cct | College Dublin
Computing • IT • Business

Networking & Virtualization / Continuous Assessment 1

Programme Title/Year:	*Level 8- Higher Diploma in Computing*
Module Title(s):	Networking & Virtualization
Lecturer Name(s)	Michael Weiss, mweiss@cct.ie
Assessment Title:	Proof of Concept Linux Virtual Network Project
Assessment Type:	*INDIVIDUAL assignment*
Assessment Weighting:	(50% of CA total)
Date Issued:	Friday 7th March 2025
Due Date (Deadline):	Sunday 23rd March 2025 (before 23.59)
Late Submission Penalty:	Late submissions will be accepted up to **5** days after the deadline. All late submissions are subject to a penalty of **10%** of the mark awarded. Submissions received more than 5 days after the deadline above **will not** be accepted.
Method of Submission:	**Moodle**
Feedback Method:	**Results posted in Moodle gradebook**
Instructions for Submission:	Document to be submitted in a professional **.pdf** file format. It will be saved in the format **student_number_report.pdf** *No work accepted through email*
Student Name:	Sivonei Ribeiro dos Santos
Student Number:	2024568

1. Introduction

The objective of this work is to develop a proof of concept for Dublin City College (DCC), which has contracted us to create a virtualized network infrastructure based on Linux. DCC wants to understand how its new network environment and services will perform on a Linux operating system, allowing for the evaluation of functionalities such as connectivity, security, and performance in a virtualized environment.

The proof of concept will consist of several stages, including the construction of a virtualized client-server network using VirtualBox software. The project will involve the installation of an Ubuntu Server 24.04, which will act as a web server, and an Ubuntu 24.04 client, which will be used to test connectivity with the server. Additionally, it will be necessary to install the Apache web server, configure a "Under Construction" page for DCC, and use the Lynx browser to view this page. Network configurations will also be performed, such as assigning static IPs, conducting connectivity tests, and setting up a basic firewall to ensure the security of the environment.

2. Virtual Linux Network and Web server

To start the project, two virtual machines (VMs) were installed using Ubuntu Server 24.04 on VirtualBox. The VMs were configured with two network adapters each:

ubuntuserver:
Adapter 1 (NAT): Configured to use DHCP.
Adapter 2 (Host-only): Configured with a static IP 192.168.56.100/24 for internal communication between the VMs and the host.

ubuntuclient:
Adapter 1 (NAT): Configured to use DHCP.
Adapter 2 (Host-only): Configured with a static IP 192.168.56.101/24 for internal communication.

System updates and connectivity testing.

After the network configuration, connectivity tests were performed between ubuntuserver and ubuntuclient, as well as internet connectivity tests using ping and system updates using sudo apt update & sudo apt upgrade. Below are examples of the configurations:

Apache Installation and Web Page Configuration

In this step, we installed the Apache web server on the ubuntuserver to host the test page. Then, we modified the file /var/www/html/index.html to create a test page.

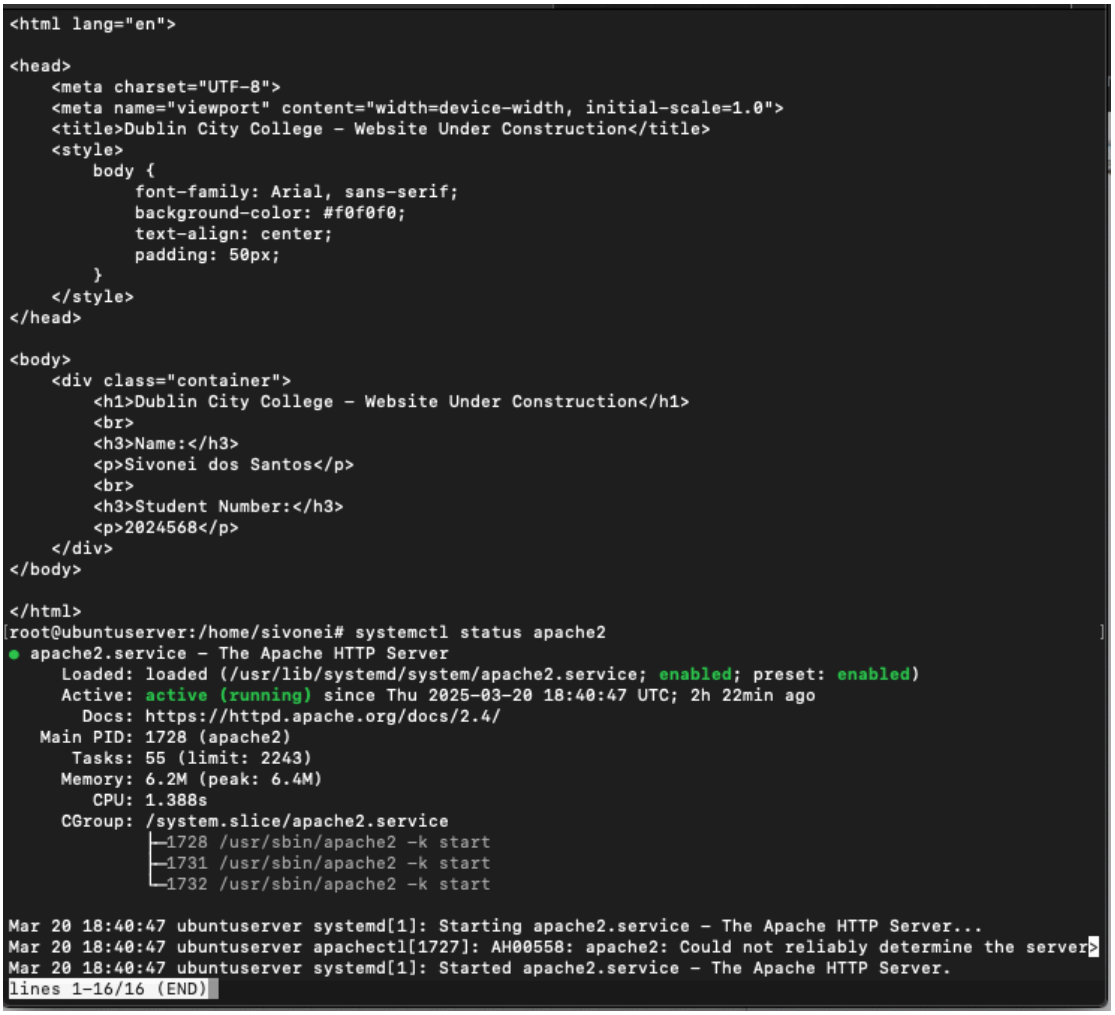

Figure 01: Ubuntuserver computer settings - apache2

Lynx and Wireshark Installation

Continuing with the project, we will now install the Lynx system on ubuntuclient and Wireshark on the host computer. These two systems will be used to validate the configurations made earlier. Lynx will be used to access the web server via the terminal with the following command: lynx http://192.168.56.100. Meanwhile, Wireshark will be used to monitor network packets. Below are the procedures carried out for the installation and configuration of Lynx and Wireshark.

