

## TP2 - Bonnes pratiques de la Sécurité du Système d'Exploitation

---

### 1. Indiquer la raison derrière chacun des points de durcissement suivants

#### a. Utiliser des dépôts signés :

**Intégrité** : En utilisant des dépôts signés, on s'assure de l'intégrité des logiciels téléchargés. Les signatures numériques garantissent que le logiciel n'a pas été altéré, assurant ainsi l'exactitude des biens logiciels.

#### b. Enlever les systèmes de fichiers non utilisés :

**Disponibilité** : En retirant les systèmes de fichiers non utilisés, on réduit la surface d'attaque(vulnérabilités), minimisant ainsi les risques de compromission et contribuant à la disponibilité des systèmes. **lsmod**

**Intégrité** : La suppression des systèmes de fichiers inutiles contribue à maintenir l'intégrité du système en éliminant les points d'entrée potentiels pour les attaquants.

Exemple :

Si le module "cdrom" est détecté comme actif sur l'ordinateur, mais qu'aucun lecteur de CD n'est physiquement présent sur la machine, il serait préférable de le désactiver ou de le supprimer.

#### c. S'assurer qu'il n'y a pas d'UID dupliqué :

**Intégrité** : En évitant les UID dupliqués, on garantit l'exactitude et la complétude des identifiants d'utilisateur, contribuant ainsi à l'intégrité des autorisations du système.

En effet, si deux utilisateurs partagent le même UID (identifiant d'utilisateur), cela peut entraîner un problème de traçabilité et de sécurité. L'UID est utilisé pour identifier de manière unique chaque utilisateur dans un système, et s'il y a des duplications, cela peut causer des confusions et des risques potentiels. Voici comment cela pourrait se manifester :

Supposons que deux utilisateurs, A et B, partagent le même UID :

1. L'utilisateur A crée un fichier appelé "document.txt".
2. Puisque les deux utilisateurs ont le même UID, l'utilisateur B peut également accéder et modifier le fichier "document.txt" créé par l'utilisateur A.

Cela crée un problème de traçabilité, car il devient difficile de déterminer quel utilisateur a effectivement créé ou modifié un fichier donné, ce qui compromet la capacité à suivre les actions spécifiques de chaque utilisateur.

Pour éviter ce problème, il est essentiel de garantir que chaque utilisateur ait un UID unique. Cela peut être géré en attribuant des UID différents lors de la création des comptes utilisateurs, en évitant les duplications qui pourraient entraîner des conflits de sécurité et des problèmes de traçabilité.

**d. Limiter l'utilisation et l'accès aux fichiers de vidage mémoire (core dump) :**

**Confidentialité :** Limiter l'accès aux fichiers de vidage mémoire préserve la confidentialité en évitant la divulgation involontaire d'informations sensibles sur l'état du système lors d'un crash.

**e. S'assurer que la variable PATH de l'utilisateur root ne contient pas le répertoire de travail courant (.) ou tout autre répertoire inscriptible par tous :**

**Intégrité :** En évitant l'inclusion du répertoire de travail courant dans la variable PATH, on réduit le risque d'exécution de commandes malveillantes, préservant ainsi l'intégrité du système.

**Confidentialité :** Cela contribue également à la confidentialité en empêchant l'inclusion de répertoires inscriptibles par tous dans la variable PATH, réduisant les risques liés à l'injection de commandes malveillantes.

Permettre à l'attaquant d'exécuter n'importe quelle commande est rendu possible par des vulnérabilités, comme l'exemple où la commande `cp /bin/ps ./ls` peut être utilisée pour substituer la commande `ls` par `ps`. Lorsque `ls` est ensuite exécuté, en réalité, la commande `ps` est lancée à la place. Cette manipulation expose le système à des risques de sécurité, car l'attaquant peut introduire du code malveillant contenu dans `ps` pour exploiter des vulnérabilités.

