INTERNSHIP REPORT

[MAJOR PROJECT]

NAME: DHRUTHI V ARADHYA

DOMAIN: CYBER SECURITY

COMPANY: SMARTED

DURATION: 05TH APRIL 2025 – 05TH JULY 2025

TITLE: CONFIGURING AND SECURING A NETWORK INFRASTRUCTURE

CHAPTER 1

INTRODUCTION

OVERVIEW:

In today's digital landscape, organizations rely heavily on computer networks for communication, data transfer, and critical business operations. With this dependency comes the increasing risk of cyber threats, data breaches, and service disruptions. As a result, configuring and securing a network infrastructure has become a vital priority to ensure both operational efficiency and protection against malicious activity.

The project **configuring and securing a network infrastructure** was designed to provide hands-on experience in setting up a secure and efficient network infrastructure. It began with a thorough assessment and planning phase, where the existing network layout was analysed using diagrams and tools. The goal was to identify weak points in the current setup and define specific requirements related to performance, security, and scalability based on organizational needs.

The next phase involved selecting appropriate hardware components like routers, switches, and firewalls, along with essential software tools for monitoring and management. Using structured IP addressing, VLAN configurations, and routing protocols, the network was logically segmented and optimized for both internal and external traffic. Emphasis was placed on quality of service (QoS) to prioritize critical services, ensuring seamless data flow.

Network security was addressed through firewall configuration, role-based access control, encrypted communication protocols (like SSL/TLS), and intrusion prevention mechanisms. Regular monitoring, patch management, and incident response strategies were also implemented to maintain ongoing security and stability. Performance and security testing were conducted to validate the setup against defined benchmarks and risks.

This project provided practical exposure to core networking and cybersecurity concepts. It enabled the application of theoretical knowledge in a real-world context and helped develop a deeper understanding of how secure and scalable networks are planned, deployed, and maintained. The experience gained through this project lays a strong foundation for future roles in network administration and cybersecurity.

CHAPTER 2

OBJECTIVES

✓ 1. To configure basic network-level security on a Linux machine

This objective involves setting up fundamental protections on a Linux system to control and monitor how the system communicates with other devices. Network-level security includes configuring IP settings, securing ports, and managing incoming and outgoing traffic to reduce vulnerabilities and unauthorized access.

✓ 2. To implement a firewall that follows best practices

A firewall acts as a barrier between a trusted network and untrusted sources, such as the internet. This objective focuses on configuring the firewall (using UFW in this case) based on industry best practices, such as setting default-deny policies and explicitly allowing only required services. This minimizes attack surfaces and enhances system security.

✓ 3. To allow only essential services such as SSH, HTTP, and HTTPS

Certain services like SSH (for remote management), HTTP (for web traffic), and HTTPS (for secure web traffic) are critical for system operation and user access. This objective ensures that these specific ports are open so legitimate users and services can interact with the system, while all others are restricted.

✓ 4. To deny all other unauthorized access

By default, any non-essential or potentially harmful traffic should be blocked. This objective ensures that all unnecessary or suspicious connections are denied, reducing the risk of intrusions, exploits, or misuse of system resources.

CHAPTER 3

METHODOLOGY

The methodology followed in project was structured to simulate a secure network setup using a Linux environment. It began with updating the system and ensuring that the Uncomplicated Firewall (UFW) was installed. This tool was selected for its simplicity and effectiveness in managing firewall rules on Ubuntu-based systems. Using a Bash script helped automate the entire configuration process, making it reusable and consistent.