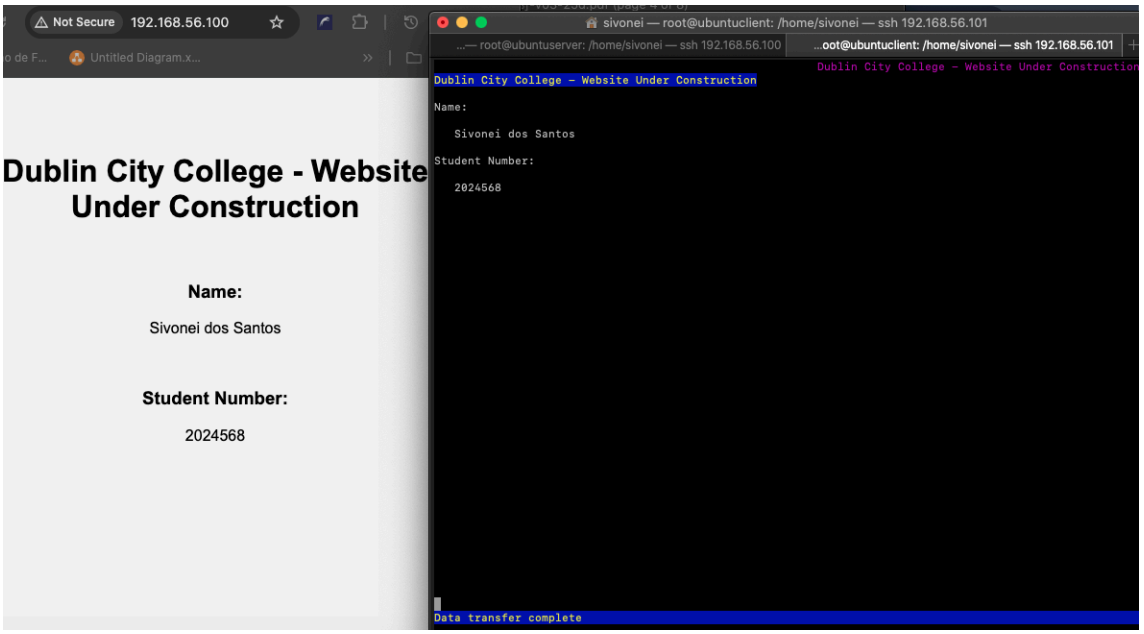

Figure 02: Accessing the web page that is hosted on the Ubuntu server

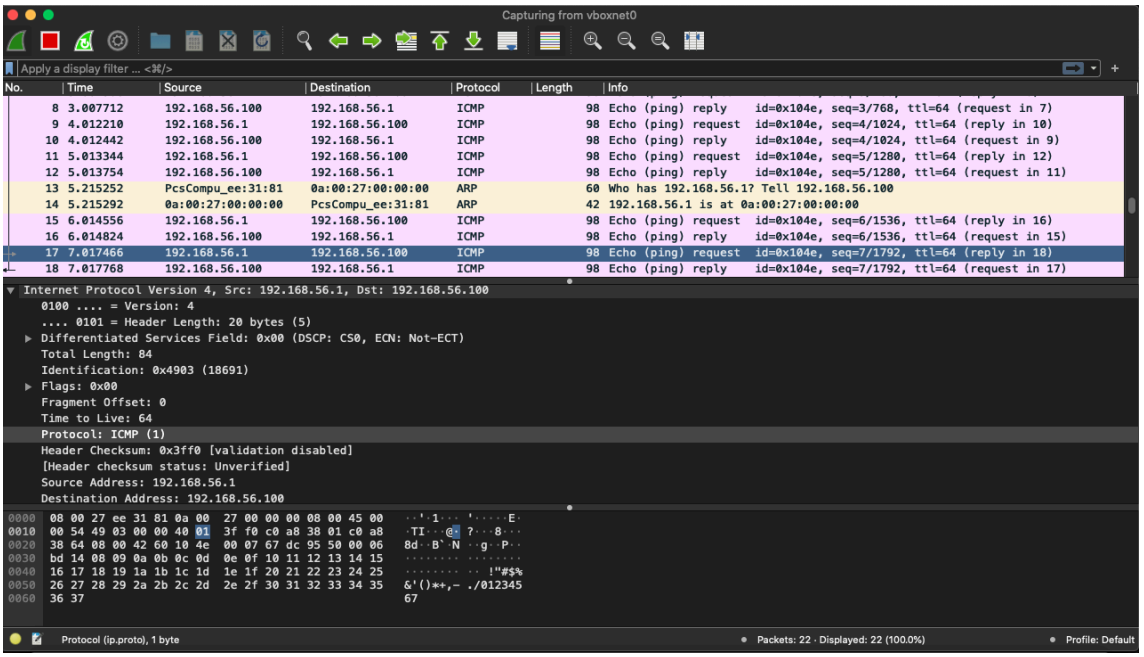

Figure 03: ICMP traffic between the ubuntuserver and my host computer.

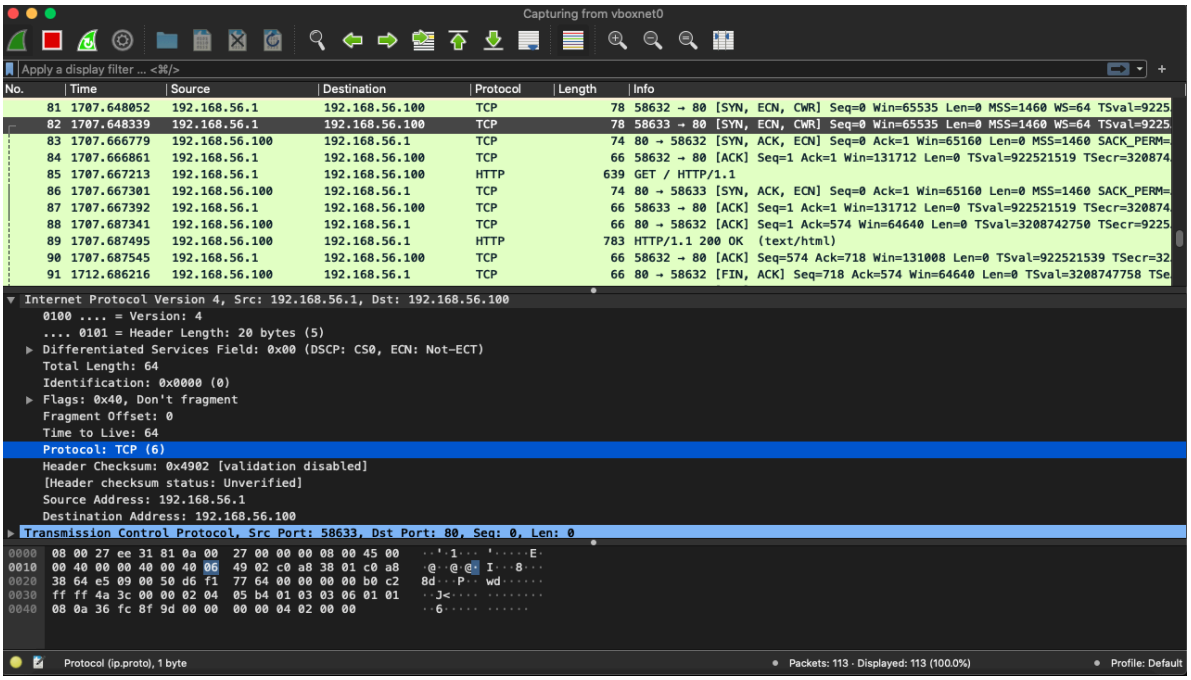

Figure 04: TCP/IP 3-way handshake that occurs while the HTTP contents (webpage).

3. HOSTNAME MANAGEMENT - Renaming the linux servers:

Continuing with the project, we will rename the virtual servers ubuntuserver and ubuntuclient to facilitate their identification in the infrastructure of Dublin City College (DCC). The server ubuntuserver will be renamed to web-server-568, while ubuntuclient will be changed to web-client-568. This task will be performed in the host configuration file, located at /etc/hostname, to ensure that these changes are permanent.

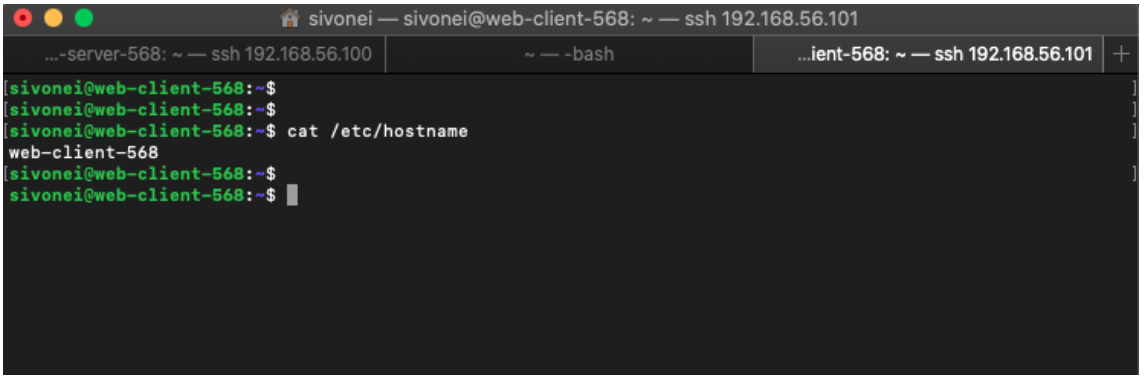

Figure 05: web-client-568

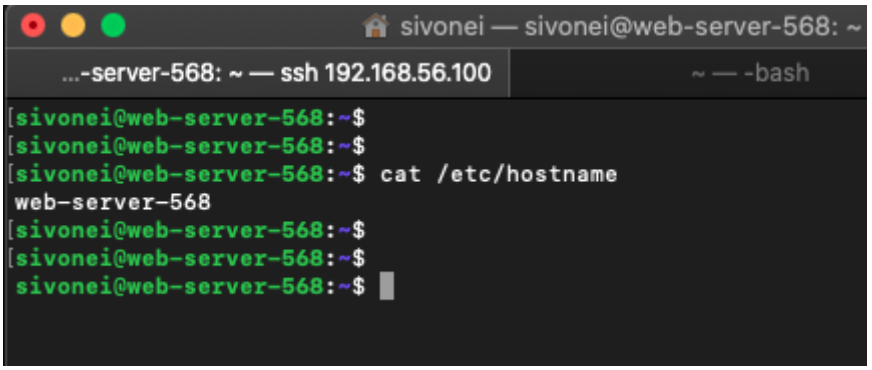

Figure 06: web-server-568

4. IP ADDRESS MANAGEMENT

To ensure that the network settings are persistent after reboots, Netplan, a network configuration tool in Ubuntu, was used. The Netplan configuration file (/etc/netplan/01-netcfg.yaml) was edited on both VMs to define the static IPs and network adapters.

