Bayssié Loïc
Bedat Joritz

Sécurité des systèmes d'exploitation
Guide de durcissement
Ubuntu

TOULOUSE
ynov
CAMPUS

Bayssié Loïc
Bedat Joritz

Table des matières

Hardening durant l'installation

Utiliser des mots de passe forts

Dans les premières étapes de l'installation, un utilisateur sera créé et ajouté au groupe d'administration, il est donc impératif de choisir un mot de passe fort pour ce compte.

Utiliser le chiffrement de disque

Il est primordial d'activer le chiffrement de disque pendant l'installation, celui-ci protège vos données principalement en cas de vol de disque dur. Dans un premier temps l'option à choisir est: *"use entire disk and set up encrypted LVM"*. Dans un second temps il vous sera demandé de créer une passphrase qui sera demandée à chaque boot du PC et permettra d'accéder aux données déchiffrées.

Mises à jour de sécurité automatiques

Pendant l'installation il sera question de la gestion des mises à jour du système, il faudra alors choisir l'option *"Install security updates automatically"* qui permet de mettre à jour automatiquement le système ainsi que les outils utilisés.

Installation minimale

Il est question ici de n'installer que le strict nécessaire sur le système qui sera utilisé. L'installation de packages et/ou d'outils peut parfois induire le démarrage de certains services sans que l'utilisateur en soit nécessairement conscient. Dans le cas d'outils peu utilisés, des services peuvent alors tourner dans une configuration vulnérable. Installer le strict minimum permet de réduire la surface d'attaque.

Accès, authentification, autorisations

Mises à jour automatiques

Durant l'installation il était possible de configurer les mises à jour de sécurité automatique, si cela n'a pas été fait il est possible d'entrer la commande suivante :

```
apt install unattended-upgrades
```

Ou pour les versions plus anciennes :

```
apt-get update && apt-get upgrade
```

Et pour Ubuntu 16.04 LTS ou plus récentes :

```
apt update && apt upgrade
```

Tâches planifiées basées sur le temps

Fonctionnement

Vérifier que crone est activé :
```
# systemctl is-enabled cron
```

Vérifier que cron est en fonctionnement :
```
# systemctl status cron | grep 'Active: active (running) '
```

Pour l'activer et le lancer si ce n'est pas le cas vous pouvez lancer :
```
# systemctl --now enable cron
```

Permissions de /etc/crontab

Vérifier les permissions sur crontab :
```
# stat /etc/crontab
```

Modifier les permissions sur crontab :
```
# chown root:root /etc/crontab
# chmod og-rwx /etc/crontab
```

Permissions /etc/cron.hourly

Vérifier les permissions :

```
# stat /etc/cron.hourly/
```

Modifier les permissions :

```
# chown root:root /etc/cron.hourly/
# chmod og-rwx /etc/cron.hourly/
```

Permissions /etc/cron.daily

Vérifier les permissions :

```
# stat /etc/cron.daily/
```

Modifier les permissions :

```
# chown root:root /etc/cron.daily/
# chmod og-rwx /etc/cron.daily/
```

Permissions /etc/cron.weekly

Vérifier les permissions :

```
# stat /etc/cron.weekly/
```

Modifier les permissions :

```
# chown root:root /etc/cron.weekly/
# chmod og-rwx /etc/cron.weekly/
```

Permissions /etc/cron.monthly

Vérifier les permissions :

```
# stat /etc/cron.monthly/
```

Modifier les permissions :

```
# chown root:root /etc/cron.monthly/
# chmod og-rwx /etc/cron.monthly/
```

Permissions /etc/cron.d

Vérifier les permissions :

```
# stat /etc/cron.d/
```

Modifier les permissions :

```
# chown root:root /etc/cron.d/
# chmod og-rwx /etc/cron.d/
```

Crone users

Vérifier l'existence de /etc/cron.deny :

```
# stat /etc/cron.deny
stat: cannot stat `/etc/cron.deny': No such file or directory
```

On vérifie les droits sur /etc/cron.allow

```
# stat /etc/cron.allow
Access: (0640/-rw-r-----) Uid: ( 0/ root) Gid: ( 0/ root)
```

Supprimer /etc/cron.deny

```
# rm /etc/cron.deny
```

Créer /etc/cron.allow

```
# touch /etc/cron.allow
```

Définir les droits

```
# chmod g-wx,o-rwx /etc/cron.allow
# chown root:root /etc/cron.allow
```