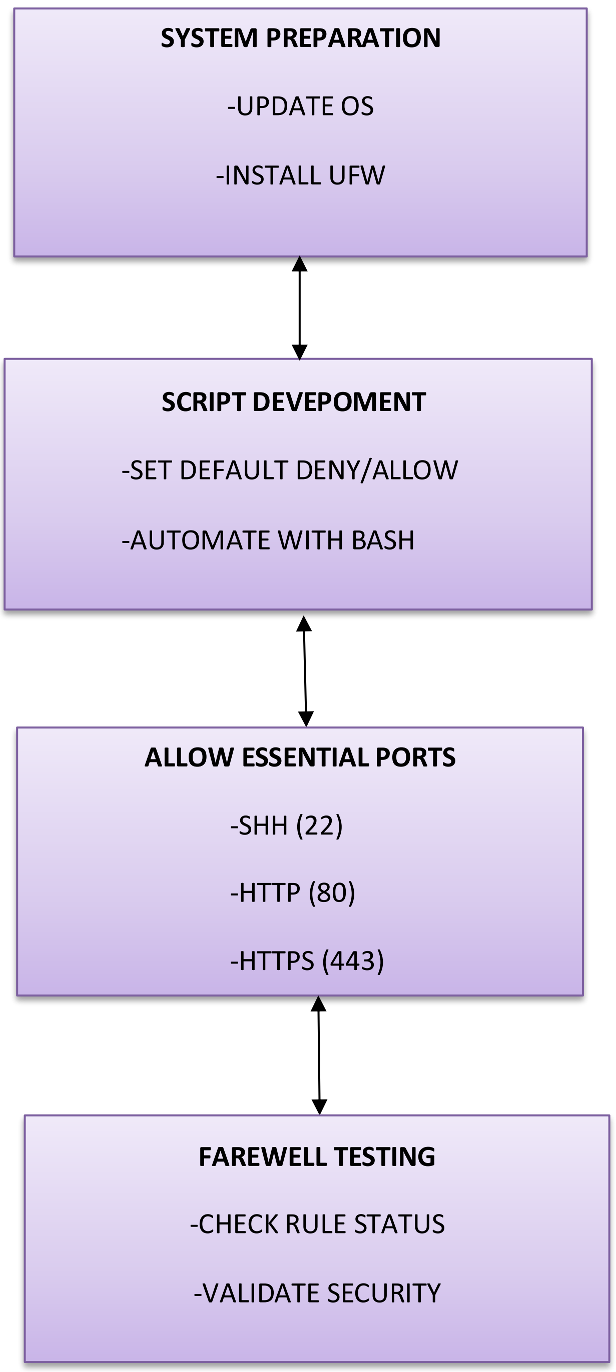
Once the system was prepared, the script was designed to apply industry-standard firewall settings. The default policy was set to deny all incoming connections and allow all outgoing traffic, which is a foundational principle in network security. This ensures that the system does not accept unsolicited requests while maintaining the ability to communicate with external servers for updates and services.

To maintain essential functionality, specific ports were opened deliberately: port 22 for SSH (remote access), port 80 for HTTP, and port 443 for HTTPS traffic. These services are commonly required for administrative and web access. No other services were exposed, significantly reducing the attack surface of the system. Each rule added to the firewall was tested for effectiveness immediately after deployment.

The firewall's status was verified using UFW commands to ensure that only the intended services were accessible. This verification step confirmed the accuracy and effectiveness of the firewall configuration. Overall, this hands-on approach offered practical exposure to scripting, system security, and firewall management in a controlled and educational setting.

In addition to the core configuration steps, careful attention was given to maintaining clarity and reusability within the script. Each command was organized logically and commented for ease of understanding, making it suitable for both manual execution and automation in a real-world environment. The script was designed to be system-agnostic within Ubuntu environments, ensuring that it could be used across multiple machines without modification. This approach reflects professional practices in network administration, where clear documentation and reusable tools are essential for scaling secure configurations across infrastructure.

3.1 FLOW CHART:



CHAPTER 4

TOOLS AND TECHNOLOGY

✓ 1. Ubuntu (Linux Operating System)

Ubuntu is a widely used open-source Linux distribution. It offers a secure and stable environment for networking and system administration tasks. In this project, Ubuntu was used as the platform for configuring the firewall and running shell scripts. Its built-in support for tools like UFW makes it ideal for security-related projects.

✓ 2. UFW (Uncomplicated Firewall)

UFW is a user-friendly command-line tool for managing the Linux iptables firewall. It simplifies the process of allowing or blocking network traffic through specified ports. UFW was used to implement firewall rules that control access to the system, such as allowing SSH (port 22), HTTP (port 80), and HTTPS (port 443), while denying all other unauthorized connections.

✓ 3. Bash (Shell Scripting)

Bash is the default command-line shell in most Linux distributions, including Ubuntu. In this project, a Bash script was written to automate the configuration of the firewall. The script updated the system, installed UFW, set default policies, allowed selected ports, and displayed the status of the firewall — reducing manual effort and ensuring consistency.

✓ 4. Terminal (Linux Command Line Interface)

The terminal was used to execute all system commands, run the shell script, and verify firewall settings. It provided direct interaction with the Ubuntu system, allowing real-time monitoring and control during the configuration process.

