**Mid test**

Course name: Open Source Platform and Network Administration

Course code: OSP201

|  |  |
| --- | --- |
| Student name: | Huỳnh Ngọc Quang |
| Student ID: | SE181838 |

**Part 1: 6 marks**

**Question 1:** In the context of network security and user account protection on Linux operating systems, the sudo tool is crucial. Discuss the benefits of using sudo compared to direct login with the root account and provide examples of how to configure sudo permissions for users to perform specific tasks without full root access. Consider including how to configure sudo in distributed environments and how to adjust sudo policies to enhance security.

Advantages over Direct Root Login:

Audit Trail: Each command executed using sudo is logged, which helps in auditing and tracing actions back to the individual user.

Minimize Risk: By not logging in as root directly, the 'root' account remains disabled, minimizing the risk of brute-force attacks or accidental misuse of administrative privileges.

Configuring Sudo in Distributed Environments:

Use tools like Ansible, Puppet, or Chef to manage and distribute sudoers configurations across multiple systems.

Ensure consistency in sudo policies across systems to avoid security gaps.

Enhancing Security with Sudo Policies:

Configure sudo to require a password every time, reducing the risk of abuse if a user leaves a terminal unattended.

Limit command execution by defining specific commands that users can execute in the sudoers file.

Use groups in the sudoers file to manage permissions more efficiently and ensure that users have only the necessary privileges.

**Question 2:** In the field of computer security, Discretionary Access Control (DAC) plays a crucial role in managing file and directory access permissions. Analyze the methods and importance of using DAC in a Linux environment. Include examples of how to set up and manage access permissions using commands like chmod, chown, and setfacl. Additionally, discuss the challenges DAC might face and how organizations can address these challenges to enhance system security.

**Using DAC with Commands:**

**chmod: Changes the permissions of a file or directory.**

**chown: Changes the owner of a file or directory.**

**setfacl: Sets file access control lists to provide detailed and flexible permissions.**

**Challenges and Mitigation:**

**Overly Permissive Defaults: Sometimes, systems have overly permissive settings by default. Regular audits and user training can help mitigate this.**

**Complex Management: As the number of files and directories grows, managing permissions can become complex. Using automated tools or scripts can help manage these settings effectively.**

**Question 3:** Encryption and secure shell (SSH) hardening are essential components of securing data transmission and access control in Linux environments. Discuss the significance of implementing encryption and SSH hardening techniques. Provide detailed explanations of the steps involved in encrypting data at rest and in transit, including the use of tools such as GNU Privacy Guard (GPG) for data encryption and the configuration steps for hardening SSH, such as disabling root login and using key-based authentication. Additionally, evaluate the potential vulnerabilities that these measures help mitigate and any remaining risks that system administrators need to be aware of.

Implementing encryption and SSH hardening techniques is crucial for securing data transmission and access control in Linux environments. Here’s a detailed breakdown:

Data Encryption

Data at Rest

Using GPG for Data Encryption:

Installation: Install GPG using sudo apt-get install gnupg or similar command for your Linux distribution.

Generate a Key Pair:

gpg --full-generate-key

Encrypt a File:

gpg -c filename

This command encrypts the file using a symmetric cipher.

Decrypt a File:

gpg filename.gpg

Encrypting Entire Disks or Partitions:

Use tools like LUKS (Linux Unified Key Setup):

sudo cryptsetup luksFormat /dev/sdX

sudo cryptsetup luksOpen /dev/sdX encrypted

sudo mkfs.ext4 /dev/mapper/encrypted

Data in Transit

TLS/SSL Encryption:

Use OpenSSL to generate certificates for encrypting web traffic.

Example commands to generate a self-signed certificate:

openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365

SSH Encryption:

SSH inherently encrypts traffic between client and server.

SSH Hardening Techniques

Disabling Root Login:

Edit the SSH configuration file /etc/ssh/sshd\_config:

PermitRootLogin no

Restart SSH service:

sudo systemctl restart sshd

Using Key-Based Authentication:

Generate SSH Key Pair:

ssh-keygen -t rsa -b 4096

Copy Public Key to Server:

ssh-copy-id user@server

Configure SSH to Use Public Key Authentication:

Ensure PubkeyAuthentication yes is set in /etc/ssh/sshd\_config.

Disabling Password Authentication:

In /etc/ssh/sshd\_config, set:

PasswordAuthentication no

Restart SSH service:

sudo systemctl restart sshd

Change Default SSH Port:

Edit /etc/ssh/sshd\_config to use a non-standard port:

Port 2222

Adjust firewall rules accordingly.

Enforce Strong Ciphers and Algorithms:

In /etc/ssh/sshd\_config, specify strong ciphers:

ciphers aes256-ctr,aes192-ctr,aes128-ctr

Disable weak ciphers and algorithms.

Potential Vulnerabilities Mitigated

Unauthorized Access: Disabling root login and using key-based authentication significantly reduce the risk of unauthorized access.