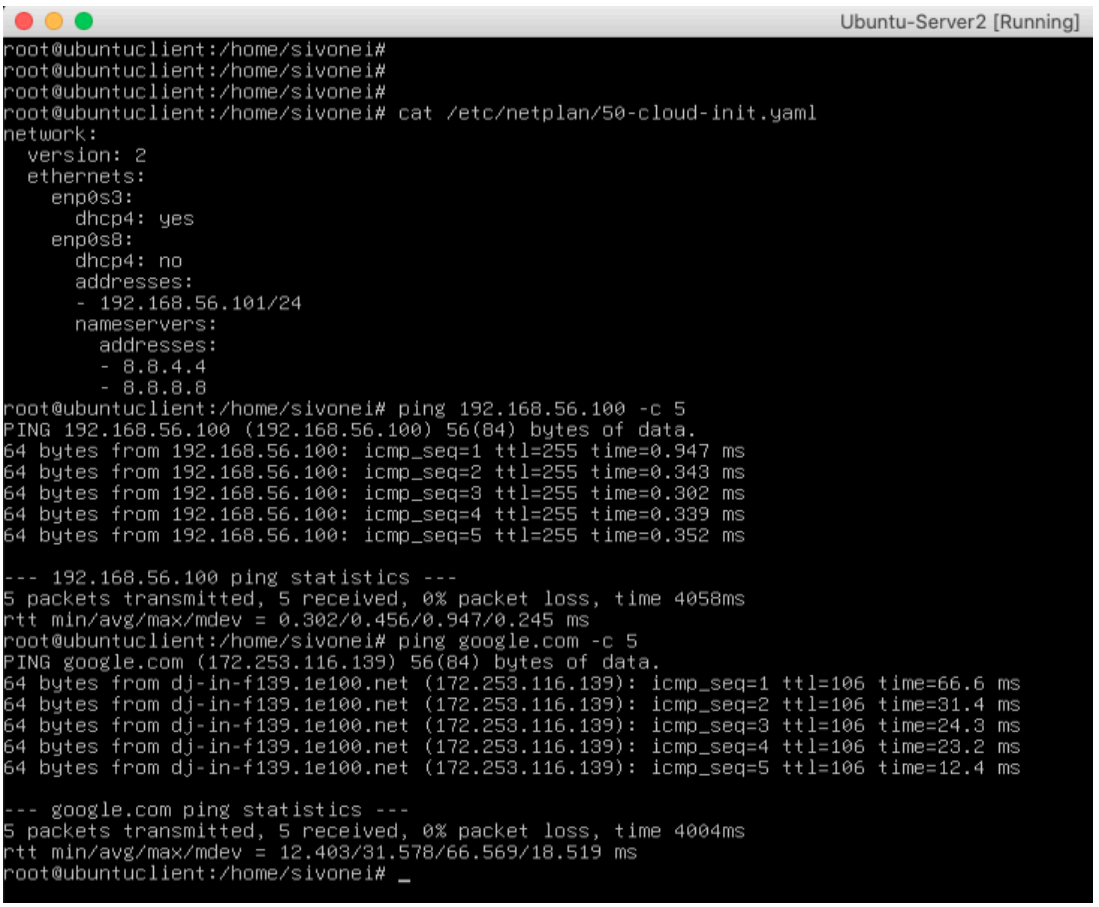

Figure 07: Ubuntuclient computer settings

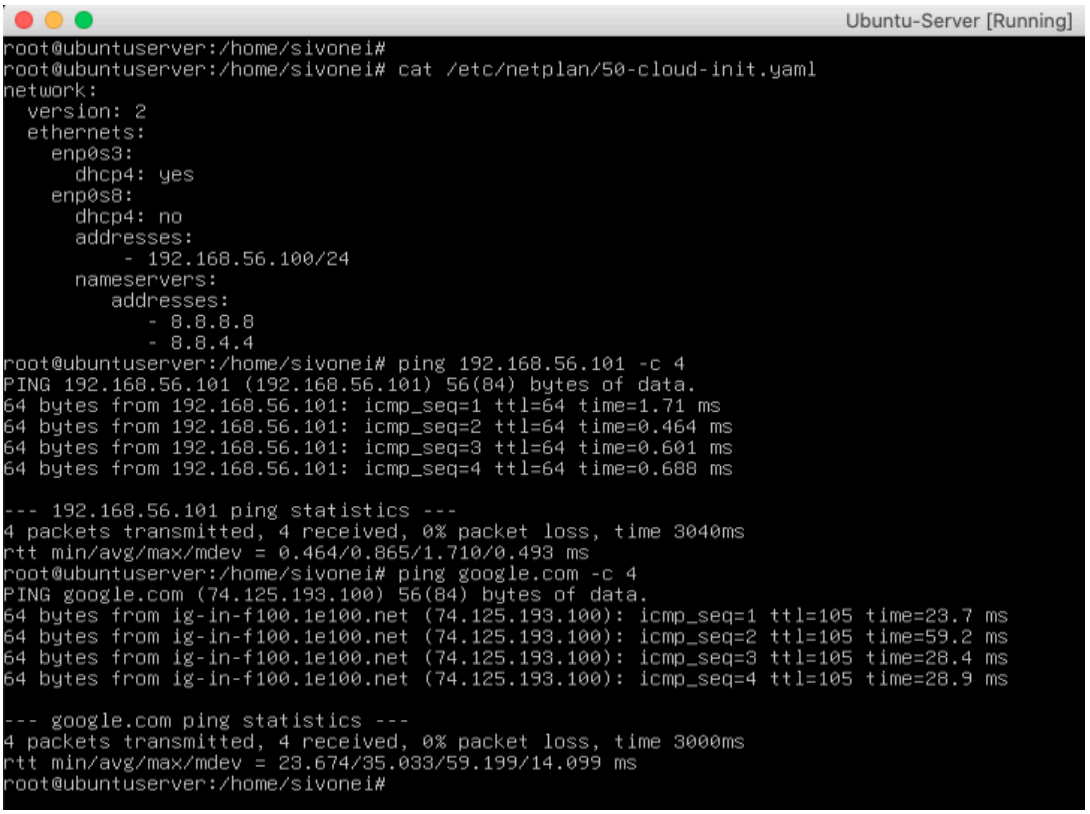

Figure 08: Ubuntuserver computer settings

5. Using SSH for secure transfer of data between the two server

In this task, we will connect to the Ubuntu server remotely via SSH using the Mac OSX Terminal. Then, we will use Wireshark on the host operating system to demonstrate how the communication between the client and the server is encrypted.

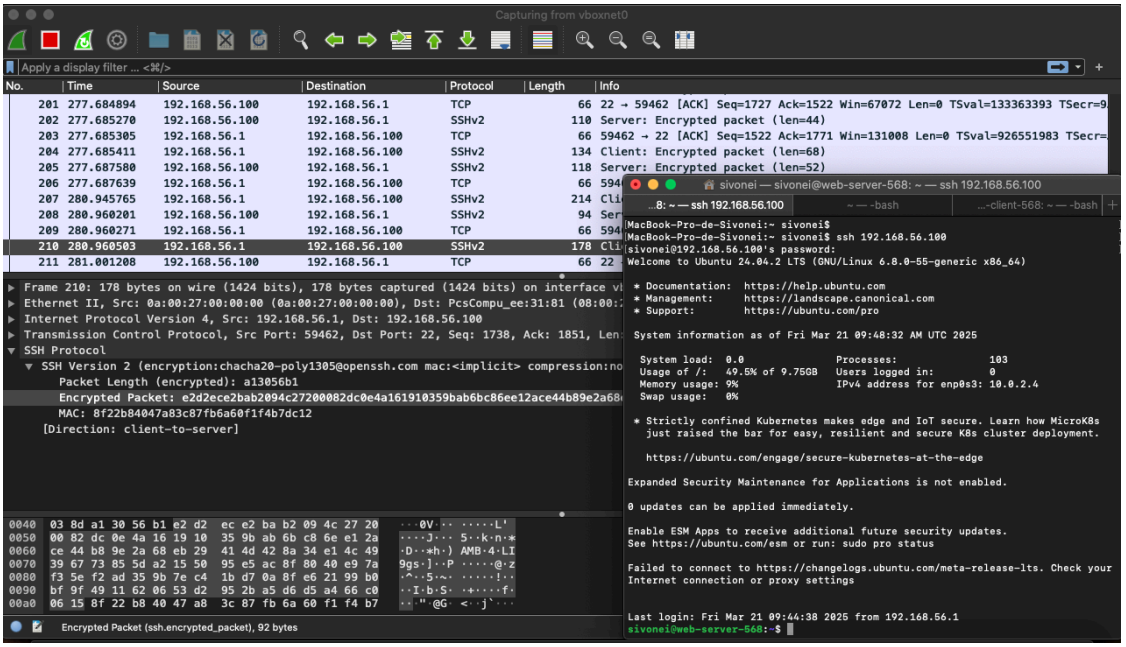

Figure 08: logging into the web-server-568 using SSH remotely.

6. Security configuration ufw - FIREWALL

In this task, we will use UFW (Uncomplicated Firewall) to manage web traffic between two servers. Initially, we will block HTTP traffic and provide screenshots demonstrating that web access is being denied. Then, we will allow HTTP traffic through the firewall and show that web access is restored. Additionally, we will perform a similar procedure with SSH traffic, where we configure UFW to allow and block SSH access, capturing the screens that evidence these changes.