CHAPTER 5

CODE AND IMPLEMENTATION

```
#!/bin/bash

echo "=== Basic Network Security Setup ==="

# 1. Update and install ufw if not present

sudo apt update

sudo apt install ufw -y


# 2. Enable UFW (Ubuntu Firewall)

sudo ufw enable


# 3. Set default policies

sudo ufw default deny incoming

sudo ufw default allow outgoing


# 4. Allow essential services

sudo ufw allow ssh         # Allow SSH

sudo ufw allow 80/tcp      # Allow HTTP

sudo ufw allow 443/tcp     # Allow HTTPS


# 5. Deny everything else explicitly (optional)

# sudo ufw deny 23         # Deny Telnet
```

6. Show status

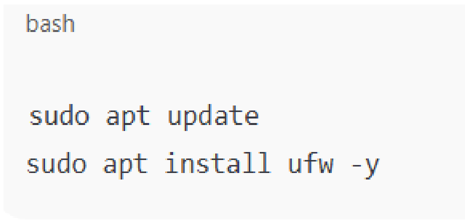
echo "=== Current Firewall Rules ==="

sudo ufw status verbose

echo "Network Security Configuration Complete."

5.1 FUNCTIONALITY:

Step 1: Update System & Install UFW

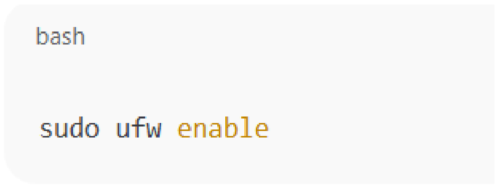


bash

```
sudo apt update
sudo apt install ufw -y
```

- sudo apt update refreshes the list of available packages from Ubuntu repositories.
- sudo apt install ufw -y installs the Uncomplicated Firewall (UFW) if it's not already installed. The -y flag auto-confirms the installation prompt.

Step 2: Enable UFW



bash

```
sudo ufw enable
```

- Activates the firewall. If this is the first time enabling it, you'll see a confirmation. From this point, UFW begins enforcing the firewall rules.

Step 3: Set Default Policies

```bash
sudo ufw default deny incoming
sudo ufw default allow outgoing
```

- deny incoming: blocks all incoming connections by default (unless explicitly allowed)
- allow outgoing: permits all outgoing traffic (e.g., downloading packages, browsing the internet).

Step 4: Allow Essential Services

```bash
sudo ufw allow ssh
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

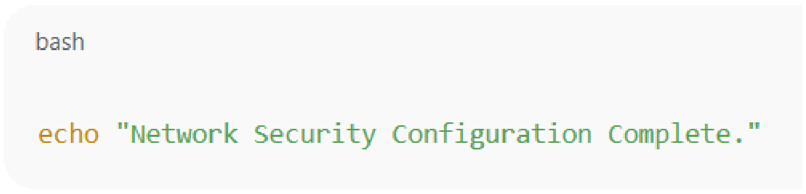
- ssh: opens port 22, allowing remote terminal access via SSH.
- 80/tcp: allows unencrypted web traffic (HTTP).
- 443/tcp: allows secure web traffic (HTTPS).

Step 5: Show Firewall Status

```bash
sudo ufw status verbose
```

- Displays all the current firewall rules in a detailed (verbose) format, including default policies and allowed services.

Step 6: Final Echo



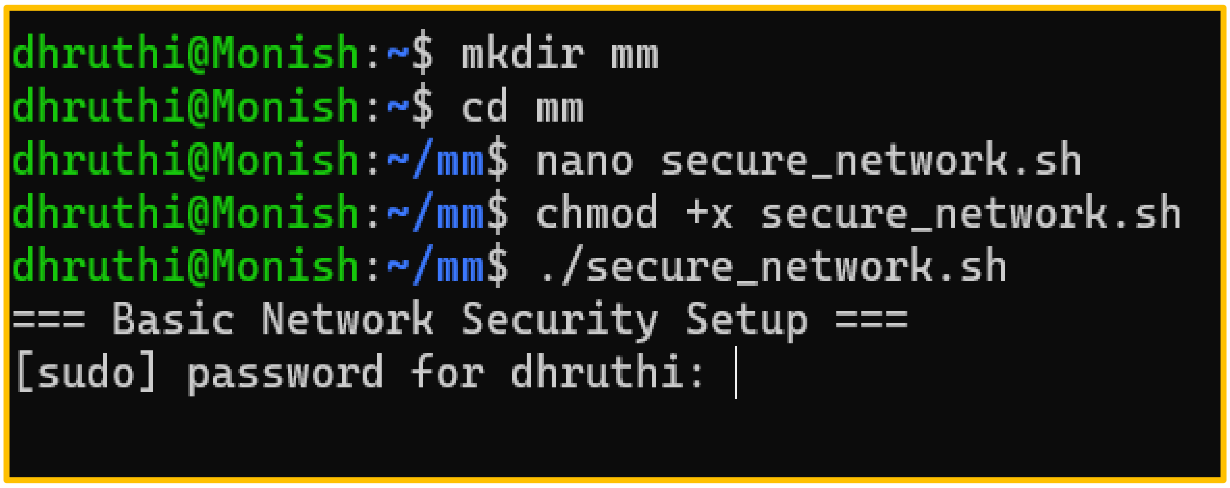
bash