Brute Force Attacks: Disabling password authentication and using non-standard ports help mitigate brute force attacks.

Man-in-the-Middle (MitM) Attacks: Encryption ensures data integrity and confidentiality, protecting against MitM attacks.

Remaining Risks

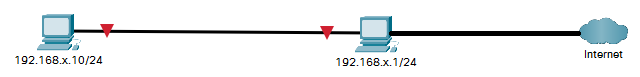
Key Management: Loss or theft of private keys can compromise security.

Software Vulnerabilities: Flaws in encryption software or SSH implementation can still be exploited.

Human Factors: Misconfiguration or lax security practices by users or administrators can lead to vulnerabilities.

By implementing these measures, system administrators can significantly enhance the security of their Linux environments, though continuous monitoring and updates are essential to mitigate emerging threat

**Part 2:** **4 marks**

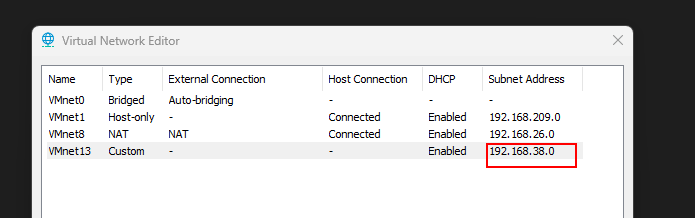
****

Note: x is the two last number of your student ID

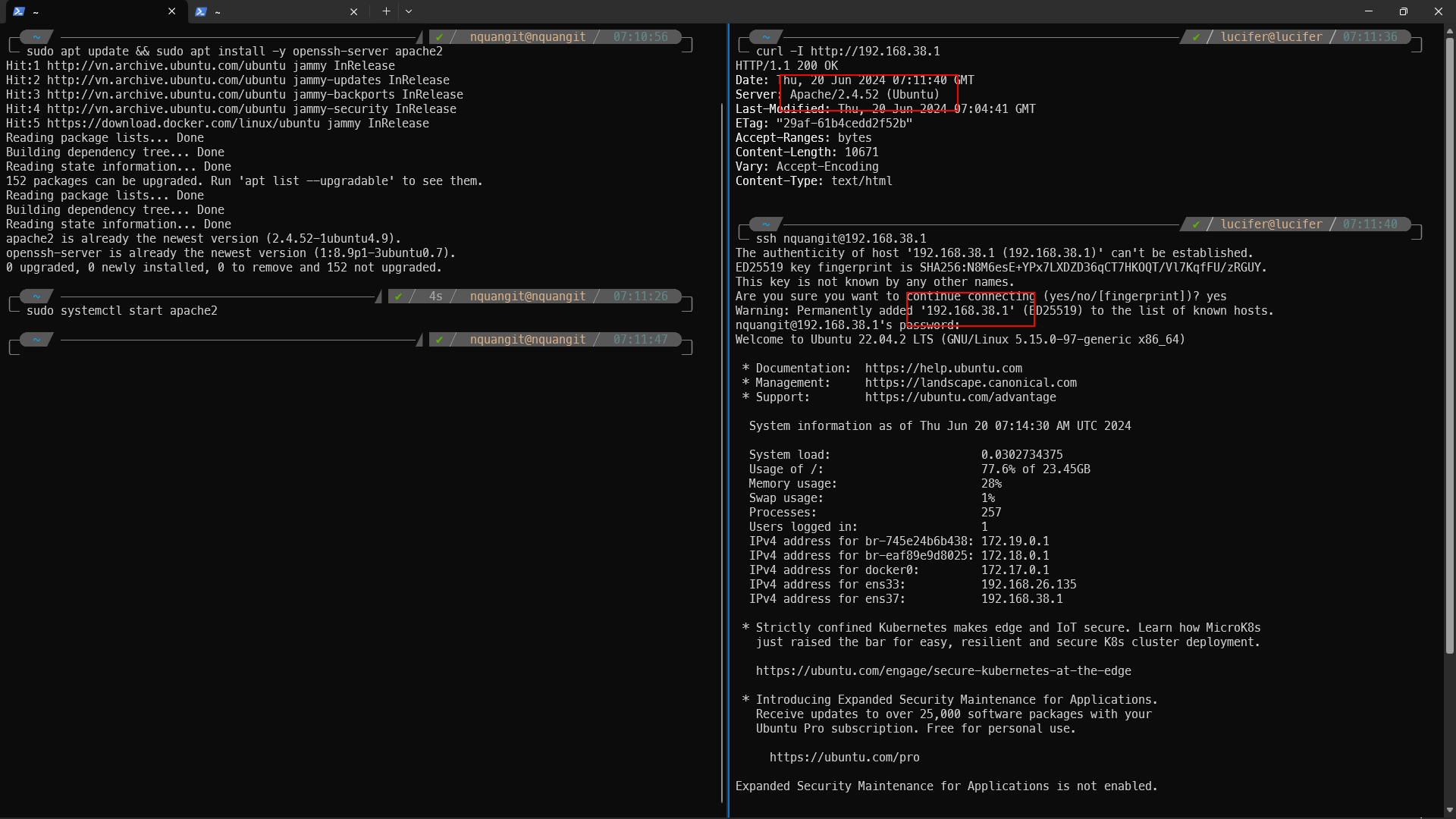
* Install and configure the SSH service on 192.168.x.1
* Configure to ensure the security of the SSH service.
* Install and configure the HTTP service on 192.168.x.1
* Configure to ensure the security of the HTTP service.