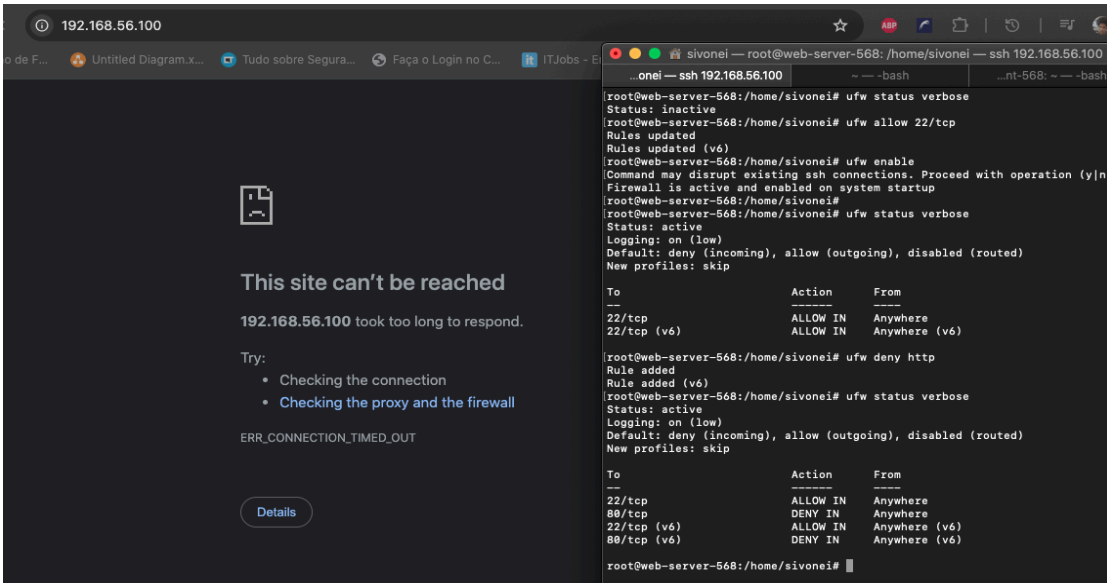

Figure 09: HTTP traffic is being denied by Firewall

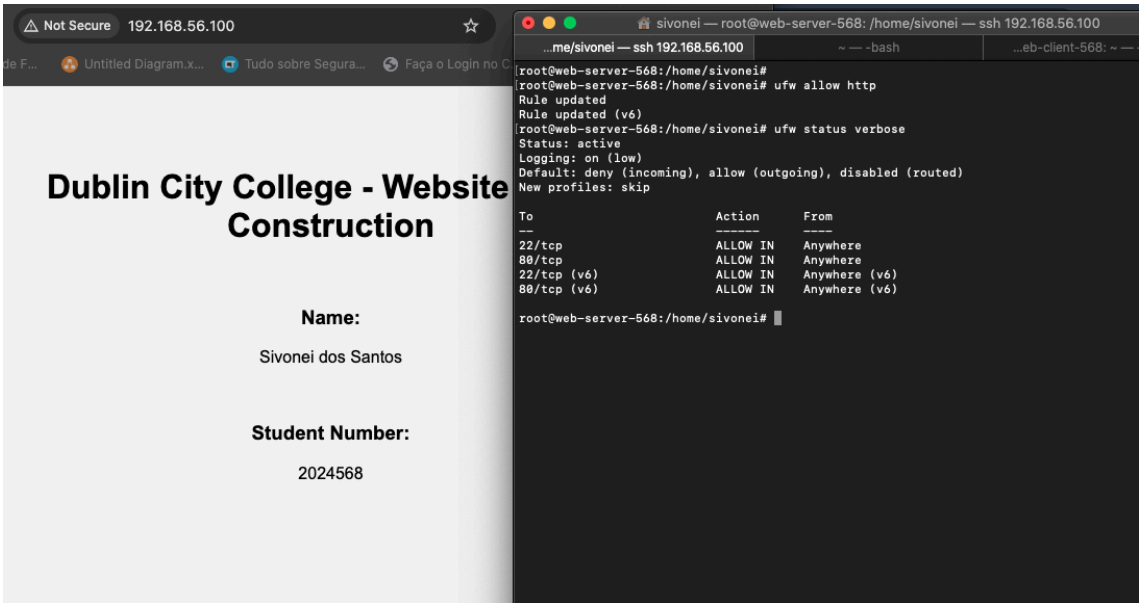

Figure 10: HTTP traffic is being allowed by Firewall

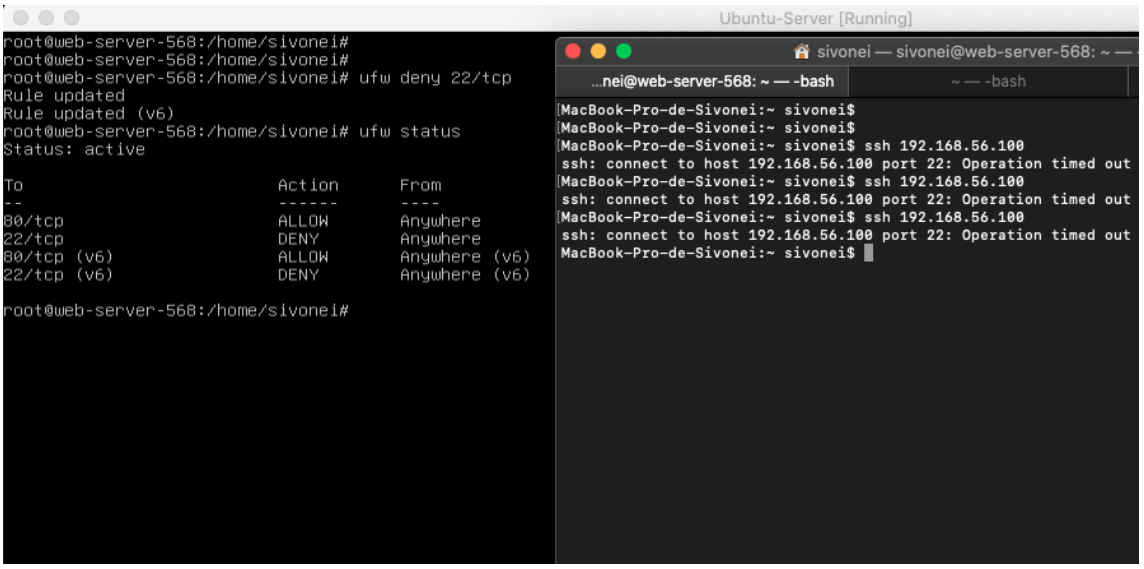

Figure 11: SSH traffic is being denied by Firewall

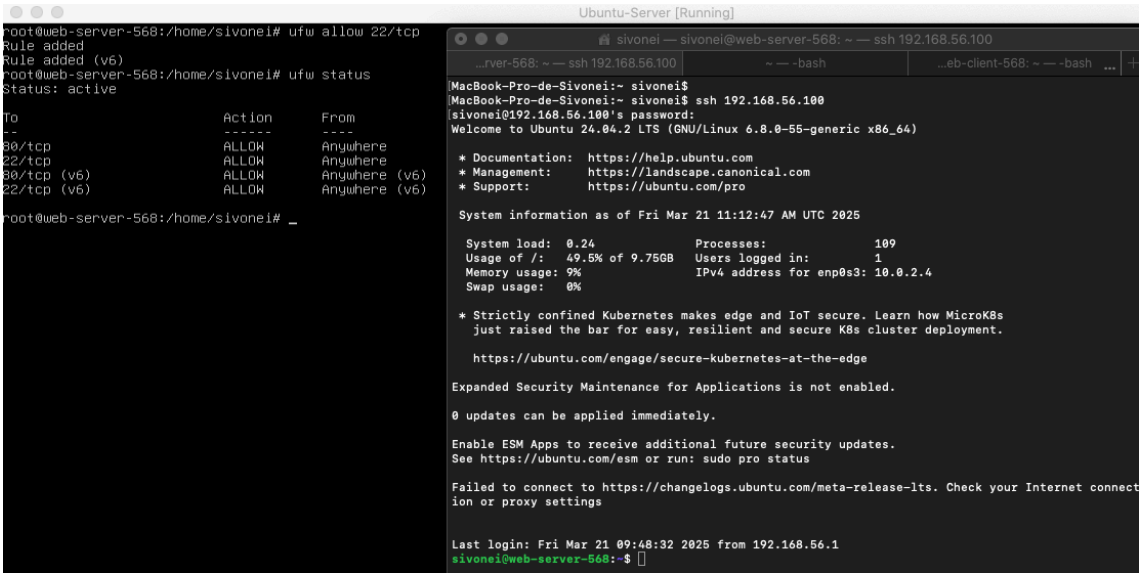

Figure 12: SSH traffic is allowed denied by Firewall

7. The SAMBA file server

Now, the goal is to install the Samba server on the web server web-server-568 and share a folder over the network. We will use a text editor, such as nano, to create a simple file. Then, we will configure the Samba share so that users are prompted to provide a password when accessing the folder. Additionally, we will demonstrate the ability to add a new file to the shared folder and modify it. To do this, it will be necessary to adjust the permissions of the shared folder to allow file additions.