**2. Donnez un exemple concret du principe de minimisation, du moindre privilège, de défense en profondeur et d'activité de veille et maintenance.**

- **Minimisation :**

Exemple : Désactivez un service inutile.

Commande : `sudo systemctl disable nom_du_service`

On peut de même lister tous les modules du noyau avec la commande `lsmod`, ensuite on identifie les modules non utilisés (regarder la colonne "Used By", et les modules qui ont "0" dans cette colonne sont potentiellement non utilisés), ensuite on désactive le module temporairement pour tester avec la commande suivante : `sudo modprobe -r example_module`

- Commande : **chmod permissions fichier**

- Commande : **sudo ufw enable**

- Commande : `sudo apt-get update && sudo apt-get upgrade`

```

rahmonex@cyclop-os: ~
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name                               Version
+++-+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
ii accountsservice                     0.6.55-0ubuntu12~20.04.4
ii acl                                  2.2.53-6
ii acpi-support                         0.143
ii acpid                               1:2.0.32-1ubuntu1
ii adduser                             3.118ubuntu2
ii adwaita-icon-theme                  3.36.1-2ubuntu0.20.04.2
ii aisleriot                           1:3.22.9-1
ii alacarte                            3.36.0-1+cyclop1
ii alsa-base                           1.0.25+dfsg-0ubuntu5
ii alsa-topology-conf                  1.2.2-1
ii alsa-ucm-conf                       1.2.2-1ubuntu0.11
ii alsa-utils                          1.2.2-1ubuntu2.1
ii amd64-microcode                    3.20191218.1ubuntu1
ii anacron                             2.3-29
ii apg                                  2.2.3.dfsg.1-5
ii app-install-data-partner            19.04
ii apparmor                             2.13.3-7ubuntu5.1
ii apport                              2.20.11-0ubuntu27.21
ii apport-gtk                          2.20.11-0ubuntu27.21
ii apport-symptoms                     0.23
ii appstream                           0.12.10-2
ii appstream-data-cyclop-os            1.0.0
ii appstream-data-cyclop-os-icons      1.0.0
ii appstream-data-cyclop-os-icons-hidpi 1.0.0
ii apt                                  2.0.6
ii apt-config-icons                    0.12.10-2
ii apt-config-icons-hidpi              0.12.10-2
ii apt-utils                           2.0.6
ii aptdaemon                           1.1.1+bzr982-0ubuntu32.3
ii aptdaemon-data                      1.1.1+bzr982-0ubuntu32.3

```

Après avoir examiné la liste des packages installés sur mon système, j'ai constaté que certains ne sont pas nécessaires. Pour optimiser la configuration, je prévois de faire un tri en supprimant les packages inutiles à l'aide de la commande : `sudo apt-get remove nom_du_package`.

#### 4. Vérifier si /tmp est configuré dans sa propre partition ou avec son propre système de fichier (tmpfs). Dans le cas contraire, qu'est-ce qu'il faut faire pour remédier à ce problème. Quel est le principe de sécurité appliqué dans ce cas?


Pour vérifier si /tmp est configuré dans sa propre partition ou avec son propre système de fichier tmpfs, on peut utiliser la commande `df -h /tmp`

```
rahmonex@cyclop-os:~$ df -h /tmp
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda5       141G   15G  119G  12% /
```

/tmp est monté sur une partition dédiée, car on peut voir les détails de cette partition. Si /tmp était configuré avec tmpfs (signifie que /tmp utilise un espace en mémoire vive) on aurait pas eu des détails sur la partition.

