

➤ **Exercice 1 : Analyse de l'environnement**

1. Quelle est la version du noyau Linux utilisée ?

```
vagrant@docker:~$ uname -r
3.10.0-1160.119.1.el7.x86_64
vagrant@docker:~$ uname -a
Linux docker 3.10.0-1160.119.1.el7.x86_64 #1 SMP Tue Jun 4 14:43:51 UTC 2024 x86_64 x86_64 x86_64 GNU/L
vagrant@docker:~$ hostnamectl
  Static hostname: docker
    Icon name: computer-vm
      Chassis: vm
     Machine ID: 17eba3d662974e43a4dbe4e894f51510
      Boot ID: f07f727f39224aa6ab11432374d3033f
  Virtualization: kvm
Operating System: CentOS Linux 7 (Core)
      CPE OS Name: cpe:/o:centos:centos:7
        Kernel: Linux 3.10.0-1160.119.1.el7.x86_64
      Architecture: x86-64
```

2. Analyse des services de sécurité actifs

➤ **Services de pare-feu**

a) Le service de pare-feu par défaut sur CentOS est firewalld :

cde :**sudo systemctl status firewalld**

```
vagrant@docker:~$ sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
```

Vérifier les règles actives de firewalld (zones, ports ouverts) : **sudo firewall-cmd --list-all**

```
X vagrant@docker:~$ sudo firewall-cmd --list-all
public (active)
target: default
icmp-block-inversion: no
interfaces: enp0s3 enp0s8
sources:
services: dhcpv6-client ssh
ports:
protocols:
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
```

**sudo firewall-cmd --list-all-zones**

```
sources:
services:
ports:
protocols:
forward: no
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:

work
target: default
icmp-block-inversion: no
interfaces:
sources:
services: cockpit dhcpv6-client ssh
ports:
protocols:
forward: no
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
```

Ces commandes vous montreront quelles zones sont actives, quels ports sont ouverts, et quelles interfaces sont associées à quelles zones, pour comprendre la surface d'attaque réseau.

➤ **SELinux (Security-Enhanced Linux)**

SELinux est un mécanisme de contrôle d'accès obligatoire (MAC) qui ajoute une couche de sécurité supplémentaire au-delà des permissions Linux traditionnelles.

Vérifier l'état de SELinux : `sestatus`

La sortie vous indiquera si SELinux est enforcing (appliqué), permissive (avertit mais n'interdit pas), ou disabled (désactivé).

Pour une sécurité maximale, il devrait être enforcing.

```
vagrant@docker: ~]# sestatus
SELinux status:          enabled
SELinuxfs mount:         /sys/fs/selinux
SELinux root directory:  /etc/selinux
Loaded policy name:     targeted
Current mode:           permissive
Mode from config file: permissive
Policy MLS status:      enabled
Policy deny_unknown status: allowed
Max kernel policy version: 31
```

### ➤ Service d'audit du système (auditd)

auditd est le démon de l'audit système qui enregistre les événements liés à la sécurité dans les journaux.

Vérifier l'état de auditd : `sudo systemctl status auditd`

```
vagrant@docker: ~]# sudo systemctl status auditd
● auditd.service - Security Auditing Service
  Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; vendor preset: enabled)
  Active: active (running) since Sat 2025-06-28 10:44:20 UTC; 1h 19min ago
    Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
  Process: 651 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUCCESS)
  Process: 639 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
 Main PID: 642 (auditd)
   Tasks: 2
  Memory: 2.7M
   CGroup: /system.slice/auditd.service
           └─642 /sbin/auditd

Jun 28 10:44:20 docker systemd[1]: Starting Security Auditing Service...
Jun 28 10:44:20 docker auditd[642]: Started dispatcher: /sbin/audispd pid: 644
Jun 28 10:44:20 docker auditd[642]: Init complete, auditd 2.8.5 listening for events (startup state enab
Jun 28 10:44:20 docker augenrules[651]: /sbin/augenrules: No change
Jun 28 10:44:20 docker systemd[1]: Started Security Auditing Service.
```

### Lister les règles d'audit actives : `sudo auditctl -l`

Ceci affichera les règles configurées pour surveiller des activités spécifiques (accès aux fichiers sensibles, changements de configuration, etc.).

```
vagrant@docker: ~]# sudo auditctl -l
No rules
```

### ➤ SSH (Secure Shell)

SSH est le service le plus courant pour l'accès à distance sécurisé, Vérifier l'état du service SSH (`sshd`) : `sudo systemctl status sshd`

```
vagrant@docker: ~]# sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor pres
  Active: active (running) since Sat 2025-06-28 10:44:26 UTC; 1h 22min ago
    Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1016 (sshd)
    Tasks: 1
   Memory: 3.2M
    CGroup: /system.slice/sshd.service
            └─1016 /usr/sbin/sshd -D

Jun 28 10:44:26 docker systemd[1]: Starting OpenSSH server daemon...
Jun 28 10:44:26 docker sshd[1016]: Server listening on 0.0.0.0 port 22.
Jun 28 10:44:26 docker sshd[1016]: Server listening on :: port 22.
Jun 28 10:44:26 docker systemd[1]: Started OpenSSH server daemon.
Jun 28 10:53:29 docker sshd[1625]: Accepted password for vagrant from 192.168
```

```
sudo cat /etc/ssh/sshd_config | grep -E "PermitRootLogin|PasswordAuthentication|Port|AllowUsers|DenyUsers"
```

Examinez des paramètres comme `PermitRootLogin no`, `PasswordAuthentication no` (préférer les clés SSH), et le port sur lequel SSH écoute (par défaut 22).

```
vagrant@docker: ~]# X vagrant@docker: ~]# sudo cat /etc/ssh/sshd_config | grep -E "PermitRootLogin|PasswordAuthentication|Port|AllowUsers|DenyUsers"
#Port 22
#PermitRootLogin yes
#PasswordAuthentication yes
#PasswordAuthentication yes
## PasswordAuthentication. Depending on your PAM configuration,
# the setting of "PermitRootLogin without-password".
# PAM authentication, then enable this but set PasswordAuthentication
#GatewayPorts no
```

## ➤ Fail2ban (pour la protection contre les attaques par force brute)

Si Fail2ban est installé, il surveille les journaux pour détecter les tentatives de connexion échouées et bannit temporairement les adresses IP malveillantes.

Vérifier l'état de Fail2ban : **sudo systemctl status fail2ban**

```
vagrant@docker: ~ $ sudo systemctl status fail2ban
● fail2ban.service could not be found.
```

Vérifier les jails (règles de surveillance) actives de Fail2ban : **sudo fail2ban-client status**

```
sudo fail2ban-client status <nom_de_la_jail> # Ex: sudo fail2ban-client status sshd
```

## ➤ Services réseau (ports ouverts)

Identifier les services qui écoutent sur des ports réseau est crucial pour évaluer la surface d'attaque.