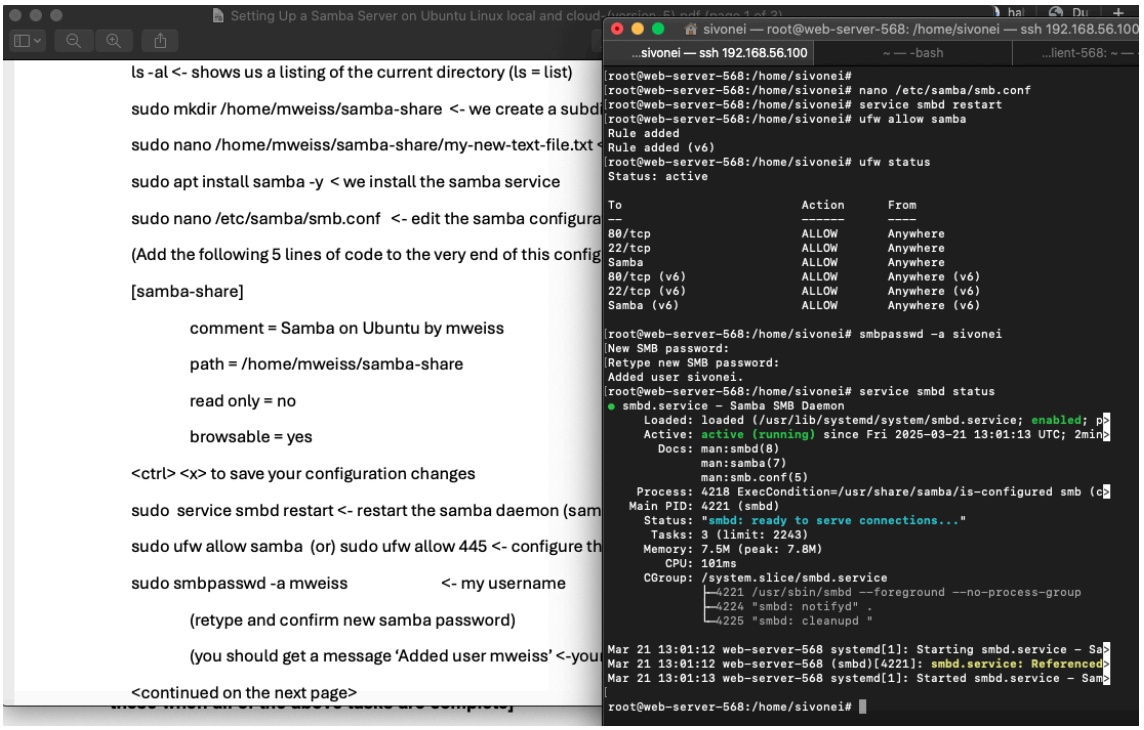

Figure 13: Samba server is configured

Samba Configuration and Permissions

Initially, access was granted to the shared folder at smb://192.168.56.100/samba-share, and the existing text file was read. Then, the permissions of the samba-share directory were changed to allow full access to all users by using the command chmod -R 777 samba-share/. After this modification, it was possible to create a new folder within the shared directory. Additionally, a new text document was created to store relevant information. Finally, the existing text file was modified to update its content as needed.