Commande /etc/at

Vérifier que /etc/at.deny n'existe pas

```
# stat /etc/at.deny
stat: cannot stat `/etc/at.deny': No such file or directory
```

Vérifier les droits sur /etc/at.allow

```
# stat /etc/at.allow
Access: (0640/-rw-r-----) Uid: ( 0/ root) Gid: ( 0/ root)
```

Supprimer /etc/at.deny

```
# rm /etc/at.deny
```

Créer /etc/at.allow

```
# touch /etc/at.allow
```

Définir les permissions

```
# chmod g-wx,o-rwx /etc/at.allow
# chown root:root /etc/at.allow
```

PAM

Installation et configuration de PAM : pwquality

Pwquality permet d'assurer une certaine qualité des mots de passe utilisés

Installation :
```
apt install libpam-pwquality
```

Il est possible d'effectuer des modification dans le fichier :

```
/etc/security/pwquality.conf
```

- Longueur :
  - `minlen = 14` - Le mot de passe doit faire 14 caractères de long minimum.
- Complexité:
  - `minclass = 4` - Le nombre minimum de caractères différents (majuscule, minuscule, chiffre, caractère spécial)
  - `dcredit = -1` - Nécessite au moins un chiffre
  - `ucredit = -1` - Nécessite au moins une majuscule
  - `ocredit = -1` - Nécessite au moins un caractère spécial
  - `lcredit = -1` - Nécessite au moins une minuscule

- La suite se trouve dans le fichier suivant : `/etc/pam.d/common-password`

  `retry=3` Autorise trois tentatives de connexion avant de renvoyer une erreur.

Comptes & Environnement

Expiration du mot de passe

Lancer cette commande pour vérifier la politique d'expiration du mot de passe
```
# grep PASS_MAX_DAYS /etc/login.defs
PASS_MAX_DAYS 365
```

Lancer la commande suivante pour voir l'expiration des mots de passe de chaque utilisateur
```
# grep -E '^[^:]+:[^!*]' /etc/shadow | cut -d: -f1,5
```

Paramétrer l'expiration des mots de passe à 365 jours conformément à la politique demandée
```
PASS_MAX_DAYS 365
```

Configurer cette date d'expiration pour les utilisateurs listés
```
# chage --maxdays 365 <user>
```

Avertissement expiration du mot de passe

Lancer cette commande pour vérifier la politique d'avertissement d'expiration du mot de passe

```
# grep PASS_WARN_AGE /etc/login.defs
PASS_WARN_AGE 7
```

Vérifier l'avertissement d'expiration des mots de passe de chaque utilisateur

```
# grep -E ^[^:]+:[^\!*] /etc/shadow | cut -d: -f1,6
```

Paramétrer l'avertissement d'expiration des mots de passe à 7 jours conformément à la politique demandée

```
PASS_WARN_AGE 7
```

Configurer la date d'avertissement d'expiration pour les utilisateurs listés

```
# chage --warndays 7 <user>
INACTIVE=30
```

Inactivité du mot de passe changé

Vérifier la période d'inactivité d'un mot de passe changé

```
# useradd -D | grep INACTIVE
```

Vérifier la période d'inactivité d'un mot de passe par utilisateur

```
# grep -E ^[^:]+:[^\!*] /etc/shadow | cut -d: -f1,7
```

Taper cette commande pour définir la durée par défaut d'inactivité d'un mot de passe

```
# useradd -D -f 30
```

Taper cette commande pour définir la durée par utilisateur d'inactivité d'un mot de passe

```
# chage --inactive 30 <user>
```

Root restreint au mode terminal

On vérifie les entrées de chaque terminal

```
# cat /etc/securetty
```

Dans ce même fichier, il faut tous les postes qui ne sont pas présents physiquement sur le réseau.

Restreindre l'accès à la commande su

Créer un groupe vide spécifique à la commande su

```
# groupadd sugroup
```

Ajouter la ligne suivante au fichier `/etc/pam.d/su` en spécifiant le groupe vide

```
auth required pam_wheel.so use_uid group=sugroup
```

Maintenance Système

Permissions sur /etc/passwd

Accès voulu :

```
# stat /etc/passwd

Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

Commandes de configuration de permissions voulues :

```
# chown root:root /etc/passwd
# chmod 644 /etc/passwd
```

Permissions sur /etc/gshadow-

Accès voulu :

```
# stat /etc/gshadow-

Access: (0640/-rw-r-----)  Uid: (    0/    root)   Gid: (    0/    root)
```

Commandes de configuration de permissions voulues :