Lister tous les ports ouverts et les processus associés : **sudo ss -tulnp**

```
X vagrant@docker: ~ $ sudo ss -tulnp
Netid State      Recv-Q Send-Q      Local Address:Port          Peer Address:Port
udp  UNCONN     0      0          127.0.0.1:323
users:(("chronyd",pid=680,fd=5))
udp  UNCONN     0      0          *:842
users:(("rpcbind",pid=669,fd=7))
tcp  UNCONN     0      0          *:68
users:(("dhclient",pid=777,fd=6))
tcp  UNCONN     0      0          *:68
users:(("dhclient",pid=776,fd=6))
tcp  UNCONN     0      0          *:111
users:(("rpcbind",pid=669,fd=6))
tcp  UNCONN     0      0          [::1]:323
users:(("chronyd",pid=680,fd=6))
tcp  UNCONN     0      0          [::]:842
users:(("rpcbind",pid=669,fd=10))
tcp  UNCONN     0      0          [::]:111
users:(("rpcbind",pid=669,fd=9))
tcp  LISTEN    0      128         *:111
users:(("rpcbind",pid=669,fd=8))
tcp  LISTEN    0      128         *:22
users:(("sshd",pid=1016,fd=3))
tcp  LISTEN    0      100        127.0.0.1:25
users:(("master",pid=1240,fd=13))
tcp  LISTEN    0      128         [::]:111
users:(("rpcbind",pid=669,fd=11))
tcp  LISTEN    0      128         [::]:22
users:(("sshd",pid=1016,fd=4))
```

## ➤ Services généraux systemd

Pour une vue d'ensemble des services actifs et de leur type : Lister tous les services active (running) :

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

```
UNIT           LOAD   ACTIVE SUB   DESCRIPTION
audited.service loaded  active running Security Auditing Service
chronynd.service loaded  active running NTP client/server
containerd.service loaded  active running containerd container runtime
crond.service  loaded  active running Command Scheduler
dbus.service   loaded  active running D-Bus System Message Bus
docker.service loaded  active running Docker Application Container Engine
firewalld.service loaded  active running firewalld - dynamic firewall daemon
getty@tty1.service loaded  active running Getty on tty1
gssproxy.service loaded  active running GSSAPI Proxy Daemon
irqbalance.service loaded  active running irqbalance daemon
lvm2-lvmetad.service loaded  active running LVM2 metadata daemon
NetworkManager.service loaded  active running Network Manager
polkit.service  loaded  active running Authorization Manager
postfix.service loaded  active running Postfix Mail Transport Agent
rpcbind.service loaded  active running RPC bind service
rsyslog.service loaded  active running System Logging Service
sshd.service   loaded  active running OpenSSH server daemon
systemd-journald.service loaded  active running Journal Service
systemd-logind.service loaded  active running Login Service
systemd-udevd.service loaded  active running udev Kernel Device Manager
tuned.service   loaded  active running Dynamic System Tuning Daemon

LOAD  = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB   = The low-level unit activation state, values depend on unit type.

21 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

Ceci listera tous les services qui sont actuellement en cours d'exécution. Vous devrez ensuite passer en revue cette liste pour identifier ceux qui sont liés à la sécurité. Lister tous les services activés au démarrage (peuvent être liés à la sécurité) :

**systemctl list-unit-files --type=service --state=enabled**

```
vagrant@docker: ~ % systemctl list-unit-files --type=service --state=enabled
UNIT FILE                                     STATE
audited.service                                enabled
autovt@.service                               enabled
chronynd.service                             enabled
crond.service                                 enabled
dbus-org.freedesktop.nm-dispatcher.service   enabled
docker.service                                enabled
getty@.service                                 enabled
irqbalance.service                           enabled
kdump.service                                 enabled
lvm2-monitor.service                         enabled
microcode.service                           enabled
NetworkManager-dispatcher.service           enabled
NetworkManager-wait-online.service          enabled
NetworkManager.service                       enabled
postfix.service                               enabled
rhel-autorelabel-mark.service                enabled
rhel-autorelabel.service                     enabled
rhel-configure.service                      enabled
rhel-dmesg.service                          enabled
rhel-domainname.service                     enabled
rhel-import-state.service                   enabled
rhel-loadmodules.service                   enabled
rhel-readonly.service                      enabled
rpcbind.service                            enabled
rsyslog.service                            enabled
sshd.service                                enabled
systemd-readahead-collect.service          enabled
```

## Exercice 2 : Recherche de fichiers et analyse de logs

1. **Rechercher tous les fichiers contenant des secrets potentiels** ;Mots-clés à rechercher sans tenir compte de la casse : password, api\_key, token, secret\_key.

Commande :

```
Sudo grep -riE 'password|api_key|token|secret_key' /etc /var/www 2>/dev/null
```

```
x vagrant@docker: ~ $ sudo grep -riE 'password|api_key|token|secret_key' /etc /var /www 2>/dev/null
```

```
/var/log/boot.log-20250628:[ OK ] Started Forward Password Requests to Wall Directo
Binary file /var/cache/yum/x86_64/7/base/gen/primary_db.sqlite matches
Binary file /var/cache/yum/x86_64/7/extras/gen/primary_db.sqlite matches
Binary file /var/cache/yum/x86_64/7/updates/gen/primary_db.sqlite matches
Binary file /var/cache/yum/x86_64/7/epel/gen/primary_db.sqlite matches
Binary file /var/cache/man/index.db matches
/var/db/Makefile:      echo "Warning: The shadow password database $@"; \
```

## 2. Analyser les logs d'authentification

Consulter le fichier de log en temps réel ou les dernières entrées :

Commande : `sudo tail -n 30 /var/log/secure`

```
vagrant@docker: ~ sudo tail -n 30 /var/log/secure
Jun 28 12:05:21 docker sudo: pam_unix(sudo:session): session closed for user root
Jun 28 12:06:58 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl status sshd
Jun 28 12:07:00 docker sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:07:59 docker sudo: pam_unix(sudo:session): session closed for user root
Jun 28 12:08:54 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/cat /etc/ssh/sshd_configJun 28 12:08:54 docker su
do: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:08:54 docker sudo: pam_unix(sudo:session): session closed for user root
Jun 28 12:09:32 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/cat /etc/ssh/sshd_configJun 28 12:09:32 docker su
do: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:24:45 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl status fail2ban
Jun 28 12:24:45 docker sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:32:00 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/sbin/ss -tulnp
Jun 28 12:32:00 docker sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:32:00 docker sudo: pam_unix(sudo:session): session closed for user root
Jun 28 12:44:04 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/grep -riE password|api_key|token|secret_key /etc/
var/www
Jun 28 12:44:04 docker sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:44:05 docker sudo: pam_unix(sudo:session): session closed for user root
Jun 28 12:44:35 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/cd /etc/var/www
Jun 28 12:44:35 docker sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:46:06 docker sudo: vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/grep -riE password|api_key|token|secret_key /etc/
var/www
Jun 28 12:46:06 docker sudo: pam_unix(sudo:session): session closed for user root
Jun 28 12:46:23 docker sudo: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jun 28 12:46:23 docker sudo: pam_unix(sudo:session): session closed for user root
```

## 3. Listez les fichiers modifiés il y a moins de N jours

Pour lister les fichiers modifiés au cours des 7 derniers jours (par exemple) : `find /etc -type f -mtime -7`