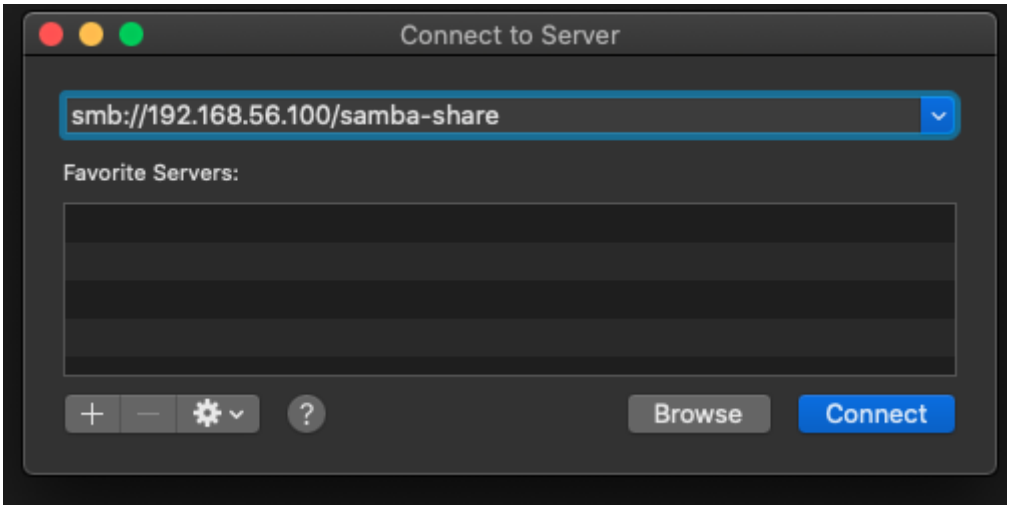

Figure 14: Connect to server 192.168.56.100

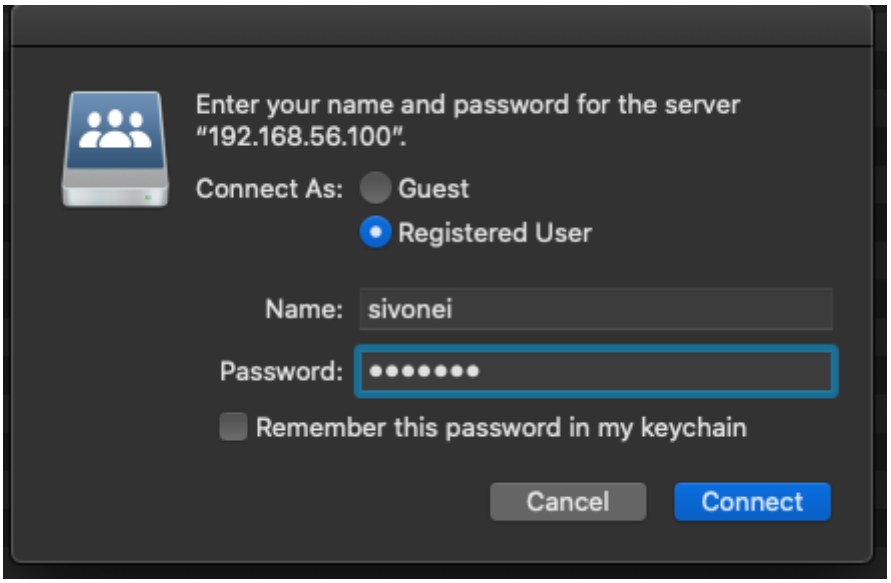

Figure 15: Credentials to connect 192.168.56.100

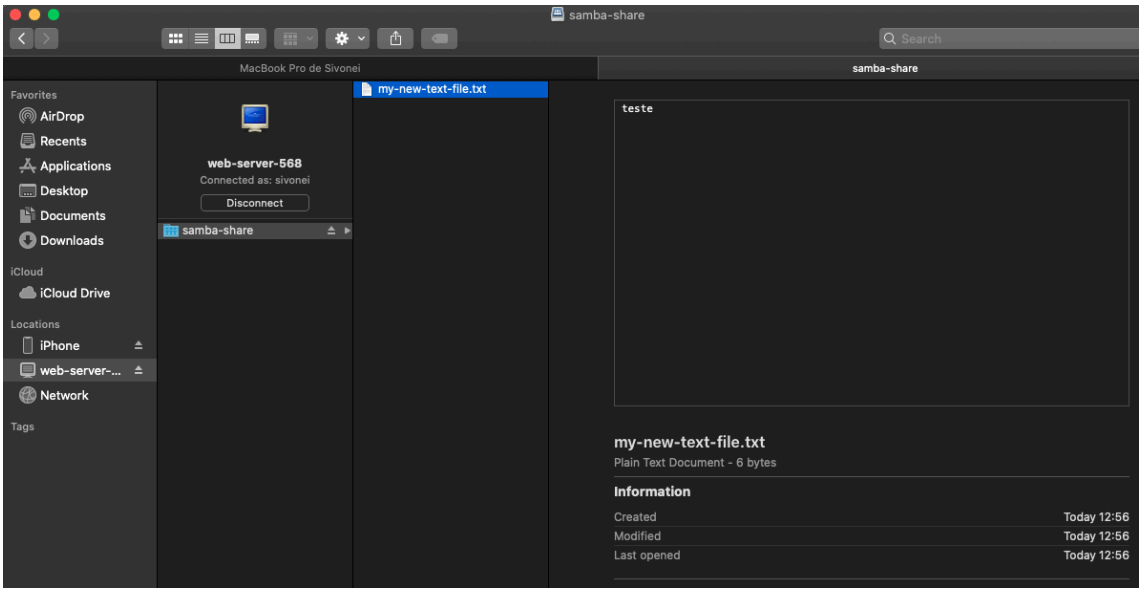

Figure 16: Access to the samba-share folder on server 192.168.56.100

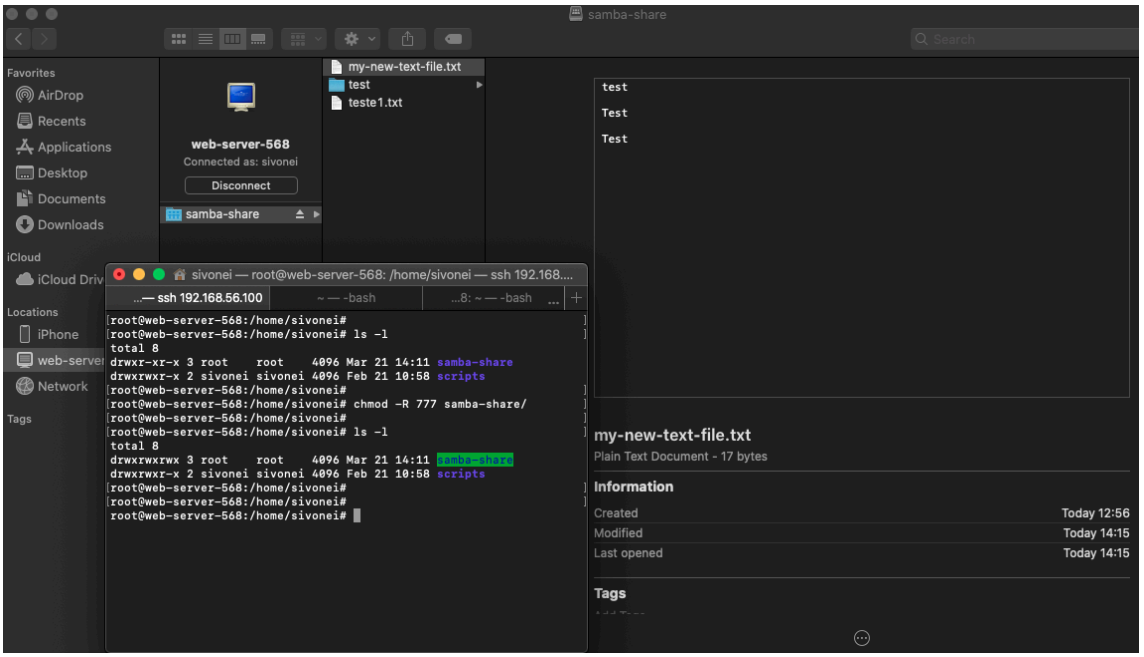

Figure 17: Access to samba-share folder on server 192.168.56.100 after configuration

What is Samba and Server Message Block (SMB)

Samba is a free and open-source software that implements the Server Message Block (SMB) protocol, enabling communication and sharing of files and printers between Unix/Linux and Windows systems. It acts as a server, allowing users to access resources on a network, regardless of the operating system they are using. Samba is widely used in mixed environments where Linux and Windows systems need to coexist and share resources efficiently (Alibaba Cloud, 2023; Ubuntu, 2023).

Examples of Samba Use

File Sharing:
Samba allows users of different operating systems to access and edit files in a shared directory. For example, a Windows user can access a folder on a Linux server as if it were a local resource. This is particularly useful in corporate environments where teams need to collaborate on documents and projects (Alibaba Cloud, 2023).

Network Printing:
Samba facilitates printer sharing on a network. Printers connected to a Linux server can be accessed by Windows computers, allowing users to send print jobs from any device on the network (Ubuntu, 2023).

Integration in Corporate Environments:
Many companies use Samba to integrate diverse systems, enabling different departments to share resources seamlessly. This includes access to files, printers, and

other network devices, regardless of the operating system in use (Alibaba Cloud, 2023).

Server Message Block (SMB)

The Server Message Block (SMB) is a network protocol that operates at the application layer of the OSI model. It was originally developed by IBM and later enhanced by Microsoft, becoming a standard for file and printer sharing in Windows networks. SMB allows devices on a network to communicate and share resources efficiently, such as files, printers, and serial ports (Ubuntu, 2023).

Samba, by implementing the SMB protocol, provides an effective solution for mixed environments where Linux and Windows systems need to interoperate. It is widely adopted in corporate and home networks, offering flexibility and compatibility between different operating systems (Alibaba Cloud, 2023; Ubuntu, 2023).

8. RESEARCH AND CHALLENGE ACTIVITIES

Security configuration IPTABLES

The iptables utility is a Linux-based firewall that comes pre-installed on many distributions. It is a leading solution for software-based firewalls and a critical tool for Linux system administrators to learn and understand (Liquid Web, n.d.; UpCloud, n.d.).

To get started with iptables, you need to add rules to allow inbound traffic for the services you require.

It is important to allow connections that are already established or related. This is particularly useful for traffic such as SSH, where you may initiate an outbound connection and want to accept inbound traffic from the connection you have intentionally established (Liquid Web, n.d.).

To automate the process of restoring rules upon reboot, you can install an additional package for iptables that handles loading saved rules: the iptables-persistent package (UpCloud, n.d.).

To demonstrate its functionality, we performed the following steps: installing iptables-persistent, configuring rules for the SSH and HTTP protocols, and then testing access, which worked normally.

Next, we deleted the two rules created for SSH and HTTP and created another rule to block access to these services. Below is a screenshot of the configurations and the test results.

```
root@web-server-568:/home/sivonei#
root@web-server-568:/home/sivonei# iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
root@web-server-568:/home/sivonei#
root@web-server-568:/home/sivonei# iptables -A INPUT -p tcp --dport ssh -j ACCEPT
root@web-server-568:/home/sivonei#
root@web-server-568:/home/sivonei# iptables -A INPUT -p tcp --dport http -j ACCEPT
root@web-server-568:/home/sivonei#
root@web-server-568:/home/sivonei# iptables -L -v -n --line-numbers
Chain INPUT (policy ACCEPT 751 packets, 73214 bytes)
num   pkts bytes target     prot opt in     out     source               destination
1       74  5288 ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0           ctstate RELATED,ESTABLISHED
2        0     0 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
3        0     0 ACCEPT     6    --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80
```