Si /tmp n'est pas configuré avec tmpfs et que l'on souhaite remédier à ce problème pour améliorer la sécurité, on peut monter /tmp dans un système de fichiers en mémoire (tmpfs). L'idée est d'isoler temporairement les fichiers temporaires en mémoire vive, ce qui peut être plus sécurisé que de les stocker sur un disque. Et pour se faire, on réalise ces étapes :

- Modifier le fichier /etc/fstab avec la commande `sudo gedit /etc/fstab`
- Ajouter une ligne pour /tmp avec tmpfs avec la commande `tmpfs /tmp tmpfs defaults,noatime,mode=1777 0 0` et on sauvegarde le fichier.



```
Open  ▾  [Icon]  fstab /etc  Save  ⋮  -  □  ×
1 # /etc/fstab: static file system information.
2 #
3 # Use 'blkid' to print the universally unique identifier for a
4 # device; this may be used with UUID= as a more robust way to name devices
5 # that works even if disks are added and removed. See fstab(5).
6 #
7 # <file system> <mount point> <type> <options> <dump> <pass>
8 # / was on /dev/sda5 during installation
9 tmpfs /tmp tmpfs defaults,noatime,mode=1777 0 0
10 UUID=b13af9b1-4d06-47c7-94fa-ca5cf87fe3a3 / ext4 errors=remount-ro 0 1
11 # /boot/efi was on /dev/sda1 during installation
12 UUID=026A-F608 /boot/efi vfat umask=0077 0 1
13 /swapfile none swap sw 0 0
```

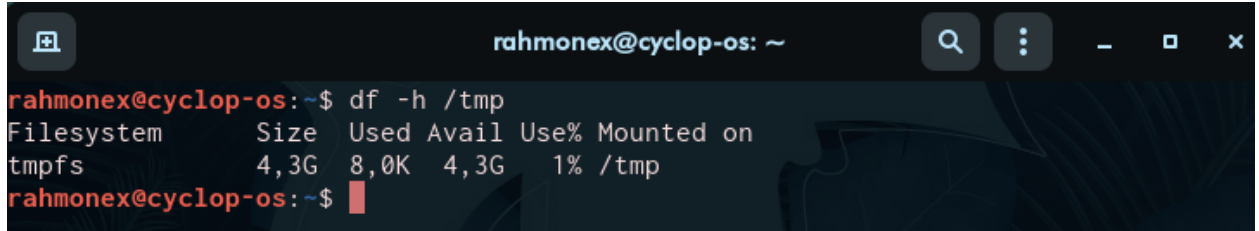
tmpfs : Type de système de fichiers en mémoire vive.

/tmp : Point de montage du système de fichiers.

defaults,noatime,mode=1777 : Options de montage par défaut avec désactivation des mises à jour du temps d'accès et définition des permissions.

0 0 : Options pour le système de fichiers, indique qu'il ne sera pas sauvegardé et l'ordre de vérification n'a pas d'importance.

- Remonter /tmp ou redémarrez le système avec la commande `sudo mount -o remount /tmp`



```
rahmonex@cyclop-os: ~  
rahmonex@cyclop-os:~$ df -h /tmp  
Filesystem      Size  Used Avail Use% Mounted on  
tmpfs           4,3G   8,0K  4,3G   1% /tmp  
rahmonex@cyclop-os:~$
```

On remarque qu' après redémarrage cela a bien été pris en compte.

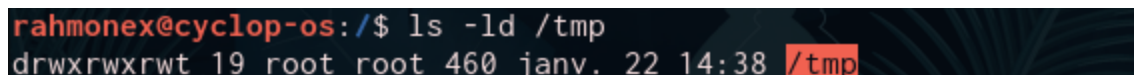
En effet, les fichiers temporaires stockés sur le disque, tels que dans un système de fichiers classique, peuvent parfois contenir des informations sensibles ou des traces d'activité. En utilisant tmpfs, la fenêtre d'opportunité pour qu'un attaquant exploite ces fichiers temporaires est réduite, car ils sont effacés à chaque redémarrage du système.

Ainsi, cette configuration contribue également à la sécurité en limitant la persistance des données temporaires, ce qui est bénéfique pour réduire les risques liés aux traces d'activité potentiellement exploitées par des attaquants.