```
vagrant@docker: ~ sudo find /etc -type f -mtime -7
/etc/fstab
/etc/resolv.conf
/etc/pki/ca-trust/extracted/java/cacerts
/etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt
/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem
/etc/pki/ca-trust/extracted/pem/email-ca-bundle.pem
/etc/pki/ca-trust/extracted/pem/objsign-ca-bundle.pem
/etc/yum.repos.d/CentOS-Base.repo
/etc/yum.repos.d/CentOS-CR.repo
/etc/yum.repos.d/CentOS-fasttrack.repo
/etc/yum.repos.d/CentOS-x86_64-kernel.repo
/etc/yum.repos.d/epel-testing.repo
/etc/yum.repos.d/epel.repo
/etc/yum.repos.d/docker-ce.repo
/etc/group
/etc/gshadow
/etc/passwd-
/etc/hosts
/etc/shadow-
/etc/udev/rules.d/60-vboxadd.rules
/etc/udev/hwdb.bin
/etc/passwd
/etc/shadow
/etc/shells
/etc/sysconfig/network-scripts/ifcfg-lo
/etc/sysconfig/network-scripts/ifcfg-enp0s3
/etc/sysconfig/network-scripts/ifcfg-enp0s8
```

Le dossier `/etc` est considéré comme un **dossier critique** dans les systèmes d'exploitation basés sur Linux (comme CentOS) pour plusieurs raisons fondamentales :

Le répertoire contient la quasi-totalité des fichiers de configuration essentiels et spécifiques à la machine pour l'ensemble du système d'exploitation et de ses services. Cela inclut :/etc

**Configuration réseau** (, , network-scriptshostsresolv.conf)

**Gestion des utilisateurs et groupes** (, , passwdshadowgroup)

**Configuration des services système** (, , systemdcronlogrotatersyslog)

**Configuration des applications** (Apache, Nginx, MySQL, SSH, Postfix, etc.)

**Configuration du démarrage** ( pour les points de montage, pour le chargeur de démarrage)fstabgrub

Variables d'environnement système

### **Impact sur le Fonctionnement :**

Toute modification incorrecte ou suppression de fichiers dans /etc peut avoir des conséquences désastreuses et immédiates :

**Empêcher le démarrage du système** : Une erreur dans /etc/fstab ou la configuration de GRUB peut rendre le système inamorçable./etc/fstab

**Briser la connectivité réseau** : Des problèmes dans les fichiers de configuration réseau peuvent empêcher le serveur de communiquer avec d'autres machines ou Internet.

**Désactiver des services cruciaux** : La modification ou la suppression de fichiers de configuration de services comme SSH, Apache ou la base de données peut rendre ces services inutilisables, interrompant des applications ou l'accès à distance.

**Compromettre la sécurité** : Une mauvaise configuration des fichiers de sécurité (comme /etc/, ou les paramètres de pare-feu) peut ouvrir des failles de sécurité majeures.sudoerssshd\_config

**Rendre les logins impossibles** : La corruption de ou peut empêcher les utilisateurs de se connecter./etc/passwd ;/etc/shadow

**Absence de Données Utilisateur :**

Contrairement à qui contient les données des utilisateurs, ou qui contient les données variables (logs, spoule), ou qui contient des applications tierces, est purement dédié à la configuration du système. Cela signifie que la perte ou la corruption de ce répertoire affecte la capacité du système à fonctionner plutôt que la perte de données spécifiques à l'utilisateur./home/var/opt/etc

### **Exercice 3 : Analyse des permissions sensibles**

- **Quels sont les groupes à privilèges sur votre système ?**

**root** (ou UID 0):C'est le super-utilisateur. Le groupe root est généralement le groupe principal de l'utilisateur root. Il a des droits illimités sur le système. Bien que techniquement un UID, il est souvent associé à un groupe du même nom..

**wheel (ou sudo)**:Ce groupe est crucial pour l'administration. Les utilisateurs membres du groupe wheel peuvent exécuter des commandes avec les privilèges de root en utilisant la commande sudo

**adm**: Historiquement utilisé pour les tâches d'administration et l'accès aux fichiers de log.Privilèges : Lecture des fichiers de log système (par exemple, dans /var/log).

**cdrom**:Permet aux utilisateurs d'accéder et de monter les lecteurs CD/DVD-ROM/périphériques optiques.

**disk**:Permet l'accès direct aux périphériques disque bruts. C'est un groupe très puissant et potentiellement dangereux s'il est mal utilisé, car il permet d'écrire directement sur les partitions. Généralement, seuls les utilisateurs root devraient en être membres.

**lp**: Anciennement utilisé pour gérer les imprimantes locales. Moins courant aujourd'hui avec l'avènement de CUPS et des imprimantes réseau.

**mail**: Gérer les files d'attente de courrier et les boîtes aux lettres.

**operator**:Un groupe destiné aux opérateurs système qui peuvent effectuer certaines tâches de maintenance de base (par exemple, arrêt du système, sauvegarde) sans avoir les droits complets de root.

**sshusers** (ou un nom similaire):Si AllowGroups sshusers est défini dans /etc/ssh/sshd\_config, seuls les membres de ce groupe peuvent se connecter.

**systemd-journal**:Permet aux utilisateurs de lire les journaux du système via journalctl.

**docker**:Si Docker est installé, les membres de ce groupe peuvent exécuter des commandes Docker sans utiliser sudo.

- **Quels utilisateurs en font partie ? : Sudo cat /etc/group**

```
vagrant@docker ~ % su
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:vagrant
cdrom:x:11:
mail:x:12:postfix
man:x:15:
dialout:x:18:
floppy:x:19:
games:x:20:
tape:x:33:
video:x:39:
ftp:x:50:
lock:x:54:
audio:x:63:
nobody:x:99:
users:x:100:
utmp:x:22:
utempter:x:35:
input:x:999:
systemd-journal:x:190:
systemd-network:x:192:
dbus:x:81:
```

- Y a-t-il des utilisateurs ayant un accès root implicite ou explicite ?

L'utilisateur root a un User ID (UID) de 0. Tout autre utilisateur avec un UID de 0 a également un accès root complet et direct. C'est très rare et généralement déconseillé pour des raisons de sécurité. Commande : `grep ':0:' /etc/passwd`

```
vagrant@docker ~ % grep ':0:' /etc/passwd
root:x:0:0:root:/root:/bin/bash
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/bin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
operator:x:11:0:operator:/root:/sbin/nologin
```

- Quels utilisateurs ont accès à un shell de connexion ? (*Distinguer les utilisateurs humains des comptes de service.*) : `awk -F: '{print $1 ":" $7}' /etc/passwd`

```
vagrant@docker ~ % awk -F: '{print $1 ":" $7}' /etc/passwd
root:/bin/bash
bin:/sbin/nologin
daemon:/sbin/nologin
adm:/sbin/nologin
lp:/sbin/nologin
sync:/bin/sync
shutdown:/sbin/shutdown
halt:/sbin/halt
mail:/sbin/nologin
operator:/sbin/nologin
games:/sbin/nologin
ftp:/sbin/nologin
nobody:/sbin/nologin
systemd-network:/sbin/nologin
ibus:/sbin/nologin
polkitd:/sbin/nologin
shd:/sbin/nologin
postfix:/sbin/nologin
chrony:/sbin/nologin
vagrant:/bin/zsh
pc:/sbin/nologin
pcuser:/sbin/nologin
nfsnobody:/sbin/nologin
ss:/sbin/nologin
boxadd:/bin/false
```

- Lister les utilisateurs qui peuvent se connecter, vous pouvez filtrer ces shells :

```
grep -vE '(/sbin/nologin|/bin/false)$' /etc/passwd | awk -F: '{ print $1 ":" $7 }'
```

```
vagrant@docker ~ % grep -vE '(/sbin/nologin | /bin/false)$' /etc/passwd | awk -F: '{print $1 ":" $7}'
root:/bin/bash
bin:/sbin/nologin
daemon:/sbin/nologin
adm:/sbin/nologin
lp:/sbin/nologin
sync:/bin/sync
shutdown:/sbin/shutdown
halt:/sbin/halt
mail:/sbin/nologin
operator:/sbin/nologin
games:/sbin/nologin
ftp:/sbin/nologin
nobody:/sbin/nologin
systemd-network:/sbin/nologin
dbus:/sbin/nologin
polkitd:/sbin/nologin
sshd:/sbin/nologin
postfix:/sbin/nologin
chrony:/sbin/nologin
vagrant:/bin/zsh
rpc:/sbin/nologin
rpcuser:/sbin/nologin
nfsnobody:/sbin/nologin
tss:/sbin/nologin
vboxadd:/bin/false
```

## 🔗 Partie 2 – Mini-Projet (2h)

*En tant que Expert DevSecOps nous sommes affectés à un mandat pour auditer et sécuriser un serveur Linux qui hébergera une application web sensible.*

*Tâches : effectuer un audit initial, corriger les problèmes, automatiser les vérifications, et valider le déploiement de l’application web.*

### Tâches à réaliser

#### 1.1. Audit initial

a) Collecte des informations système (OS, kernel, services) :