```
# chown root:root /etc/gshadow-
# chown root:shadow /etc/gshadow-

# chmod o-rwx,g-wx /etc/gshadow-
```

Permissions sur /etc/shadow

Accès voulu :

```
# stat /etc/shadow
Access: (0640/-rw-r-----)  Uid: (    0/    root)   Gid: (    0/    root)
```

Commandes de configuration de permissions voulues :

```
# chown root:root /etc/shadow
# chown root:shadow /etc/shadow

# chmod o-rwx,g-wx /etc/shadow
```

Permissions sur /etc/group

Accès voulu :

```
# stat /etc/group

Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
```

Commandes de configuration de permissions voulues :

```
# chown root:root /etc/group

# chmod u-x,go-wx /etc/group
```

Vérifier l'existence de fichier librement éditables

Faites la commande suivante et vérifiez qu'elle ne retourne aucun fichier :

```
# df --local -P | awk '{if (NR!=1) print $6}' | xargs -I '{}' find '{}' -xdev
-type f -perm -0002
```

Vérifier qu'aucun utilisateur ait de mot de passe vide

Exécutez la commande suivante et vérifiez qu'elle ne retourne aucun résultat:

```
# awk -F: '($2 == "" ) { print $1 " does not have a password "}' /etc/shadow
```

Vérifier l'intégrité du chemin de root

Exécutez la commande suivante et vérifiez qu'il n'y ait aucun retour :

```
#!/bin/bash

if echo $PATH | grep -q "::" ; then
   echo "Empty Directory in PATH (::)"
fi
if echo $PATH | grep -q ":$" ; then
        echo "Trailing : in PATH"
fi
for x in $(echo $PATH | tr ":" " ") ; do
  if [ -d "$x" ] ; then
    ls -ldH "$x" | awk '
$9 == "." {print "PATH contains current working directory (.)"}
$3 != "root" {print $9, "is not owned by root"}
substr($1,6,1) != "-" {print $9, "is group writable"}
substr($1,9,1) != "-" {print $9, "is world writable"}'
  else
    echo "$x is not a directory"
  fi
done
```

Vérifier que tous les utilisateurs aient un dossier 'home'

Exécutez la commande suivante et vérifiez qu'il n'y ait aucun retour :

```
#!/bin/bash
grep -E -v '^(halt|sync|shutdown)' /etc/passwd | awk -F: '($7 != "'"$(which
nologin)"'" && $7 != "/bin/false") { print $1 " " $6 }' | while read -r user
dir; do
        if [ ! -d "$dir" ]; then
                echo "The home directory ($dir) of user $user does not exist."
        fi
done
```

Vérifiez que les home aient des permissions à 750 ou plus restrictives

Exécutez la commande suivante et vérifiez qu'il n'y ait aucun retour :

```
#!/bin/bash
grep -E -v '^(halt|sync|shutdown)' /etc/passwd | awk -F: '($7 != "'"$(which
nologin)"'" && $7 != "/bin/false") { print $1 " " $6 }' | while read user
dir; do
  if [ ! -d "$dir" ]; then
    echo "The home directory ($dir) of user $user does not exist."
  else
    dirperm=$(ls -ld $dir | cut -f1 -d" ")
    if [ $(echo $dirperm | cut -c6) != "-" ]; then
      echo "Group Write permission set on the home directory ($dir) of user
$user"
    fi
    if [ $(echo $dirperm | cut -c8) != "-" ]; then
      echo "Other Read permission set on the home directory ($dir) of user
$user"
    fi
    if [ $(echo $dirperm | cut -c9) != "-" ]; then
      echo "Other Write permission set on the home directory ($dir) of user
$user"
    fi
    if [ $(echo $dirperm | cut -c10) != "-" ]; then
      echo "Other Execute permission set on the home directory ($dir) of user
$user"
    fi
  fi
done
```

S'assurer qu'aucun utilisateur n'ait de fichiers .netrc

Exécutez la commande suivante et vérifiez qu'il n'y ait aucun retour :

```
#!/bin/bash

grep -E -v '^(root|halt|sync|shutdown)' /etc/passwd | awk -F: '($7 !=
"'"$(which nologin)"'" && $7 != "/bin/false") { print $1 " " $6 }' | while
read user dir; do
  if [ ! -d "$dir" ]; then
    echo "The home directory ($dir) of user $user does not exist."
  else
    if [ ! -h "$dir/.netrc" -a -f "$dir/.netrc" ]; then
      echo ".netrc file $dir/.netrc exists"
    fi
  fi
done
```

S’assurer que les fichiers .netrc ne soient pas accessibles par des groupes ou librement

Exécutez la commande suivante et vérifiez qu’il n’y ait aucun retour :