## 5. Vérifier que le sticky bit est activé sur tous les répertoires inscriptibles par tous. Quel est le principe de sécurité appliqué dans ce cas

Pour vérifier que le sticky bit est activé sur tous les répertoires inscriptibles par tous, on peut utiliser la commande find. Le sticky bit est généralement activé sur les répertoires avec l'autorisation "rwxrwxrwt". On peut utiliser la commande `find / -type d -perm -002 -exec ls -ld {} \;` qui recherche tous les répertoires (type "d") avec des autorisations inscriptibles par tous. Si le sticky bit est activé, on aperçoit un "t" à la fin des permissions du répertoire.

On peut de même vérifier un répertoire spécifique avec la commande `ls -ld [nom du répertoire]` et vérifier si on obtient "drwxrwxrwt" :



```
rahmonex@cyclop-os:/$ ls -ld /tmp  
drwxrwxrwt 19 root root 460 janv. 22 14:38 /tmp
```

On remarque que /tmp a bien le sticky bit actif.

```
rahmonex@cyclop-os:~$ find / -type d -perm -002 -exec ls -ld {} \;  
find: '/lost+found': Permission denied  
find: '/root': Permission denied  
find: '/etc/cups/ssl': Permission denied  
find: '/etc/ssl/private': Permission denied  
find: '/etc/polkit-1/localauthority': Permission denied  
find: '/snap/core18/2812/etc/ssl/private': Permission denied  
find: '/snap/core18/2812/root': Permission denied  
drwxrwxrwt 2 root root 3 oct. 27 19:00 /snap/core18/2812/tmp  
find: '/snap/core18/2812/var/cache/ldconfig': Permission denied  
find: '/snap/core18/2812/var/lib/private': Permission denied  
drwxrwxrwt 2 root root 3 oct. 27 06:57 /snap/core18/2812/var/tmp  
find: '/snap/core18/2796/etc/ssl/private': Permission denied  
find: '/snap/core18/2796/root': Permission denied  
drwxrwxrwt 2 root root 3 sept. 1 18:52 /snap/core18/2796/tmp  
find: '/snap/core18/2796/var/cache/ldconfig': Permission denied  
find: '/snap/core18/2796/var/lib/private': Permission denied  
drwxrwxrwt 2 root root 3 sept. 1 06:54 /snap/core18/2796/var/tmp  
find: '/snap/core22/1033/etc/ssl/private': Permission denied  
find: '/snap/core22/1033/root': Permission denied  
drwxrwxrwt 3 root root 29 nov. 23 09:13 /snap/core22/1033/run/lock  
drwxrwxrwt 2 root root 3 nov. 23 09:12 /snap/core22/1033/tmp  
find: '/snap/core22/1033/var/cache/ldconfig': Permission denied  
find: '/snap/core22/1033/var/cache/private': Permission denied
```

En ce qui concerne le principe de sécurité appliqué dans ce cas, le sticky bit est généralement utilisé sur les répertoires où plusieurs utilisateurs ont le droit d'écrire. Lorsqu'il est activé, seuls les propriétaires des fichiers peuvent les supprimer, même si le répertoire est inscriptible par d'autres utilisateurs.

Par exemple, dans le répertoire /tmp (qui est souvent configuré avec le sticky bit), tous les utilisateurs peuvent écrire des fichiers, mais ils ne peuvent supprimer que leurs propres fichiers. Cela permet de prévenir la suppression accidentelle de fichiers par d'autres utilisateurs et contribue à maintenir la sécurité et la confidentialité des données dans ces répertoires partagés.

#### 6. Écrire une commande bash qui permet de chercher tous les *binaires suid* dans votre système. Est-ce dangereux de les avoir ?

On peut de même utiliser la commande find pour rechercher tous les binaires suid (Set User ID) sur notre système, on peut par exemple utiliser cette commande : `find / -type f -perm -4000 2>/dev/null`. Cette commande recherche tous les fichiers (-type f) avec le bit suid activé (-perm -4000) à partir de la racine du système (/). La redirection 2>/dev/null supprime les erreurs liées aux répertoires auxquels on n'a pas accès.