Figure 18: SSH and HTTP ports accepted by firewall

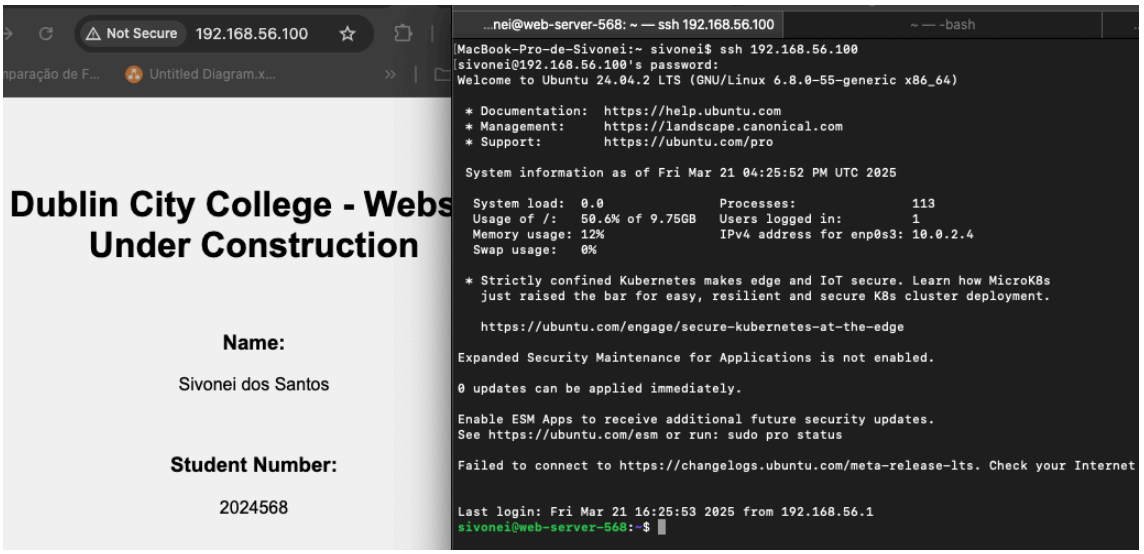

Figure 19: Successful connection to SSH and HTTP firewall ports

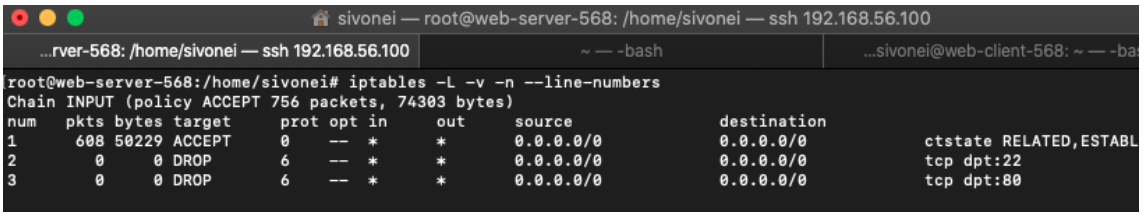

Figure 20: SSH and HTTP ports danied by firewall

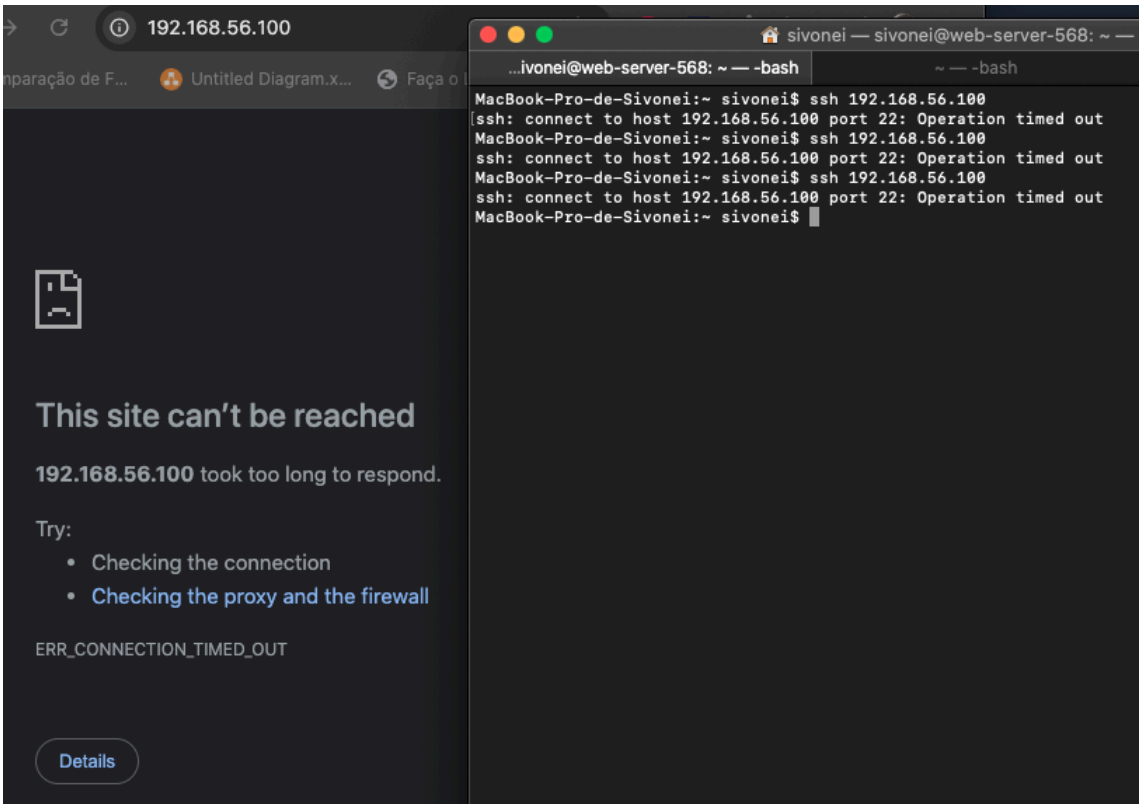

Figure 21: Refused connection to SSH and HTTP firewall ports

Linux Challenge Task: IInstall the V-Box Guest Additions onto the ubuntuserver VM

The task involves installing the VirtualBox Guest Additions on the ubuntuserver VM to enable shared clipboard functionality between the host machine and the VM. This feature allows users to copy text from the host (macOS) and paste it into the VM (Linux), or vice versa. While the Guest Additions provide this functionality, an alternative approach using the native macOS terminal and SSH is suggested for improved security and reliability.

Proposed Solution: Using the Native macOS Terminal

Given the challenges encountered with the VirtualBox Guest Additions and the shared clipboard feature, I recommend using the native macOS terminal connected via SSH to the ubuntuserver VM. This approach is more secure, reliable, and does not require additional software installations like the Guest Additions.

Why Use the macOS Terminal?

Native Tool: The macOS terminal is a built-in tool, ensuring compatibility and stability.

Secure Connection: SSH provides a secure encrypted connection between the host and the VM.

Ease of Use: Copying and pasting text between the host and the VM can be done seamlessly using the terminal.

No Additional Software: This method avoids the need to install and configure the VirtualBox Guest Additions, reducing potential issues.

Screenshot Demonstration

Below is a screenshot demonstrating the process of copying the DCC website HTML code from a document on the macOS host and pasting it into a nano text document on the ubuntuserver VM via SSH.

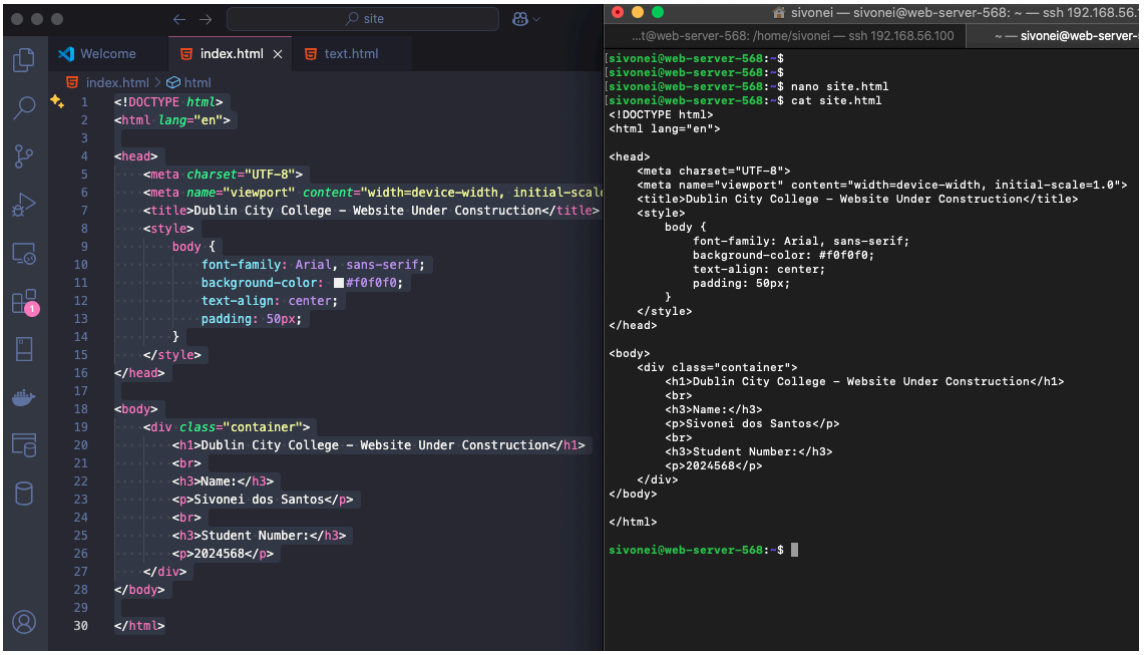

Figure 22: Screenshot Demonstration copy from macos to vm linux server

Conclusion

While the VirtualBox Guest Additions provide shared clipboard functionality, using the native macOS terminal with SSH offers a more secure and reliable alternative. This method eliminates the need for additional software and ensures seamless text transfer between the host and the VM. For environments where security and simplicity are priorities, this approach is highly recommended.

Differences Between Linux and Windows Boot Processes

The boot processes of Linux and Windows differ significantly in terms of bootloaders, kernel loading mechanisms, and essential files required to start the operating systems. Below is a detailed comparison of the two processes.

Linux Boot Process

1. BIOS/UEFI Initialization
The boot process begins with the BIOS (Basic Input/Output System) or UEFI (Unified Extensible Firmware Interface) firmware performing hardware checks and initializing system components. This step is common to both Linux and Windows (GeeksforGeeks, n.d.; Bisht, 2023).

2. Bootloader
In Linux, the GRUB (Grand Unified Bootloader) is typically used as the bootloader. GRUB is loaded from the Master Boot Record (MBR) or UEFI partition and presents the user with a menu to select the operating system or kernel to boot (GeeksforGeeks, n.d.).

3. Kernel Loading
GRUB loads the Linux kernel (commonly named vmlinuz) and the initial RAM disk (initrd.img) into memory. The initrd contains temporary files and drivers needed to mount the root filesystem (GeeksforGeeks, n.d.).

4. Root Filesystem Mounting
The kernel mounts the root filesystem as specified in the GRUB configuration file (/boot/grub/grub.cfg) and executes the /sbin/init program, which is the first process started by the kernel (Bisht, 2023).

5. Runlevel Programs
The system executes runlevel programs based on the defined runlevel. In modern Linux distributions, systemd is used to manage services and runlevels, replacing the traditional SysV init system (GeeksforGeeks, n.d.).

Essential Files for Ubuntu Linux
vmlinuz: The Linux kernel image.

initrd.img: The initial RAM disk image.

/boot/grub/grub.cfg: The GRUB configuration file.

Windows Boot Process

1. BIOS/UEFI Initialization
Similar to Linux, the Windows boot process begins with the BIOS or UEFI firmware initializing hardware and performing checks (Bisht, 2023).

