DevOps Shack

Security in Linux

[Click Here To Enrol To Batch-6 | DevOps & Cloud DevOps]

Linux security encompasses a range of practices and tools designed to protect the system and its data from unauthorized access, misuse, or harm. Security in Linux involves managing user permissions, setting up firewalls, encrypting data, and more. Here is a detailed overview of key security aspects in Linux with examples.

1. User and Group Management

Proper user and group management is fundamental to Linux security. It ensures that users have appropriate permissions and access rights.

Adding and Managing Users

- Add a new user:

sudo adduser username

- Set or change a user password:

sudo passwd username

- Add a user to a group:

sudo usermod -aG groupname username

Example

Add user 'john'

sudo adduser john

Set password for 'john'

sudo passwd john

Add 'john' to 'sudo' group

sudo usermod -aG sudo john

2. File Permissions and Ownership

File permissions and ownership control who can read, write, or execute a file.

Viewing Permissions

- Use ls -l to view file permissions:

ls -l filename

Example output:

-rw-r--r-- 1 user group 0 Aug 6 10:00 filename

Changing Permissions

- Change permissions using chmod:

chmod 755 filename # Sets rwxr-xr-x

chmod u+x filename # Adds execute permission for the owner

Changing Ownership

- Change ownership using chown:

chown user:group filename

Example

Create a file

touch example.txt

Change permissions to rwxr-xr--

chmod 754 example.txt

Change ownership to user 'john' and group 'developers'

sudo chown john:developers example.txt

3. Pluggable Authentication Modules (PAM)

PAM provides a way to develop programs that are independent of authentication schemes. It allows administrators to configure authentication policies without modifying the application.

PAM Configuration Files

- Located in /etc/pam.d/

Example: Restricting SSH Access

Edit /etc/pam.d/sshd to add a restriction:

auth required pam_listfile.so item=user sense=deny file=/etc/ssh/deniedusers onerr=succeed

Create /etc/ssh/deniedusers and add usernames to restrict:

baduser

4. Firewalls

A firewall controls incoming and outgoing network traffic based on predetermined security rules.

Using UFW (Uncomplicated Firewall)

- Enable UFW:

sudo ufw enable

- Allow SSH connections:

sudo ufw allow ssh

- Allow a specific port:

sudo ufw allow 8080

- Deny a specific port:

sudo ufw deny 23

- Check the status:

sudo ufw status

Example

Enable UFW

sudo ufw enable

Allow HTTP traffic

sudo ufw allow 80

Allow HTTPS traffic

sudo ufw allow 443

Deny telnet traffic

sudo ufw deny 23

Check UFW status

sudo ufw status

5. SELinux (Security-Enhanced Linux)

SELinux provides a mechanism for supporting access control security policies.

SELinux Modes

- **Enforcing:** SELinux policy is enforced.
- **Permissive:** SELinux prints warnings instead of enforcing.
- **Disabled:** SELinux is turned off.

Checking SELinux Status

sestatus

Changing SELinux Mode

- Edit /etc/selinux/config and set SELINUX=enforcing, permissive, or disabled.
- To change the mode without rebooting:

sudo setenforce 1 # Enforcing mode

sudo setenforce 0 # Permissive mode

6. SSH (Secure Shell)

SSH is used for securely connecting to remote servers.

SSH Key Pair Authentication

- Generate SSH key pair:

ssh-keygen -t rsa -b 4096 -C "your_email@example.com"

- Copy the public key to the remote server:

ssh-copy-id user@remote_host

Example

Generate SSH key pair

ssh-keygen -t rsa -b 4096 -C "john@example.com"

Copy public key to remote server

ssh-copy-id john@192.168.1.100

Connect to remote server

ssh john@192.168.1.100

7. Data Encryption

Encrypting data protects it from unauthorized access.

Encrypting Files with GPG

- Encrypt a file:

gpg -c filename

- Decrypt a file:

gpg filename.gpg

Example

Encrypt a file

gpg -c confidential.txt

Decrypt the file

gpg confidential.txt.gpg

8. Auditing with Auditd

Auditd is used for tracking security-relevant information.

Installing Auditd

sudo apt install auditd # For Debian/Ubuntu

sudo yum install audit # For CentOS/RHEL

Configuring Auditd

- Edit rules in /etc/audit/audit.rules.
- Example rule to monitor file access:

-w /etc/passwd -p wa -k passwd_changes

This rule logs write and attribute changes to /etc/passwd.

Viewing Audit Logs

sudo ausearch -f /etc/passwd

sudo aureport

9. Intrusion Detection Systems (IDS)

IDS tools like AIDE and Tripwire monitor system integrity.

Installing AIDE

sudo apt install aide # For Debian/Ubuntu

sudo yum install aide # For CentOS/RHEL

Initializing AIDE Database

sudo aideinit

Checking System Integrity

sudo aide --check

10. Securing Network Services

Ensure that unnecessary network services are disabled to reduce the attack surface.

Listing Active Services

sudo systemctl list-units --type=service --state=running

Disabling Unnecessary Services

sudo systemctl disable service_name

sudo systemctl stop service_name

Example of Comprehensive Security Configuration

Scenario: Secure an Ubuntu Server

1. **User Management:**

sudo adduser alice

sudo passwd alice

sudo usermod -aG sudo alice

2. **File Permissions:**

touch /var/log/secure.log

chmod 640 /var/log/secure.log

chown root:adm /var/log/secure.log

3. **UFW Firewall:**

sudo ufw enable

sudo ufw allow ssh

sudo ufw allow http

sudo ufw allow https

```
sudo ufw deny 23
sudo ufw status
```

4. **SELinux Configuration (if applicable):**

```
sudo setenforce 1
echo "SELINUX=enforcing" | sudo tee -a /etc/selinux/config
```

5. **SSH Configuration:**

```
ssh-keygen -t rsa -b 4096 -C "alice@example.com"
ssh-copy-id alice@192.168.1.100
```

6. **File Encryption:**

```
gpg -c confidential.txt
gpg confidential.txt.gpg
```

7. **Auditd Configuration:**

```
sudo apt install auditd
echo "-w /etc/passwd -p wa -k passwd_changes" | sudo tee -a /etc/audit/audit.rules
sudo systemctl restart auditd
sudo ausearch -f /etc/passwd
sudo aureport
```

8. **AIDE Configuration:**

```
sudo apt install aide
sudo aideinit
sudo cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db
sudo aide --check
```

9. **Disable Unnecessary Services:**

```
sudo systemctl list-units --type=service --state=running
sudo systemctl disable apache2
sudo systemctl stop apache2
```

Summary

Linux security is a multi-faceted field that requires attention to user management, file permissions, network security, data encryption, and more. By employing a combination of these techniques and tools, administrators can significantly enhance the security posture of their Linux systems. Regular monitoring, auditing, and updating of security practices are essential to protect against evolving threats.