En ce qui concerne la dangerosité des binaires suid, cela dépend du contexte. Les binaires suid sont des programmes qui s'exécutent avec les droits d'utilisateur du propriétaire plutôt qu'avec les droits



de l'utilisateur qui les lance. Cela peut être potentiellement dangereux s'ils sont mal configurés ou s'ils présentent des vulnérabilités.

Il est important de comprendre la nature de chaque binaire suid et de s'assurer qu'ils sont nécessaires pour les fonctionnalités du système. Si un binaire suid n'est pas requis, il est recommandé de le désactiver ou de le supprimer pour réduire la surface d'attaque potentielle. Une gestion appropriée des binaires suid contribue à renforcer la sécurité du système.

## 7. Expliquer comment durcir quelques paramètres du noyau (sysctl -a).

Le commandement `sysctl -a` permet d'afficher tous les paramètres du noyau sur un système Linux. Pour durcir (renforcer la sécurité) quelques paramètres du noyau, on peut ajuster certains des paramètres les plus critiques. Quelques exemples :

- Désactiver le renvoi de paquets ICMP (ping) :

On peut utiliser la commande `sudo sysctl -w net.ipv4.icmp_echo_ignore_all=1`, cela désactive la réponse aux requêtes ICMP, ce qui peut aider à éviter certaines formes d'attaque de déni de service.

```
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.icmp_echo_ignore_all=1
[sudo] password for rahmonex:
net.ipv4.icmp_echo_ignore_all = 1
rahmonex@cyclop-os:/$
```

- Désactiver le transfert de paquets IP redirigés :

On peut utiliser les commandes

```
sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
```

```
sudo sysctl -w net.ipv4.conf.default.accept_redirects=0
```

Cela peut aider à prévenir les attaques de redirection malicieuse de paquets.

```
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.all.accept_redirects = 0
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.conf.default.accept_redirects=0
net.ipv4.conf.default.accept_redirects = 0
rahmonex@cyclop-os:/$
```

- Désactiver le transfert de paquets IP redirigés :

Les commandes suivantes :

```
sudo sysctl -w net.ipv4.conf.all.accept_source_route=0
```

```
sudo sysctl -w net.ipv4.conf.default.accept_source_route=0
```

désactive la possibilité pour un paquet IP de spécifier son propre itinéraire.

```
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.conf.all.accept_source_route=0
net.ipv4.conf.all.accept_source_route = 0
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.conf.default.accept_source_route=0
net.ipv4.conf.default.accept_source_route = 0
rahmonex@cyclop-os:/$
```

- Désactiver le transfert de paquets IP avec la possibilité d'IP Spoofing :

On peut utiliser ces commandes qui aide à prévenir les attaques basées sur l'usurpation d'adresse IP :

```
sudo sysctl -w net.ipv4.conf.all.rp_filter=1
```

```
sudo sysctl -w net.ipv4.conf.default.rp_filter=1
```

```
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.all.rp_filter = 1
rahmonex@cyclop-os:/$ sudo sysctl -w net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.default.rp_filter = 1
rahmonex@cyclop-os:/$
```

## 8. Vous voulez renforcer la politique de mots de passe sur votre serveur Linux.

### a. Comment imposer une durée de vie aux mots de passe pour les utilisateurs ?

Pour imposer une durée de vie aux mots de passe des utilisateurs sur un serveur Linux, on peut utiliser la commande **chage**.

Par exemple, on peut définir une durée de vie de maximum 90 jours pour le mot de passe pour l'utilisateur avec la commande **sudo chage -M 90 nom\_utilisateur**.