```
[vagrant@centos8 ~]$ uname -r
4.18.0-348.7.1.el8_5.x86_64
[vagrant@centos8 ~]$ cat /etc/*release*
CentOS Linux release 8.5.2111
Derived from Red Hat Enterprise Linux 8.5
NAME="CentOS Linux"
VERSION="8"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="8"
PLATFORM_ID="platform:el8"
PRETTY_NAME="CentOS Linux 8"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/:o:centos:centos:8"
HOME_URL="https://centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
CENTOS_MANTISBT_PROJECT="CentOS-8"
CENTOS_MANTISBT_PROJECT_VERSION="8"
CentOS Linux release 8.5.2111
CentOS Linux release 8.5.2111
cpe:/:o:centos:centos:8
[vagrant@centos8 ~]$ uname -a
Linux centos8.localdomain 4.18.0-348.7.1.el8_5.x86_64 #1 SMP Wed Dec 22 13:25:12 UTC 2021 x86_64 x86_64 GNU/Linux
```

Cde: `systemctl list-units --type=service`

```
by EAZYTraining
[vagrant@docker ~ %] sudo systemctl list-units --type=service
UNIT                      LOAD  ACTIVE SUB   DESCRIPTION
audited.service            loaded active running Security Auditing Service
chronyd.service            loaded active running NTP client/server
containerd.service         loaded active running container runtime
crond.service              loaded active running Command Scheduler
dbus.service               loaded active running D-Bus System Message Bus
docker.service              loaded active running Docker Application Container Engine
getty@tty1.service          loaded active running Getty on tty1
gssproxy.service            loaded active running GSSAPI Proxy Daemon
irqbalance.service          loaded active running irqbalance daemon
kdump.service               loaded failed failed Crash recovery kernel arming
kmod-static-nodes.service   loaded active exited Create list of required static devices
lvm2-lvmetad.service        loaded active running LVM2 metadata daemon
lvm2-monitor.service        loaded active exited Monitoring of LVM2 mirrors, snapshot
lvm2-pvscan@8:2.service     loaded active exited LVM2 PV scan on device 8:2
network.service              loaded active exited LSB: Bring up/down networking
NetworkManager-wait-online.service loaded active exited Network Manager Wait Online
NetworkManager.service       loaded active running Network Manager
nginx.service                loaded failed failed The nginx HTTP and reverse proxy server
```

```
Systemctl list-units --type=service --state=running -no-pager | grep -E '\.service'
```

```
x vagrant@docker ~ % sudo systemctl list-units --type=service --state=running --no-pager | grep -E
auditd.service          loaded active running Security Auditing Service
chrony.service           loaded active running NTP client/server
containerd.service       loaded active running containerd container runtime
crond.service            loaded active running Command Scheduler
dbus.service              loaded active running D-Bus System Message Bus
docker.service            loaded active running Docker Application Container Engine
getty@tty1.service        loaded active running Getty on tty1
gssproxy.service          loaded active running GSSAPI Proxy Daemon
irqbalance.service        loaded active running irqbalance daemon
lvm2-lvmetad.service     loaded active running LVM2 metadata daemon
NetworkManager-dispatcher.service loaded active running Network Manager Script Dispatcher Service
NetworkManager.service    loaded active running Network Manager
polkit.service             loaded active running Authorization Manager
postfix.service            loaded active running Postfix Mail Transport Agent
rpcbind.service            loaded active running RPC bini service
rsyslog.service            loaded active running System Logging Service
sshd.service               loaded active running OpenSSH server daemon
systemd-journald.service loaded active running Journal Service
```

b) Vérification des utilisateurs, groupes, permissions sensibles

cde1 : grep ':0:' /etc/passwd

```
x vagrant@docker ~ % sudo grep ':0:' /etc/passwd
root:x:0:0:root:/root:/bin/bash
sync:x:5:0:sync:/sbin:/sync
shutdown:x:6:0:shutdown:/sbin:/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
operator:x:11:0:operator:/root:/sbin/nologin
```

Cde2 : grep -E '^ (wheel|disk|kmem|adm|staff)' /etc/group

```
x vagrant@docker ~ % sudo grep -E '^ (wheel|disk|kmem|adm|staff)' /etc/group
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

Cde3 : find / -type f -perm -o+w -print 2>/dev/null utilisée pour trouver tous les fichiers sur votre système Linux qui sont "world-writable", c'est-à-dire qui peuvent être écrits par n'importe quel utilisateur sur le système.

```
/proc/2/task/2/attr/fscreate
/proc/2/task/2/attr/keycreate
/proc/2/task/2/attr/sockcreate
/proc/2/attr/current
/proc/2/attr/exec
/proc/2/attr/fscreate
/proc/2/attr/keycreate
/proc/2/attr/sockcreate
/proc/4/task/4/attr/current
/proc/4/task/4/attr/exec
/proc/4/task/4/attr/fscreate
/proc/4/task/4/attr/keycreate
/proc/4/task/4/attr/sockcreate
```

### c- Audit des Fichiers et Permissions Sensibles (récapitulatif et compléments)

**find / -type f -perm -o+w -exec ls -ld {} \; 2>/dev/null** : pour identifier les fichiers "world-writable" (accessibles en écriture par tout le monde) sur votre système Linux.

```
x vagrant@docker ~ % sudo find / -type f -perm -o+w -exec ls -ld {} \; 2>/dev/null
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/sys/kernel/ns_last_pid
-rw-rw-rw-. 1 root root 0 Jul 1 02:14 /proc/1/task/1/attr/current
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/task/1/attr/exec
-rw-rw-rw-. 1 root root 0 Jul 1 02:14 /proc/1/task/1/attr/fscreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/task/1/attr/keycreate
-rw-rw-rw-. 1 root root 0 Jul 1 02:14 /proc/1/task/1/attr/sockcreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/attr/current
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/attr/exec
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/attr/fscreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/attr/keycreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/1/attr/sockcreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/task/2/attr/current
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/task/2/attr/exec
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/task/2/attr/fscreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/task/2/attr/keycreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/task/2/attr/sockcreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/attr/current
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/attr/exec
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/attr/fscreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/2/attr/sockcreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/task/4/attr/current
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/task/4/attr/exec
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/task/4/attr/fscreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/task/4/attr/keycreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/task/4/attr/sockcreate
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/attr/current
-rw-rw-rw-. 1 root root 0 Jul 1 03:46 /proc/4/attr/exec
```

**Anomalie1 :** Tout fichier ou répertoire système crucial (/etc, /bin, /sbin, etc.) qui est world-writable est une anomalie. Les répertoires comme /tmp et /var/tmp sont censés être world-writable, mais devraient avoir le sticky bit (voir ci-dessous).

**Anomalie 2 : Des binaires non standard ou des scripts utilisateur avec des permissions SUID/SGID. Par exemple, si un script personnel a SUID, c'est un risque.**

**find / -type f \(-perm -4000 -o -perm -2000 \|) -exec ls -ld {} \|; 2>/dev/null** : utilisée pour identifier les fichiers ayant les bits SUID (Set User ID) ou SGID (Set Group ID) activés sur votre système Linux.