```bash
echo "Network Security Configuration Complete."
```

- Prints a message confirming that the configuration process is finished.

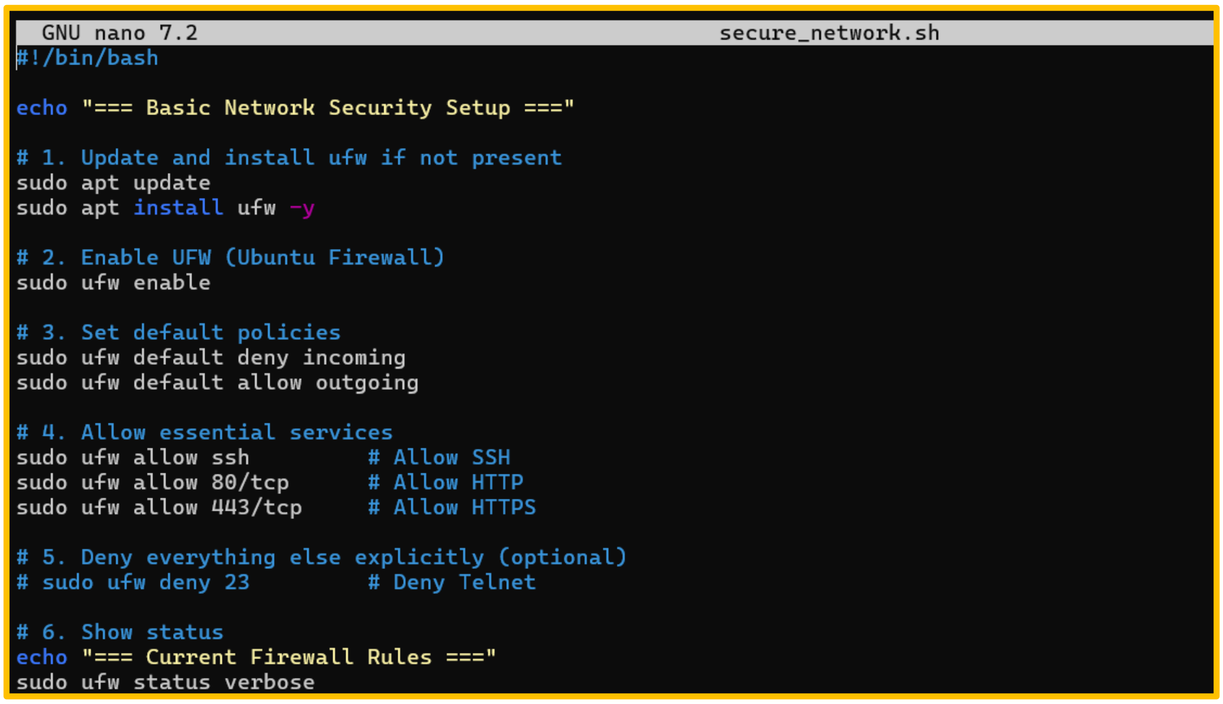
CHAPTER 6

RESULT



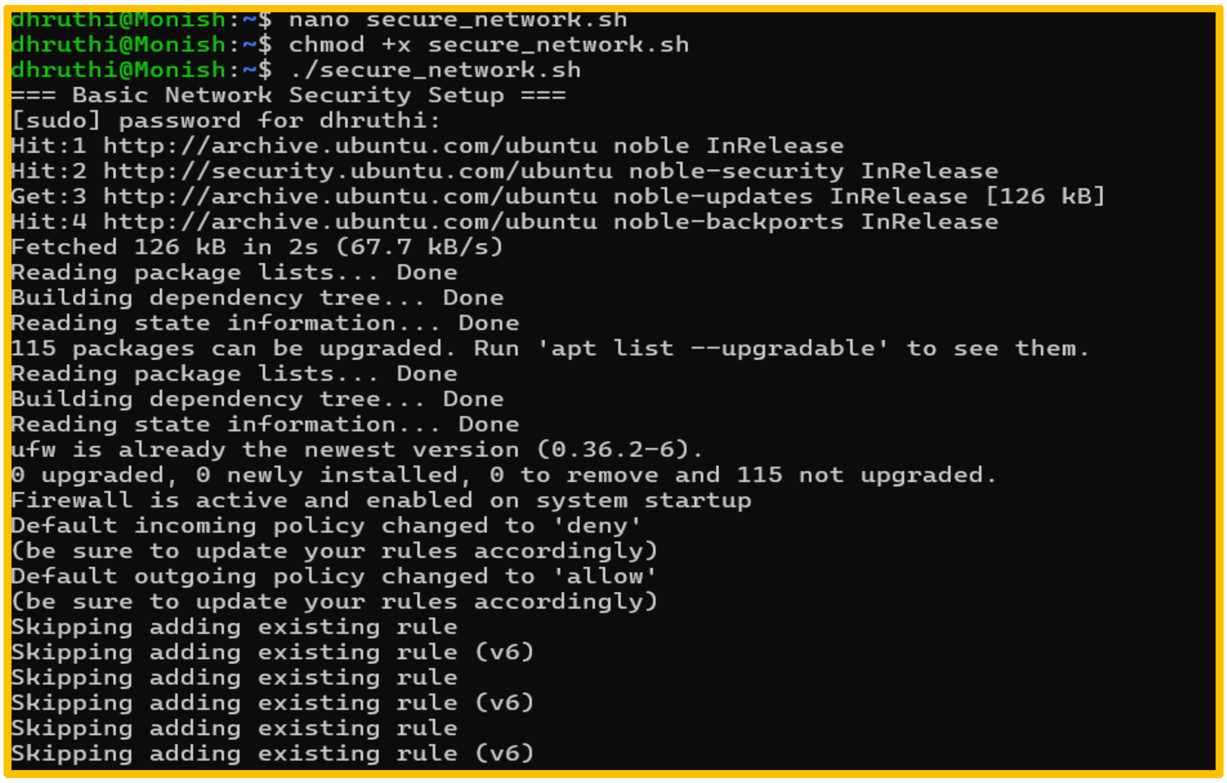
```
dhruthi@Monish:~$ mkdir mm
dhruthi@Monish:~$ cd mm
dhruthi@Monish:~/mm$ nano secure_network.sh
