Day 11-Permissions and Security in Linux

Linux Security

Linux security covers a range of topics essential for system administration and DevOps. Below is a concise overview.

Access Control

- Access control uses user and password-based authentication to determine who can log into the system.

PAM (Pluggable Authentication Modules)

- PAM is used to authenticate users to various services and programs in Linux.
- Configuration files for PAM are located in `/etc/pam.d/.`

Network Security

- Tools like `iptables` and `firewalld` control access to services running on the system.
- Example:

```
 sudo firewall-cmd --list-all
sudo iptables -L
```

SSH – Secure Shell

- SSH is used for secure remote access over an untrusted network.
- SSH hardening includes:
  - Disabling root login
  - Allowing only specific users
  - Enforcing key-based authentication
- Example:

```
grep -i ^root /etc/passwd     # Check if root account is active
sudo vi /etc/ssh/sshd_config # Modify SSH settings
```

SELinux – Security-Enhanced Linux

- SELinux enforces security policies to isolate applications from each other.
- Example:

```
getenforce       # Check SELinux status
setenforce 1     # Temporarily enable enforcing mode
```

User and Group Accounts

- Every user has:
  - Username
  - UID (User ID)
  - GID (Group ID)
  - Home directory
  - Default shell

- View this information:
  `id <username>`
  `cat /etc/passwd | grep <username>`
- User info is stored in: `/etc/passwd`
- Group info is stored in: `/etc/group`
- Superuser (root):
  - UID = 0
  - Has unrestricted system access

System and Service Accounts

- Created during installation or when software is installed
- Typically have UIDs < 100 or between 500–1000
- Used by services like `sshd` and `mail`
- Often do not have home directories or login shells

Viewing Users

- List currently logged-in users:
  `who`
- Show historical login data:
  `last`
- Switch user:
  `su <username>`
- Gain root/admin privileges:
  `sudo <command>`

Sudoers Configuration

- sudo permissions are defined in: `/etc/sudoers`
- Use `visudo` to edit this file safely

Disabling Root Login

- Check root entry:
  `grep -i ^root /etc/passwd`
- In SSH config:
  `sudo vi /etc/ssh/sshd_config`
  `PermitRootLogin no`

Access Control Files and User Management

Access control in Linux relies heavily on system files stored under the /etc directory.

Important Access Control Files

- `/etc/passwd`
  - Stores basic user information like username, UID, GID, home directory, and shell.
  - Format:
    - `USERNAME:PASSWORD:UID:GID:GECOS:HOMEDIR:SHELL`
  - GECOS contains user info such as full name, location, and phone number.

- `/etc/shadow`
  - Stores hashed passwords and password aging policies.
  - Format:
    - `USERNAME:PASSWORD:LASTCHANGE:MINAGE:MAXAGE:WARN:INACTIVE:EXPDATE`
- `/etc/group`
  - Stores information about system groups.
  - Format:
    - `GROUPNAME:PASSWORD:GID:MEMBERS`

User Management Commands

- Create a new user:
  - `useradd <username>`
- Set or change a password:
  - `passwd <username>`
- View current user:
  - `whoami`

Useful options with useradd:

- `-c "<comment>"      # Add custom comment (e.g., full name)`
- `-d <home_dir>       # Set custom home directory`
- `-e <expiry_date>    # Set account expiry date`
- `-g <GID>            # Assign user to a primary group by GID`
- `-G <group1,group2>  # Assign user to multiple secondary groups`
- `-s <shell>          # Set user login shell`
- `-u <UID>            # Assign specific UID`
- Example:
  - `useradd -c "Dev User" -d /dev/home/fatima -e 2025-12-31 -g dev -G docker, wheel -s /bin/bash -u 1101 fatima`
- Delete a user account:
  - `userdel <username>`

Group Management Commands

- Create a group:
  - `groupadd <groupname>`
  - Example:
    - `groupadd -g 1001 dev`
- Delete a group:
  - `groupdel <groupname>`

File Permissions

File Type Identifiers

The first character of `ls -l` output indicates the file type:

- d : Directory
- - : Regular file

- c : Character device file
- l : Symbolic link
- s : Socket
- p : Named pipe
- b : Block device
- Example:
  - `ls -l sample_script.sh   # -rwxrwxr-x fatima fatima ... sample_script.sh`

Permission Structure

The permission section is divided into three parts:

`-rwxrwxr-x`

1. Owner (user): rwx
2. Group: rwx
3. Others (world): r-x

Each permission has an octal value:

	Permission	Symbol	Octal Value
1	read	r	4
2	write	w	2
3	execute	x	1
4	no access	-	0

Changing Permissions with chmod

Two ways to set permissions:

- Symbolic Mode:
  - u = user, g = group, o = others, a = all
  - `chmod u+rwx,g+r,o-x test.txt`
- Numeric Mode:
  - `chmod 777 test.txt   # Full permissions to all`
  - `chmod 555 test.txt   # Read and execute for all, no write`
  - `chmod 660 test.txt   # User and group can read/write, others no access`

Changing Ownership

- Change owner and group:
  - `chown owner:group file`
- Change only owner:
  - `chown owner file`
- Change only group:
  - `chgrp group file`
- Examples:
  - `chown fatima:dev team_report.txt`

- `chown root /opt/script.sh`
- `chgrp developers app.log`

SH (Secure Shell)

SSH is used to securely log in to and execute commands on a remote machine over an untrusted network.

- Basic usage:
  - `ssh <username>@<IP/hostname>`
- By default, SSH operates on port 22.
- Example:
  - `ssh fatima@192.168.1.100`

Passwordless SSH Login

To set up passwordless login:

- Generate a key pair on your local machine:
  - `ssh-keygen -t rsa`
- Public key is saved at:
  - `/home/<user_name>.ssh/id_rsa.pub`
- Private key is saved at:
  - `/home/<user_name>/.ssh/id_rsa`
- Copy the public key to the remote server:
  - `ssh-copy-id <user_name>@<server_ip>`
- The public key will be added to:
  - `/home/<user_name>/.ssh/authorized_keys`
- Now you can connect without entering a password:
  - `ssh <user_name@<server_ip>`

iptables – Linux Firewall Utility

- iptables is a command-line tool for configuring the Linux kernel's netfilter firewall.
- Installation (for Debian/Ubuntu)
  - ` sudo apt install iptables`
- View current rules:
  - ` iptables -L`
- Three default chains:
  - INPUT : Handles incoming traffic to the host
  - OUTPUT : Handles outgoing traffic from the host
  - FORWARD : Handles traffic routed through the host
- Add a rule:
  - `iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 22 -j ACCEP`
- Flags:
  - -A : Append a rule to a chain
  - -I : Insert a rule at the top
  - -p : Protocol (e.g., tcp, udp)
  - -s : Source IP or network

    - -d : Destination IP (optional)
    - --dport : Destination port
    - -j : Action (ACCEPT, DROP, REJECT)
    - Examples:
    - Allow HTTP traffic:
        - `iptables -A INPUT -p tcp --dport 80 -j ACCEPT`
    - Allow SSH from a specific subnet:
        - `iptables -A INPUT -p tcp -s 10.0.0.0/8 --dport 22 -j ACCEPT`
    - Drop all other SSH traffic:
        - `iptables -A INPUT -p tcp --dport 22 -j DROP`
    - Delete a rule:
        - `iptables -D OUTPUT <rule_number>`
        - Example:
            - `iptables -D OUTPUT 1`
- Rules are applied in order. First match wins.
- Use `iptables-save` to view complete rule set
- Use `iptables -F` to flush (remove) all rules