Action corrective : sudo chmod 600 /chemin/vers/le\_fichier ; sudo chmod 644 /chemin/vers/le\_fichier

```
x vagrant@docker ~ % sudo find / -type f \(-perm -4000 -o -perm -2000 \|) -exec ls -ld {} \|; 2>/dev/null
-r-xr-sr-x. 1 root root 15344 Jun  9  2014 /usr/bin/wall
-rws--x--x. 1 root root 23968 Feb  2  2021 /usr/bin/chfn
-rwsr--x--x. 1 root root 23880 Feb  2  2021 /usr/bin/chsh
-rw-r--r-x. 1 root root 73888 Aug  9  2019 /usr/bin/chage
-rwsr--r-x. 1 root root 78408 Aug  9  2019 /usr/bin/gpasswd
-rw-r--r-x. 1 root root 41936 Aug  9  2019 /usr/bin/newgrp
-rw-r--r-x. 1 root root 44264 Feb  2  2021 /usr/bin/mount
-rw-r--r-x. 1 root root 32128 Feb  2  2021 /usr/bin/su
---s--x--x. 1 root root 151424 Jan 25  2023 /usr/bin/sudo
-rw-r--r-x. 1 root root 31984 Feb  2  2021 /usr/bin/umount
-rwxr--r-x. 1 root tty 19544 Feb  2  2021 /usr/bin/write
-rw-r--r-x. 1 root root 27672 Jan 25  2022 /usr/bin/pkexec
-rw-r--r-x. 1 root root 57576 May 16  2023 /usr/bin/crontab
---x--s--x. 1 root nobody 382208 Aug  4  2023 /usr/bin/ssh-agent
-rw-r--r-x. 1 root root 27856 Apr  1  2020 /usr/bin/passwd
-rw-r--r-x. 1 root root 36272 Apr  1  2020 /usr/sbin/unix_chkpwd
-rw-r--r-x. 1 root root 11232 Apr  1  2020 /usr/sbin/pam_timestamp_check
-rw-r--r-x. 1 root root 11224 Nov 16  2020 /usr/sbin/netrenport
```

**Anomalie 3 : Des fichiers orphelins peuvent indiquer des suppressions incorrectes d'utilisateurs/groupes ou des problèmes système : find / -nouser -o -nogroup -exec ls -ld {} \|; 2>/dev/null** : Cette commande puissante est conçue pour localiser tous les fichiers et répertoires de votre système Linux qui n'ont pas de propriétaire (utilisateur) ou de groupe valide. Ces fichiers sont souvent appelés fichiers « orphelins ».

```
x vagrant@docker ~ % sudo find / -nouser -o -nogroup -exec ls -ld {} \|; 2>/dev/null
x vagrant@docker ~ %
```

**ls -ld /tmp /var/tmp** : utilisée pour afficher des informations détaillées sur les répertoires et eux-mêmes, plutôt que sur leur contenu. ls -ld /tmp /var/tmp

```
x vagrant@docker ~ % ls -ld /tmp /var/tmp
rwxrwxrwt. 13 root root 4096 Jul  1 03:33 /tmp
rwxrwxrwt.  8 root root 4096 Jul  1 02:14 /var/tmp
```

**Vérifier le "sticky bit" sur les répertoires publics** : Les répertoires comme /tmp et /var/tmp doivent être world-writable mais aussi avoir le sticky bit (+t) pour empêcher les utilisateurs de supprimer les fichiers des autres.

**Anomalie 4 : Toute entrée autre que root est une anomalie et une faille de sécurité majeure.**

**awk -F: '\$3 == "0"' { print }' /etc/passwd**

```
vagrant@docker ~ % awk -F: '$3 == "0"' { print }' /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

```
[vagrant@centos8 ~]$ awk -F: '$3 == "0"' { print }' /etc/passwd
root:x:0:0:root:/root:/bin/bash
```

**Anomalie 5 : Extrêmement rare et dangereux. Normalement, le deuxième champ doit contenir un "x" (indiquant un hash dans /etc/shadow) ou un hash chiffré.**

**awk -F: '\$2 == ""' { print }' /etc/passwd**

```
[vagrant@centos8 ~]$ awk -F: '$2 == ""' { print }' /etc/passwd
```

**awk -F: '\$7 != "/bin/bash" && \$7 != "/bin/sh" && \$7 != "/sbin/nologin" && \$7 != "/bin/false"' { print }' /etc/passwd**

**grep "Failed password" /var/log/secure | awk '{print \$11, \$9, \$1}' | sort | uniq -c | sort -nr | head**

**grep -E "PermitRootLogin|PasswordAuthentication|UsePAM|AllowUsers|AllowGroups" /etc/ssh/sshd\_config**

**Anomalie : Des ports ouverts qui ne devraient pas l'être, ou des règles de pare-feu trop permissives.**

**ss -tulnpx**

```
vagrant@docker: ~ [1] awk -F: '($3 == "0") { print }' /etc/passwd
root:x:0:0:root:/bin/bash
vagrant@docker: ~ [1] sudo ss -tulnpx
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port
u_str LISTEN 0 10 /var/lib/gssproxy/default.sock 21602
    users:(("gssproxy",pid=682,fd=8))
u_str LISTEN 0 128 /run/docker.sock 20034
    users:(("dockerd",pid=1281,fd=4),("systemd",pid=1,fd=40))
u_dgr UNCONN 0 0 /run/systemd/shutdownd 12612
    users:(("systemd",pid=1,fd=32))
u_str LISTEN 0 128 /run/dbus/system_bus_socket 20036
    users:(("dbus-daemon",pid=672,fd=3),("systemd",pid=1,fd=41))
u_str LISTEN 0 100 private/defer 27063
    users:(("master",pid=1145,fd=38))
u_str LISTEN 0 128 /run/containerd/containerd.sock.ttrpc 27214
    users:(("containerd",pid=1038,fd=8))
u_str LISTEN 0 128 /run/containerd/containerd.sock 27219
    users:(("containerd",pid=1038,fd=9))
u_str LISTEN 0 100 private/trace 27066
    users:(("master",pid=1145,fd=41))
u_str LISTEN 0 100 private/verify 27069
    users:(("master",pid=1145,fd=41))

firewall-cmd --list-all --zone=public ps aux | grep '^root'
```

**Anomalie 10 :** Messages d'erreur répétitifs, tentatives de connexion échouées, accès à des fichiers non autorisés, activités d'utilisateurs inconnus, modifications inattendues de configurations.

```
journalctl -r      # Les journaux du système (du plus récent au plus ancien)
```

```
vagrant@docker: ~ [1] sudo journalctl -r
-- Logs begin at Tue 2025-07-01 02:14:02 UTC, end at Tue 2025-07-01 04:48:20 UTC. --
Jul 01 01 04:48:20 docker sudo[9613]: pam_unix(sudo:session): session opened for user root by vagrant(uid=0)
Jul 01 01 04:48:20 docker sudo[9613]:  vagrant : TTY=pts/0 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/jour
Jul 01 04:46:15 docker nm-dispatcher[9494]: req:1 'dhcp4-change' [enp0s8]: start running 'ordered scripts...
Jul 01 04:46:15 docker systemd[1]: Started Network Manager Script Dispatcher Service.
Jul 01 04:46:15 docker nm-dispatcher[9494]: req:1 'dhcp4-change' [enp0s8]: new request (3 scripts)
Jul 01 04:46:15 docker dbus[672]: [system] Successfully activated service 'org.freedesktop.nm_dispatcher'
Jul 01 04:46:15 docker systemd[1]: Starting Network Manager Script Dispatcher Service...
Jul 01 04:46:15 docker dhclient[778]: bound to 192.168.56.4 -- renewal in 279 seconds.
Jul 01 04:46:15 docker dbus[672]: [system] Activating via systemd: service name='org.freedesktop.nm_dispatcher'
Jul 01 04:46:15 docker NetworkManager[689]: <info>  [1751345175.4422] dhcp4 (enp0s8): state changed bound
Jul 01 04:46:15 docker NetworkManager[689]: <info>  [1751345175.4421] dhcp4 (enp0s8): lease time 600
Jul 01 04:46:15 docker NetworkManager[689]: <info>  [1751345175.4421] dhcp4 (enp0s8): plen 24 (255.255.25
Jul 01 04:46:15 docker NetworkManager[689]: <info>  [1751345175.4408] dhcp4 (enp0s8): address 192.168.56
Jul 01 04:46:15 docker NetworkManager[689]: <info>  [1751345175.4408] dhcp4 (enp0s8): gw 192.168.56.2 (via 192.168.56.1)


```