dhruthi@Monish:~/mm$ chmod +x secure_network.sh
dhruthi@Monish:~/mm$ ./secure_network.sh
=== Basic Network Security Setup ===
[sudo] password for dhruthi:
```

Fig: 6.1 Initialization



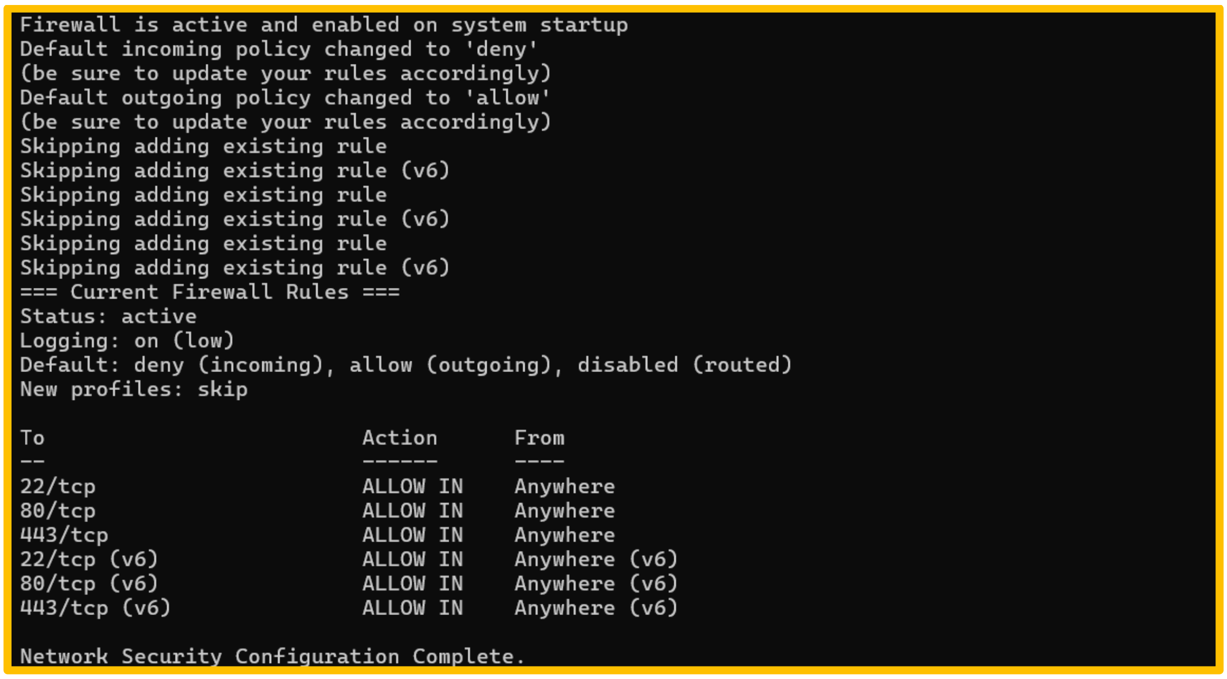
```
  GNU nano 7.2                                          secure_network.sh
#!/bin/bash