END

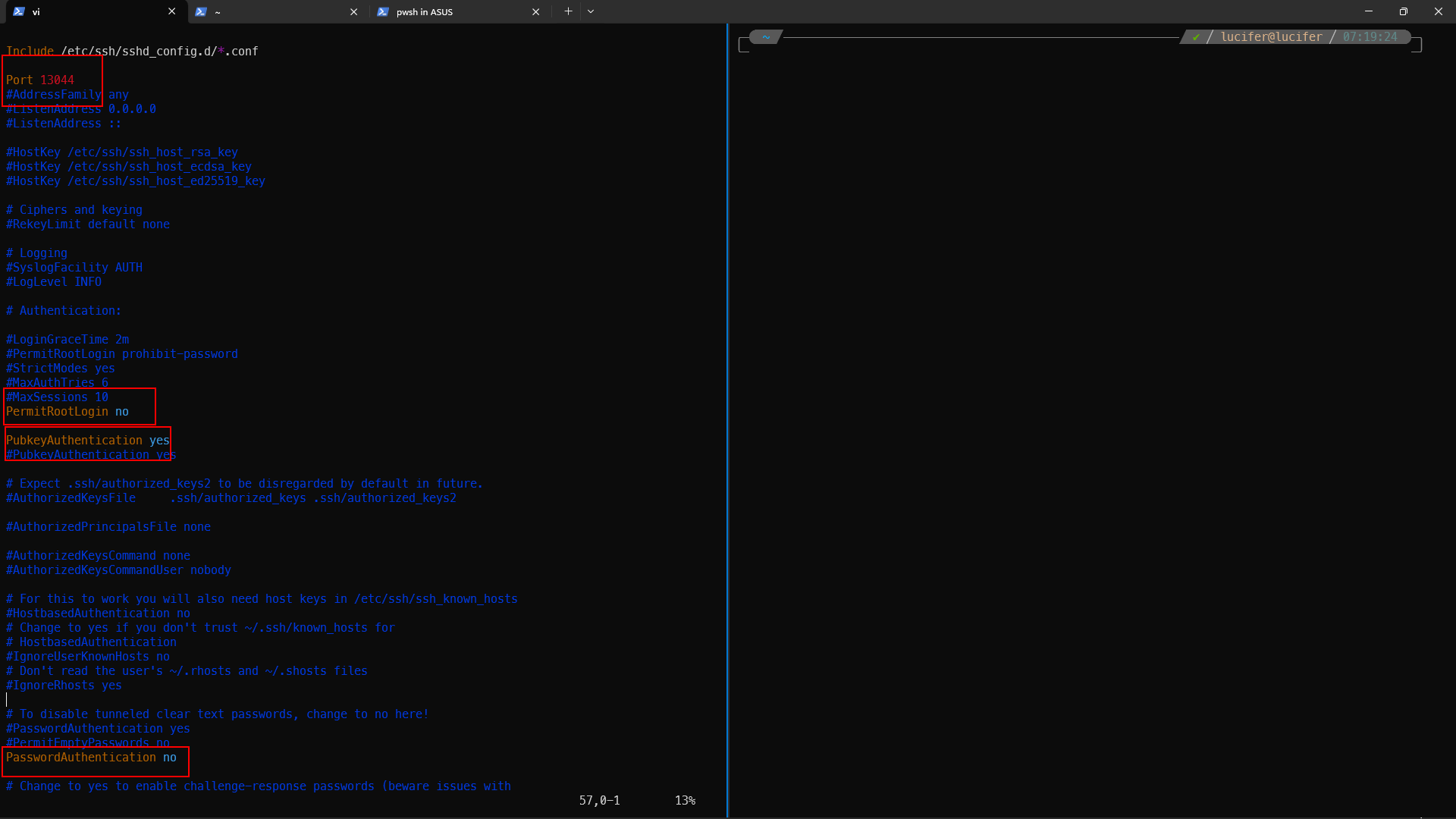
* Create a virtual network with the network address 192.168.38.0/24



* Create 2 virtual machine and connect to the custom virtual network
* The machine with IP x.1 will be Ubuntu Server and another will be BlackArch
* Configure IP address for 2 machine
* A screenshot of a computer

  Description automatically generated
* Ping successfully.
* 
* Install web and ssh service
* Testing curl the web server and ssh to ssh server from the machine with IP x.1

SECURE SSH



* Change the ssh port
* Prevent root login
* Enable ssh login with ssh-key
* Disable ssh login with the normal password authentication.