```
journalctl -r -u sshd  # Journaux spécifiques au service SSH
```

```
vagrant@docker: ~ [1] sudo journalctl -r -u sshd
-- Logs begin at Tue 2025-07-01 02:14:02 UTC, end at Tue 2025-07-01 04:51:12 UTC. --
Jul 01 02:26:06 docker sshd[1650]: Accepted publickey for vagrant from 10.0.2.2 port 65374 ssh
Jul 01 02:14:31 docker systemd[1]: Started OpenSSH server daemon.
Jul 01 02:14:31 docker sshd[1022]: Server listening on :: port 22.
Jul 01 02:14:31 docker sshd[1022]: Server listening on 0.0.0.0 port 22.
Jul 01 02:14:30 docker systemd[1]: Starting OpenSSH server daemon...
```

**Anomalie :** Services écoutant sur des ports inattendus, services non autorisés ou non nécessaires. Chaque port ouvert est une surface d'attaque potentielle.

```
cat /var/log/secure  # Authentifications et activités liées à la sécurité
cat /var/log/messages  # Messages système généraux
cat /var/log/audit/audit.log # Si auditd est configuré et actif
```

### 3. Audit des Connexions et Authentifications

Derniers logins des utilisateurs :`last`

Anomalie : Connexions d'utilisateurs inconnus, connexions à des heures inhabituelles, ou depuis des adresses IP suspectes.

**Échecs de connexion SSH :**

```
grep "Failed password" /var/log/secure | awk '{print $11, $9, $1}' | sort | uniq -c | sort -nr | head
```

Anomalie : Nombre élevé de tentatives de connexion échouées depuis une IP donnée (tentative de brute-force).

Configuration SSH (`sshd_config`) :

u_dgr	UNCONN	0	0	* 22148	*
11553				* 22485	*
u_dgr	UNCONN	0	0	* 20982	*
11553				* 18911	*
u_dgr	UNCONN	0	0	* 21499	*
11561				* 20898	*
u_dgr	UNCONN	0	0	* 25751	*
11553				* 25756	*
u_dgr	UNCONN	0	0	* 19971	*
11561				* 24138	*
u_dgr	UNCONN	0	0	* 28002	*
11553				* 11548	*
u_dgr	UNCONN	0	0	* 21977	*
28081				* 28786	*
u_dgr	UNCONN	0	0	127.0.0.1:323	0.0.0.0:
11539				[::1]:323	[::1]:
u_dgr	UNCONN	0	0	0.0.0.22	0.0.0.0:
0					
u_dgr	UNCONN	0	0		
11553					
udp	UNCONN	0	0		
*					
udp	UNCONN	0	0		
*					
tcp	LISTEN	0	128		
*					

## 5. Audit des Processus en Cours

Processus par l'utilisateur root :

ps aux | grep '^root'

**Anomalie :** Des processus inattendus ou non essentiels s'exécutant en tant que root. Un attaquant pourrait essayer d'exécuter son code avec les priviléges les plus élevés.

Processus en écoute sur des ports ouverts (vu avec ss -tulnpx):

Assurez-vous que les processus quiouvrent les ports sont légitimes.

## 6. Audit des Logiciels et Mises à Jour

Paquets obsolètes/mises à jour disponibles :

sudo yum check-update

**Anomalie/Faillie potentielle :** Des paquets obsolètes peuvent contenir des vulnérabilités connues qui n'ont pas été corrigées. Il est crucial de maintenir le système à jour.

Liste des paquets installés (pour identifier des logiciels inconnus) :

**Liste des paquets installés (pour identifier des logiciels inconnus) :**

yum list installed

**Anomalie :** Paquets inconnus ou non autorisés installés sur le système.

## 7. Vérification des Journaux (Logs)

Les journaux sont une mine d'informations sur les anomalies.

Accès aux journaux :journalctl -r        # Les journaux du système (du plus récent au plus ancien)

Jan 24 10:01:51	centos8	localdomain	sud0[72903]	vagrant : TTY:tty1 : PWD->/home/vagrant : USER=root	
Jan 24 10:01:51	centos8	localdomain	run-parts[72801]	(etc/cron.hourly) finished Bacula	
Jan 24 10:01:51	centos8	localdomain	run-parts[72802]	(etc/cron.hourly) starting Bacula	
Jan 24 10:01:51	centos8	localdomain	run-parts[72803]	(root) /etc/cron.hourly/bacula-backup	
Jan 24 10:00:56	centos8	localdomain	systemd[1]	: allocate-updated.service: Succeeded.	
Jan 24 10:00:56	centos8	localdomain	systemd[1]	: Started system activity accounting tool.	
Jan 24 10:00:56	centos8	localdomain	systemd[1]	: Started update-database for allocate.	
Jan 24 10:00:56	centos8	localdomain	systemd[1]	: Started update-database for allocate.	
Jan 24 09:59:23	centos8	localdomain	sud0[72511]	pan_unix(sudo:session): session closed for user root	
Jan 24 09:59:23	centos8	localdomain	sud0[72511]	pan_unix(sudo:session): session opened for user root	
Jan 24 09:58:22	centos8	localdomain	sud0[72621]	man.aptitude(sudo:navigation). Cache creation: /	
Jan 24 09:58:22	centos8	localdomain	sud0[72621]	vagrant : TTY:tty1 : PWD->/home/vagrant : USER=root	
Jan 24 09:55:59	centos8	localdomain	sud0[72521]	Started dm-cache.service: Successed.	
Jan 24 09:55:59	centos8	localdomain	dm[72521]	: Metadata time caching disabled when running on a b	
Jan 24 09:55:57	centos8	localdomain	sud0[72511]	: Starting dm_wkcache...	
Jan 24 09:55:57	centos8	localdomain	systemd[1]	: Starting system activity accounting tool.	
Jan 24 09:58:56	centos8	localdomain	systemd[1]	: systat-collect.service: Succeeded.	
Jan 24 09:58:56	centos8	localdomain	systemd[1]	: Started system activity accounting tool...	
Jan 24 09:48:47	centos8	localdomain	systemd[1]	: Starting system activity accounting tool...	
Jan 24 09:38:56	centos8	localdomain	systemd[1]	: Started system activity accounting tool...	
Jan 24 09:38:56	centos8	localdomain	systemd[1]	: Starting system activity accounting tool...	
Jan 24 09:28:56	centos8	localdomain	systemd[1]	: Started system activity accounting tool...	
Jan 24 09:28:56	centos8	localdomain	systemd[1]	: Started system activity accounting tool...	
Jan 24 09:18:38	centos8	localdomain	chron0[8271]	Selected source 162.159.288.1 (2.centos.pool.ntp.o	
Jan 24 09:16:19	centos8	localdomain	chron0[8271]	Selected source 99.73.39.214 (2.centos.pool.ntp.o	
Jan 24 09:13:06	centos8	localdomain	chron0[8271]	Forward time jump detected!	

journalctl -r -u sshd    # Journaux spécifiques au service SSH

cat /var/log/secure    # Authentifications et activités liées à la sécurité

cat /var/log/messages    # Messages système généraux

cat /var/log/audit/audit.log # Si auditd est configuré et actif

**Anomalie :** Messages d'erreur répétitifs, tentatives de connexion échouées, accès à des fichiers non autorisés, activités d'utilisateurs inconnus, modifications inattendues de configurations.