echo "=== Basic Network Security Setup ==="

# 1. Update and install ufw if not present
sudo apt update
sudo apt install ufw -y

# 2. Enable UFW (Ubuntu Firewall)
sudo ufw enable

# 3. Set default policies
sudo ufw default deny incoming
sudo ufw default allow outgoing

# 4. Allow essential services
sudo ufw allow ssh           # Allow SSH
sudo ufw allow 80/tcp        # Allow HTTP
sudo ufw allow 443/tcp       # Allow HTTPS

# 5. Deny everything else explicitly (optional)
# sudo ufw deny 23           # Deny Telnet

# 6. Show status
echo "=== Current Firewall Rules ==="
sudo ufw status verbose
```

Fig: 6.2 Security Setup



```
dhruthi@Monish:~$ nano secure_network.sh
dhruthi@Monish:~$ chmod +x secure_network.sh
dhruthi@Monish:~$ ./secure_network.sh
=== Basic Network Security Setup ===
[sudo] password for dhruthi:
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 2s (67.7 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
115 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
0 upgraded, 0 newly installed, 0 to remove and 115 not upgraded.
Firewall is active and enabled on system startup
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Fig: 6.3 Verifying Version



```
Firewall is active and enabled on system startup
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
Skipping adding existing rule
Skipping adding existing rule (v6)
=== Current Firewall Rules ===
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
443/tcp                    ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
443/tcp (v6)               ALLOW IN    Anywhere (v6)

Network Security Configuration Complete.
```

Fig: 6.4 Firewall is active and enabled on system startup

CONCLUSION

The project **configuring and securing a network infrastructure** provided practical knowledge and hands-on experience in configuring and securing a basic network infrastructure using a Linux environment. By focusing on firewall configuration through UFW (Uncomplicated Firewall), the project highlighted the importance of protecting systems from unauthorized access and maintaining control over network traffic.

The step-by-step approach—starting from system preparation, writing automation scripts, enabling firewall rules, and validating configurations—allowed for a deeper understanding of both theoretical concepts and real-world implementation. Automating the firewall setup using Bash scripting also demonstrated the power of scripting in enhancing efficiency, accuracy, and repeatability in system administration tasks.

By allowing only essential services such as SSH, HTTP, and HTTPS, and blocking all other incoming traffic, the project successfully implemented the principle of least privilege. This significantly reduced the attack surface and helped secure the Linux system in a controlled, measurable way.

Overall, the project emphasized the critical role of network security in any IT infrastructure. It laid a strong foundation for future learning in cybersecurity, system hardening, and enterprise-level network design. The experience gained will be valuable for pursuing advanced roles in network administration, security engineering.