2. Windows Boot Manager
The Windows Boot Manager (bootmgr) is loaded from the MBR or UEFI partition. It reads the Boot Configuration Data (BCD) to determine the location of the Windows kernel and other boot parameters (Bisht, 2023).

3. Kernel Loading
The Boot Manager loads the Windows kernel (ntoskrnl.exe) and essential drivers into memory. The kernel initializes the system and prepares it for user interaction (Bisht, 2023).

4. Session Initialization
The system initializes user sessions and loads the necessary services and applications. The Windows Session Manager (smss.exe) plays a key role in this process (Bisht, 2023).

Essential Files for Windows
ntoskrnl.exe: The Windows kernel.

BCD: Boot Configuration Data, which contains boot options.

bootmgr: The Windows Boot Manager executable.

Key Differences Between Linux and Windows Boot Processes

Bootloader:

Linux uses GRUB, while Windows uses the Windows Boot Manager (bootmgr).
GRUB is more flexible and allows multi-boot configurations, whereas Windows Boot Manager is primarily designed for Windows systems (GeeksforGeeks, n.d.).

Kernel Loading:

Linux loads the vmlinuz kernel and initrd.img, while Windows loads ntoskrnl.exe and essential drivers.
The initrd in Linux provides a temporary root filesystem, whereas Windows relies on the BCD for boot parameters (Bisht, 2023).

Configuration Files:

Linux uses /boot/grub/grub.cfg for boot configuration, while Windows uses the BCD stored in the \Boot directory (GeeksforGeeks, n.d.).

Initialization Process:

Linux uses systemd or init to manage services and runlevels, while Windows uses the Session Manager (smss.exe) to initialize user sessions (Bisht, 2023).

Part 5D: Forgotten password:

1. Reboot the VM and, during the GRUB splash-screen, hit ESC

2. With the list of options available, press: e
3. In the editor presented, add single at the end of a line similar to this one:

   linux /boot/vmlinux-3.13...-generic root=UUID=<uuid> ro

So that it looks like this:

   linux /boot/vmlinux-3.13...-generic root=UUID=<uuid> ro single

4. You will be presented with a console and the root prompt (#): use the passwd command to change your user's password:

   # passwd sivonei

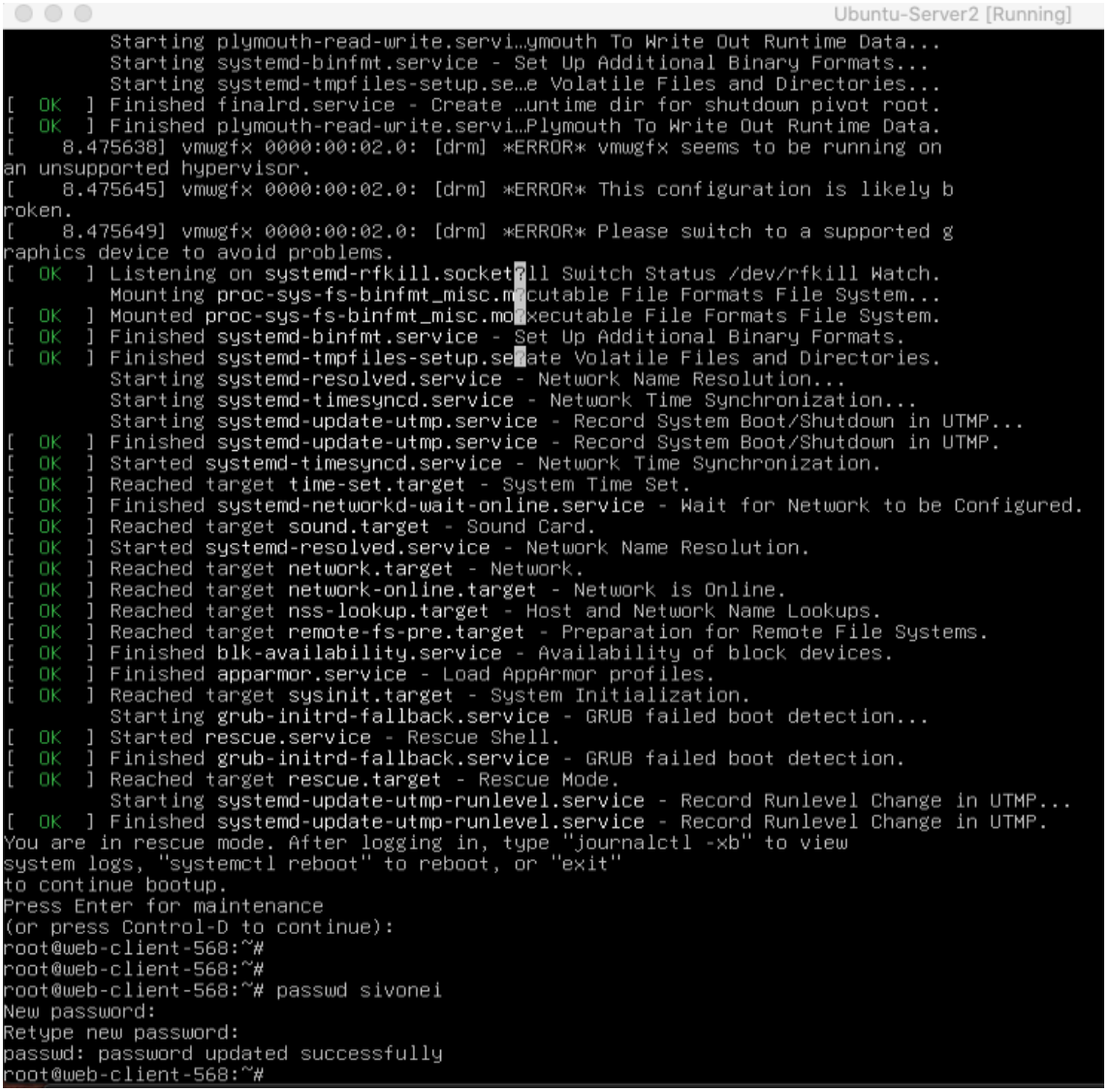

Figure 23: Password recovery

Student Reflection

This assignment gave me a great opportunity to learn more about virtualization, networking, and managing computer systems. By setting up a virtual Linux environment, configuring the network, and enabling features like SSH, I gained practical experience with tools such as Netplan, Apache, and firewalls like UFW and iptables.

One of the most important things I learned was how to set up a firewall, create rules, and remove them. This helped me improve my problem-solving skills and work more independently.

This experience also showed me how important it is to document my work and write clear, professional reports. By creating a detailed report with screenshots and references, I practiced explaining technical information in a way that is easy to understand. This is a key skill for working in IT.

As I get ready to graduate and start looking for jobs in IT, the skills I developed during this assignment will be very useful. Understanding virtualization and networking is important for roles like system administrator, network engineer, or cloud specialist. Also, being able to solve problems and adapt to new technologies will make me a stronger candidate in a fast-changing industry.

In conclusion, this assignment not only helped me improve my technical skills but also reminded me how important it is to keep learning in the IT field. I believe the knowledge and experience I gained from this project will help me succeed in my future career.

9. References:

1 - Alibaba Cloud (2023) What is Samba?. Available at: https://www.alibabacloud.com/blog/what-is-samba_599981 [Accessed: 25 Oct. 2023].

2 - Ubuntu (2023) Introduction to Samba. Available at: https://documentation.ubuntu.com/server/explanation/intro-to/samba/index.html [Accessed: 25 Oct. 2023].

3- Liquid Web (n.d.) How to Set Up a Firewall with iptables on Ubuntu 16.04. Available at: https://www.liquidweb.com/blog/set-firewall-using-iptables-ubuntu-16-04/ [Accessed: 25 Oct. 2023].

4 - UpCloud (n.d.) How to configure iptables on Ubuntu. Available at: https://upcloud.com/resources/tutorials/configure-iptables-ubuntu [Accessed: 25 Oct. 2023].

5 - GeeksforGeeks (n.d.) Difference Between Linux and Windows. Available at: https://www.geeksforgeeks.org/difference-between-linux-and-windows/ [Accessed: 25 Oct. 2023].

6 - Bisht, A. (2023) Understanding Boot Process: A Comprehensive Comparison Between Windows and Linux. Available at: https://www.linkedin.com/pulse/understanding-boot-process-comprehensive-comparison-windows-bisht-ysy5c/ [Accessed: 25 Oct. 2023].

7 - VirtualBox (n.d.) Chapter 4. Guest Additions. Available at: https://www.virtualbox.org/manual/ch04.html [Accessed: 25 Oct. 2023].

8 - Apple (n.d.) macOS Terminal User Guide. Available at: https://support.apple.com/guide/terminal/welcome/mac [Accessed: 25 Oct. 2023].

9 - Github (n.d.) HOW-TO Reset Virtualbox user password. Available at: https://github.com/massenz/HOW-TOs/blob/master/HOW-TO%20Reset%20Virtualbox%20user%20password.rst [Accessed: 25 Oct. 2023].