```

Jun 24 00:29:29 centos8 sudo[4355]: pam_unix(sudo:session): session closed for user root
Jun 24 01:49:53 centos8 sudo[4411]: vagrant : TTY=tty1 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl list-units--files --type=service
Jun 24 01:49:53 centos8 sudo[4411]: pam_systemd(sudo:session): Cannot create session: Already running in a session or user slice
Jun 24 01:49:53 centos8 sudo[4411]: pam_unix(sudo:session): session opened for user root by vagrant(id=0)
Jun 24 01:49:53 centos8 sudo[4411]: pam_unix(sudo:session): session closed for user root
Jun 24 02:03:22 centos8 sudo[4456]: vagrant : TTY=tty1 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl list-units--files --type=service --state=enabled --no-pager
Jun 24 02:03:22 centos8 sudo[4456]: pam_systemd(sudo:session): Cannot create session: Already running in a session or user slice
Jun 24 02:03:22 centos8 sudo[4456]: pam_unix(sudo:session): session opened for user root by vagrant(id=0)
Jun 24 02:03:22 centos8 sudo[4456]: pam_unix(sudo:session): session closed for user root
Jun 24 02:08:43 centos8 sudo[4460]: vagrant : TTY=tty1 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl list-units--files --type=service --state=enabled --no-pager
Jun 24 02:08:43 centos8 sudo[4460]: pam_systemd(sudo:session): Cannot create session: Already running in a session or user slice
Jun 24 02:08:43 centos8 sudo[4460]: pam_unix(sudo:session): session opened for user root by vagrant(id=0)
Jun 24 02:08:43 centos8 sudo[4460]: pam_unix(sudo:session): session closed for user root
Jun 24 02:09:12 centos8 sudo[4465]: vagrant : TTY=tty1 ; PWD=/home/vagrant ; USER=root ; COMMAND=/bin/systemctl list-units--files --type=service --state=enabled --no-pager
Jun 24 02:09:12 centos8 sudo[4465]: pam_systemd(sudo:session): Cannot create session: Already running in a session or user slice
Jun 24 02:09:12 centos8 sudo[4465]: pam_unix(sudo:session): session opened for user root by vagrant(id=0)
Jun 24 02:09:12 centos8 sudo[4465]: pam_unix(sudo:session): session closed for user root

```

### Outils d'Audit de Sécurité Plus Avancés :

Pour un audit plus approfondi et automatisé, des outils dédiés sont recommandés :

**Lynis** : Un outil d'audit de sécurité et de hardening open-source qui effectue un scan complet du système et propose des recommandations.

**OpenSCAP** : Pour la conformité aux standards de sécurité (comme PCI DSS, DISA STIG).

**Rootkit Hunter (rkhunter)** : Recherche les rootkits, les backdoors et les exploits locaux connus.

### Corrections et durcissement

#### a) Désactiver les services non essentiels

Lister tous les services actifs et leurs états

La première étape consiste à voir ce qui tourne actuellement sur votre système : **sudo systemctl list-units --type=service --all**

UNIT	LOAD	ACTION	SUB	DESCRIPTION
audited.service	loaded	active	running	Security Auditing Service
chronyd.service	loaded	active	running	MTP client/server
cpupower.service	loaded	inactive	dead	Configure CPU power related settings
crond.service	loaded	active	running	Command Scheduler
dbus.service	loaded	active	running	D-Bus System Message Bus
display-manager.service	not-found	inactive	dead	display-manager.service
dm-event.service	loaded	inactive	dead	Device-mapper event daemon
dnf-makecache.service	loaded	inactive	dead	dnf makecache
dracut-cmdline.service	loaded	inactive	dead	dracut cmdline hook
dracut-initqueue.service	loaded	inactive	dead	dracut initqueue hook
dracut-mount.service	loaded	inactive	dead	dracut mount hook
dracut-pre-mount.service	loaded	inactive	dead	dracut pre-mount hook
dracut-pre-pivot.service	loaded	inactive	dead	dracut pre-pivot and cleanup hook
dracut-pre-trigger.service	loaded	inactive	dead	dracut pre-trigger hook
dracut-pre-unbind.service	loaded	active	exited	Restore /run/intramfs on shutdown
dracut-ramdom.service	loaded	inactive	dead	Ethernet Bridging Filtering tables
ethtool.service	loaded	inactive	dead	Emergency Shell
firewalld.service	loaded	active	running	firewalld - dynamic firewall daemon
getty@tty1.service	loaded	active	running	Getty on tty1
haveged.service	loaded	active	running	Entropy Daemon based on the HAVEGE algorithm
import-state.service	loaded	active	exited	Import network configuration from /etc/sysconfig/network
initrd-cleanups.service	loaded	inactive	dead	Cleaning Up and Shutting Down DRAM
initrd-parse-etc.service	loaded	inactive	dead	Reload Configuration from the RAM
initrd-switch-root.service	loaded	inactive	dead	Switch Root
initrd-udevadm-cleanup-db.service	loaded	inactive	dead	Cleanup udevd DB
ip6tables.service	not-found	inactive	dead	ip6tables.service
ipset.service	not-found	inactive	dead	ipset.service
iptables.service	not-found	inactive	dead	iptables.service
irqbalance.service	loaded	active	running	irqbalance daemon
kmmod-static-nodes.service	loaded	active	exited	Create list of required static kernel modules
ldconfig.service	loaded	inactive	dead	Rebuild Dynamic Linker Cache
loadmodules.service	loaded	inactive	dead	Load legacy module configuration
lvm2-activation.service	not-found	inactive	dead	lvm2-activation.service
lvm2-lvmpid.service	loaded	inactive	dead	LVM poll daemon

**Cmd2 : sudo systemctl list-unit-files --type=service --state=enabled** (Liste des services qui actives automatiquement au démarrage du système)

INIT FILE	STATE
audited.service	enabled
autovt@.service	enabled
chronyd.service	enabled
crond.service	enabled
ibus-org.fedoraproject.FirewallD1.service	enabled
ibus-org.freedesktop.nm-dispatcher.service	enabled
ibus-org.freedesktop.timedate1.service	enabled
firewalld.service	enabled
getty@.service	enabled
haveged.service	enabled
import-state.service	enabled
irqbalance.service	enabled
loadmodules.service	enabled
lvm2-monitor.service	enabled
networkkManager-dispatcher.service	enabled
networkkManager-wait-online.service	enabled
networkkManager.service	enabled
nis-domainname.service	enabled
rsyslog.service	enabled
selinux-autorelabel-mark.service	enabled
sshd.service	enabled
sssd.service	enabled
syslog.service	enabled
sysstat.service	enabled
timedate.service	enabled
tuned.service	enabled
toolbox-service.service	enabled
toolbox.service	enabled

28 unit files listed.

Exemple de services souvent considérés comme "non essentiels" dans un environnement minimal :

- **Serveurs Web/Base de données (si non utilisés) : httpd, nginx, mariadb, postgresql**
- **Services de messagerie : postfix, sendmail (si le serveur n'envoie ou ne reçoit pas d'e-mails directement)**
- **Serveurs de fichiers : nfs-server, samba (si le serveur ne partage pas de fichiers)**
- **Services graphiques ou de bureau : gdm, lightdm, Xorg (si c'est un serveur sans interface graphique)**

- **Services d'impression** : cups (si le serveur n'est pas une machine d'impression)
- **Services de jeux ou multimédia** : Tout service lié à ces fonctions.
- **Services de développement ou de test** : Souvent, des outils de développement restent actifs après le déploiement.

#### b) Corriger les permissions trop larges

a) Correction des Fichiers et Répertoires "World-Writable" (Accessibles en Écriture par Tous)

*Si un fichier ou un répertoire est modifiable par "tout le monde" (le bit d'écriture est activé pour les "autres"), c'est un risque majeur.*

- 1.1 Identifier les fichiers World-Writable : `find / -type f -perm -o+w -exec ls -ld {} \; 2>/dev/null`  
Corriger les permissions d'un fichier World-Writable :

