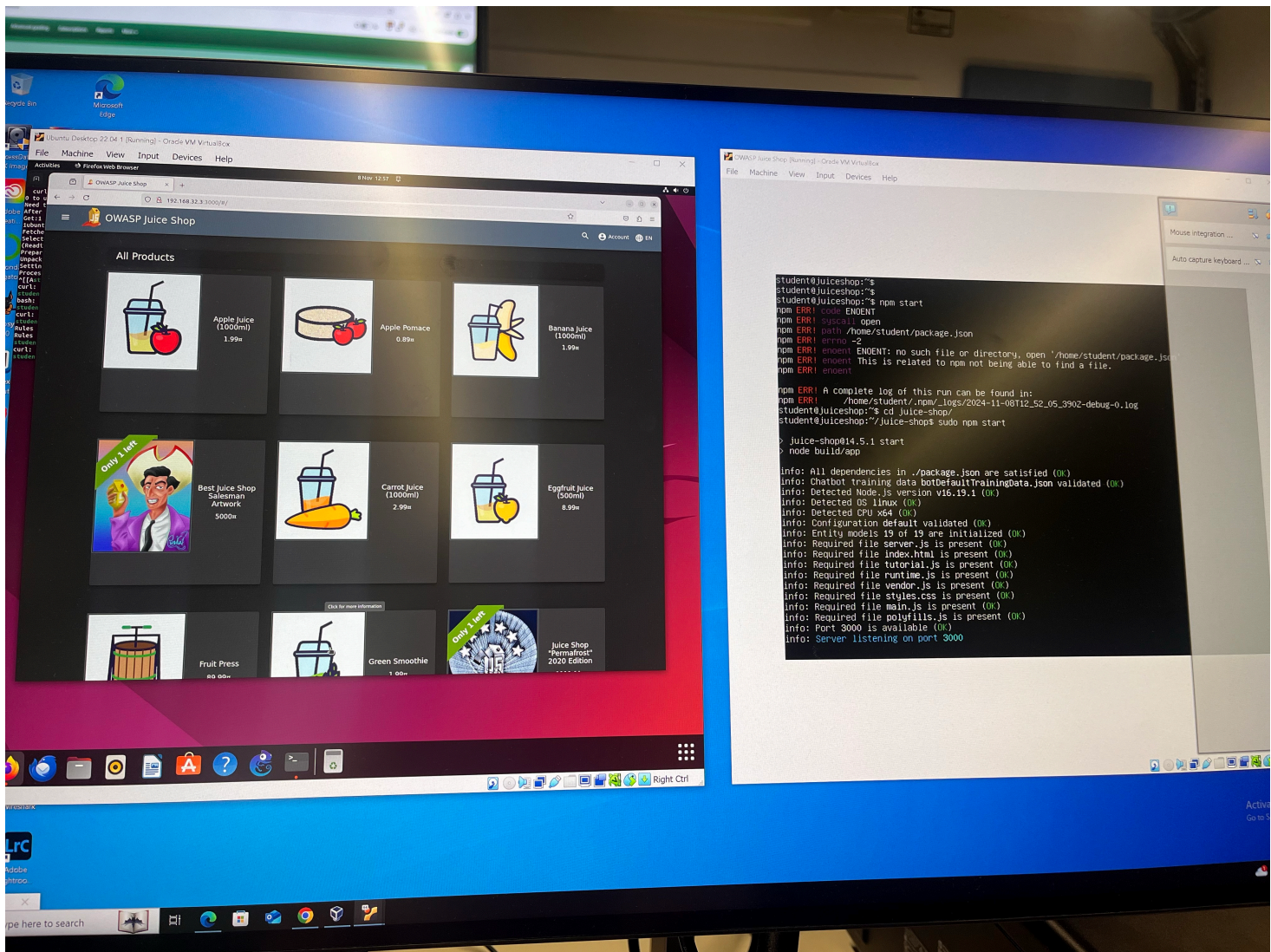
Sudo apt update && sudo apt upgrade

The next step is to install node.js as

Sudo apt install -y nodejs npm

And then you can verify the installation of nodejs by using the two command lines -

Node -v
Npm -v



After installing the packages required to run the application, it is then needed to set up the server. Firstly, we need to navigate the app directory by using the command line -

Cd /path/to/your/application

Then install the Dependencies because we are using Node.js
- why do we need to install the Dependencies

Npm install

Finally, once these steps have been configured and everything is running correctly, we can run the application from the command line.

Node app.js

- What application can be run on the Ubuntu server?
- What tools would be best for looking for vulnerabilities in the Ubuntu server application?
- Purpose of running an additional application
- Network diagram of the configuration

So, setting up the OWSAP juice shop on the Ubuntu desktop using the GitHub clone respiratory didn't work, as it wasn't reading the files even though I was putting them in the directory correctly. So, instead, I used the juice shop application, which was already set up but required some additional steps to be set up correctly.

Referencing - Chatgpt - Friday 8th November 2024

To use the Juice Shop application through Ubuntu Desktop, I first had to assign the application a static IP address, which in this instance was 192.168.32.3.

So, after the IP address was assigned, I needed to ensure there was a connection between the application server and the desktop; otherwise, I wouldn't be able to access the application server. This is where I encountered my first problem. Although I was able to ping the two virtual machines together, I was then unable to load the Juice Shop application through Firefox on the Ubuntu desktop.

As I was going through the application, researching ChatGPT, and considering some of the possible problems, I checked the firewall settings to see if the port was running through, and it was.

However, the issue was with the juice shop application itself, as I needed to open the juice shop directory and run `npm start` to start the application.

The final step that was configured is that the server was listening on port 3000, so to access the application onto ubuntu-desktop, I needed to run

<https://192.168.32.3:3000> to access the juice shop.

Connecting Juice Shop via Firefox - 192.168.32.3/3000/hashtag/

- First, download ZAP for OWSAP
- Install the necessary updates
 - `sudo apt-get install openjdk19`
- Download the Linux package and installer OWSAP from [ZAP.com](https://www.zaproxy.org/docs/linux/)
- Extract files
- Run `Zap.sh` as a program

First video - Opening ZAP, connecting URL, but the URL won't connect due to an internet Issue