```
#!/bin/bash

grep -E -v '^(root|halt|sync|shutdown)' /etc/passwd | awk -F: '($7 !=
"'"$(which nologin)"'" && $7 != "/bin/false") { print $1 " " $6 }' | while
read user dir; do
  if [ ! -d "$dir" ]; then
    echo "The home directory ($dir) of user $user does not exist."
  else
    for file in $dir/.netrc; do
      if [ ! -h "$file" -a -f "$file" ]; then
        fileperm=$(ls -ld $file | cut -f1 -d" ")
        if [ $(echo $fileperm | cut -c5)  != "-" ]; then
          echo "Group Read set on $file"
        fi
        if [ $(echo $fileperm | cut -c6)  != "-" ]; then
          echo "Group Write set on $file"
        fi
        if [ $(echo $fileperm | cut -c7)  != "-" ]; then
          echo "Group Execute set on $file"
        fi
        if [ $(echo $fileperm | cut -c8)  != "-" ]; then
          echo "Other Read set on $file"
        fi
        if [ $(echo $fileperm | cut -c9)  != "-" ]; then
          echo "Other Write set on $file"
        fi
        if [ $(echo $fileperm | cut -c10)  != "-" ]; then
          echo "Other Execute set on $file"
        fi
      fi
    done
  fi
done
```

S’assurer qu’aucun utilisateur n’ait de fichiers .rhosts

Exécutez la commande suivante et vérifiez qu’il n’y ait aucun retour :

```
#!/bin/bash

grep -E -v '^(root|halt|sync|shutdown)' /etc/passwd | awk -F: '($7 !=
"'"$(which nologin)"'" && $7 != "/bin/false") { print $1 " " $6 }' | while
read user dir; do
  if [ ! -d "$dir" ]; then
    echo "The home directory ($dir) of user $user does not exist."
  else
    for file in $dir/.rhosts; do
      if [ ! -h "$file" -a -f "$file" ]; then
        echo ".rhosts file in $dir"
      fi
    done
  fi
done
```

S’assurer que tous les groupes existant dans /etc/passwd existent dans /etc/group

Exécutez la commande suivante et vérifiez qu’il n’y ait aucun retour :

```
#!/bin/bash

for i in $(cut -s -d: -f4 /etc/passwd | sort -u ); do
  grep -q -P "^.*?:[^:]*:$i:" /etc/group
  if [ $? -ne 0 ]; then
    echo "Group $i is referenced by /etc/passwd but does not exist in
/etc/group"
  fi
done
```

S’assurer qu’il n’y ait aucun doublons dans les UIDs ou GIDs

Exécutez les commandes suivantes et vérifiez qu’il n’y ait aucun retour :

UIDs :

```
#!/bin/bash

cut -f3 -d":" /etc/passwd | sort -n | uniq -c | while read x ; do
  [ -z "$x" ] && break
  set - $x
  if [ $1 -gt 1 ]; then
    users=$(awk -F: '($3 == n) { print $1 }' n=$2 /etc/passwd | xargs)
    echo "Duplicate UID ($2): $users"
  fi
done
```

GIDs :

```
#!/bin/bash

cut -d: -f3 /etc/group | sort | uniq -d | while read x ; do
    echo "Duplicate GID ($x) in /etc/group"
done
```

S’assurer qu’il n’y ait pas de doublons dans les noms d’utilisateurs ou de groupes

Exécutez les commandes suivantes et vérifiez qu’il n’y ait aucun retour :

Users :

```
#!/bin/bash

cut -d: -f1 /etc/passwd | sort | uniq -d | while read x
do echo "Duplicate login name ${x} in /etc/passwd"
done
```

Groups:

```
#!/bin/bash

cut -d: -f1 /etc/group | sort | uniq -d | while read x
do echo "Duplicate group name ${x} in /etc/group"
done
```

S’assurer que le groupe shadow est vide

Exécutez les commandes suivantes et vérifiez qu’il n’y ait aucun retour :

```
# grep ^shadow:[^:]*:[^:]*:[^:]+ /etc/group
# awk -F: '($4 == "<shadow-gid>") { print }' /etc/passwd
```