```
sudo chmod o-w /chemin/vers/fichier
# Ou, pour des permissions plus strictes (lecture/écriture pour le propriétaire seulement) :
sudo chmod 644 /chemin/vers/fichier # Propriétaire: rw- | Groupe: r-- | Autres: r--
# ou
sudo chmod 600 /chemin/vers/fichier # Propriétaire: rw- | Groupe: --- | Autres: ---
```

- 1.2 Corriger les permissions d'un répertoire World-Writable (sans sticky bit) :

```
find / -type d -perm -o+w ! -perm -o+t -exec ls -ld {} \; 2>/dev/null
```

- 1.3 Correction des Fichiers SUID/SGID Inappropriés

```
find / -type f \! -perm -4000 -o -perm -2000 \! -exec ls -ld {} \; 2>/dev/null
```

```
sudo chmod u-s /chemin/vers/binaire # Retire le bit SUID
```

```
sudo chmod g-s /chemin/vers/binaire # Retire le bit SGID
```

# Ou retire les deux :

```
sudo chmod -s /chemin/vers/binaire
```

- 1.4 Corriger les fichiers non possédés :

```
sudo chown root:root /chemin/vers/fichier_non_posse
```

#### 2) Sécuriser les fichiers de configuration (SSH, sudoers, etc.)

##### a) Correction des Permissions Inappropriées sur les Fichiers SSH

Vérifier et corriger les clés privées SSH (`id_rsa`, `id_ed25519`, etc.) : `chmod 600 ~/ssh/id_rsa` # Faites-le pour toutes vos clés privées

Vérifier et corriger le répertoire `.ssh` : `chmod 700 ~/ssh`

Vérifier et corriger le fichier `authorized_keys` : `chmod 600 ~/ssh/authorized_keys`

Vérifier et corriger `/etc/ssh/sshd_config` :

```
sudo chown root:root /etc/ssh/sshd_config
```

```
sudo chmod 644 /etc/ssh/sshd_config
```

#### b) Correction des Permissions sur `/etc/sudoers` et `/etc/sudoers.d/`

a)Vérifier et corriger `/etc/sudoers` :

```
sudo chown root:root /etc/sudoers
```

```
sudo chmod 440 /etc/sudoers
```

b)Vérifier et corriger les fichiers dans `/etc/sudoers.d/` :

```
sudo chown root:root /etc/sudoers.d/nom_du_fichier
```

```
sudo chmod 440 /etc/sudoers.d/nom_du_fichier
```

#### c) Vérifier et renforcer la configuration SSH (port, root login, authentification par clé)

##### d) Sauvegarder la configuration actuelle

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak_$(date +%Y%m%d%H%M)
```

Ouvrir le fichier de config :

```
sudo vi /etc/ssh/sshd_config
```

# Ou si vous préférez nano :

```
# sudo nano /etc/ssh/sshd_config
```

Vérifier et Modifier les Paramètres Clés :

Port SSH

Connexion Root Directe (PermitRootLogin) : `PermitRootLogin prohibit-password` : `PermitRootLogin no`

Authentification par Mot de Passe (PasswordAuthentication) : `PasswordAuthentication no`

Limiter les Utilisateurs/Groupes Autorisés (AllowUsers, AllowGroups) : `AllowGroups sshusers`

Désactiver l'authentification basée sur les hôtes (HostbasedAuthentication) : `HostbasedAuthentication no`

## Déploiement application web

Étape 1 : Installer Nginx Installez le dépôt EPEL :

```
sudo yum install epel-release -y
```

Installez Nginx :

```
sudo yum install nginx -y
```

*Démarrez et activez Nginx au démarrage :*

*sudo systemctl start nginx*

*sudo systemctl enable nginx*

```
vagrant@docker: ~ $ sudo systemctl start nginx
vagrant@docker: ~ $ sudo systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.
vagrant@docker: ~ $ sudo systemctl status nginx
nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2025-06-29 22:35:55 UTC; 24s ago
     Main PID: 6707 (nginx)
        CGroup: /system.slice/nginx.service
                  ├─6707 nginx: master process /usr/sbin/nginx
                  ├─6708 nginx: worker process
                  ├─6709 nginx: worker process

Jun 29 22:35:55 docker systemd[1]: Starting The nginx HTTP and reverse proxy server...
Jun 29 22:35:55 docker nginx[6701]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jun 29 22:35:55 docker nginx[6701]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jun 29 22:35:55 docker systemd[1]: Started The nginx HTTP and reverse proxy server.
vagrant@docker: ~ $
```

## *Étape 2 :*

#### *Configurer le Pare-feu (Firewalld).*

*Autorisez les ports HTTP (80) et HTTPS (443) :*

```
sudo firewall-cmd --permanent --add-service=http  
sudo firewall-cmd --permanent --add-service=https  
sudo firewall-cmd --permanent --add-service=ssh
```

```
sudo firewall-cmd --reload
```

```
sudo su -c 'cloud-init --once'
X vagrant@docker ~ ~ ~ sudo firewall-cmd --permanent --add-service=http
FirewallD is not running
X vagrant@docker ~ ~ ~ sudo systemctl start firewalld
vagrant@docker ~ ~ ~ sudo firewall-cmd --permanent --add-service=http
success
vagrant@docker ~ ~ ~ sudo firewall-cmd --permanent --add-service=https
success
vagrant@docker ~ ~ ~ sudo firewall-cmd --permanent --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
vagrant@docker ~ ~ ~ sudo firewall-cmd --reload
success
```

### **Étape 3 : Configurer Nginx comme Proxy Inverse pour l'Application sur le Port 8080**

*Nous allons créer un fichier de configuration Nginx qui écoutera sur le port 80 et redirigera les requêtes vers votre application tournant sur le port 8080.*

Il est préférable de créer un fichier de configuration séparé dans `nginx/conf.d/` plutôt que de modifier `nginx.conf` directement.  
`sudo vi /etc/nginx/conf.d/myapps.conf`

Collez-v le contenu suivant :

```
server {  
    listen 80;  
    docker.localdomain : # Remplacez par votre nom de domaine ou IP du serveur
```

```
location / {
    proxy_pass http://127.0.0.1:8080; # Redirige vers votre application locale sur le port 8080
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

```

proxy_set_header X-Forwarded-Proto $scheme;
}

# Ajoutez cette section si votre application utilise des WebSockets
location /ws {
    proxy_pass http://127.0.0.1:8080;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
}

error_page 500 502 503 504 /50x.html;
location =/50x.html {
    root /usr/share/nginx/html;
}
}

```

Ensuite saisir les commandes :

**Sudo nginx -t**

**Sudo systemctl reload nginx**

**Étape 4 : Vérifier le Bon Fonctionnement**

Curl <http://127.0.0.1:80>

```

properly.</p>
<div> <class="alert">
<h2>Website Administrator</h2>
<div> <class="content">
<p>This is the default <tt>index.html</tt> page that
is generated by nginx on
Red Hat Enterprise Linux. It is located in
<tt>/usr/share/nginx/html</tt>.</p>
<p>You should now put your content in a location of
your choice and edit the <tt>root</tt> configuration
directive in the <strong>nginx</strong>
configuration file
<tt>/etc/nginx/nginx.conf</tt>.</p>
<p>For information on Red Hat Enterprise Linux, please visit the <a href="http://www.redhat.com">Red Hat, Inc. website</a>. The documentation for Red Hat Enterprise Linux is <a href="http://www.redhat.com/docs/manuals/enterprise">available on the Red Hat, Inc. website</a>.</p>
</div>
</div>
<div> <class="logos">
<a href="http://nginx.net"></a>
<a href="http://www.redhat.com"></a>
</div>
</div>
</body>
</html>
```

Vous devriez voir votre application web s'afficher.