On vérifie d'abord les paramètres actuels du mot de passe à l'utilisateur avec la commande **sudo chage -l nom\_utilisateur**

```
rahmonex@cyclop-os:/$ sudo chage -l rahmonex
Last password change                : nov. 03, 2023
Password expires                     : never
Password inactive                    : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
rahmonex@cyclop-os:/$ sudo chage -M 90 rahmonex
rahmonex@cyclop-os:/$ sudo chage -l rahmonex
Last password change                : nov. 03, 2023
Password expires                     : févr. 01, 2024
Password inactive                    : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 90
Number of days of warning before password expires : 7
rahmonex@cyclop-os:/$
```

### b. Imposer l'utilisation des mots de passe complexes avec le module PAM, en s'appuyant sur les recommandations ANSSI.

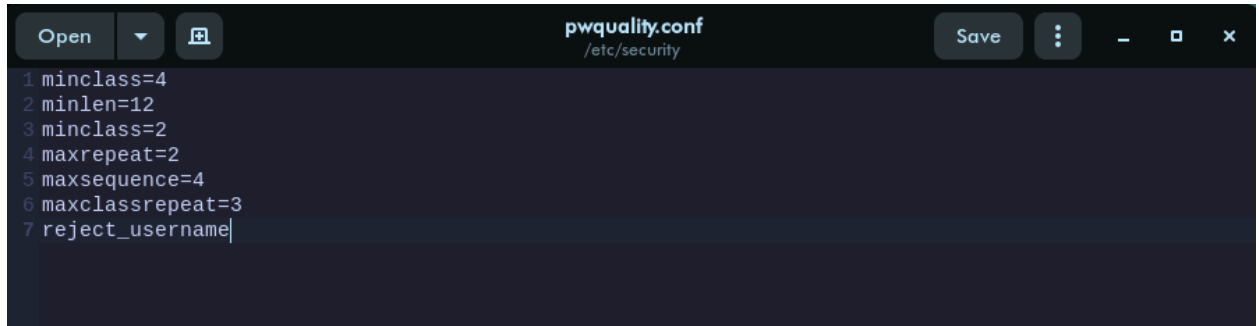
Pour imposer l'utilisation de mots de passe complexes en s'appuyant sur les recommandations de l'ANSSI (Agence nationale de la sécurité des systèmes d'information), on peut configurer le module PAM (Pluggable Authentication Modules) sur notre serveur Linux en suivant ces étapes :



- Éditer le fichier de configuration PAM pour les mots de passe (/etc/security/pwquality.conf)

```
rahmonex@cyclop-os:~$ sudo gedit /etc/security/pwquality.conf
```

- Configurer les paramètres de complexité du mot de passe selon les recommandations de l'ANSSI

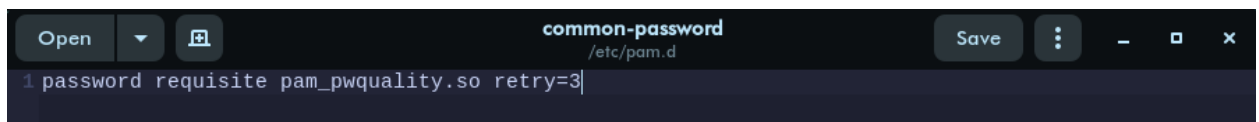


```
1 minclass=4
2 minlen=12
3 minclass=2
4 maxrepeat=2
5 maxsequence=4
6 maxclassrepeat=3
7 reject_username|
```

Ces paramètres spécifient des exigences telles que la longueur minimale du mot de passe (minlen), le nombre minimal de classes de caractères différentes (minclass), le nombre maximal de caractères répétés (maxrepeat), etc.

- Configurer /etc/pam.d/common-password

```
rahmonex@cyclop-os:~$ sudo gedit /etc/pam.d/common-password
```



```
1 password requisite pam_pwquality.so retry=3|
```

Cette ligne spécifie que le module pam\_pwquality.so est requis et qu'il doit être réessayé jusqu'à trois fois en cas d'échec.

Après avoir apporté ces modifications, le système devrait appliquer ces règles lors de la création ou de la modification des mots de passe des utilisateurs.