

## Thème du TP

### 1. Préambule

L'objectif de TP est d'explorer les mécanismes de contrôles d'accès, notamment DAC et MAC, implémentés dans les environnements Linux et Windows.

### 2. Prérequis

Trois machines virtuelles au format VirtualBox ont été préparées pour ce TP :

- SES\_TP1\_Win10 : reprise du TP1 (pour mémoire, le mot de passe du compte IEUser est *Passw0rd*) ! ;
- SSE\_TP2\_Debian : utilisation d'une distribution Debian Linux 9 installée dans une partition chiffrée (mot de passe *tp*) et dotée d'un compte utilisateur nommé *tp* (mot de passe *tp*) ayant l'autorisation de lancer des commandes à l'aide de *sudo* ;
- SSE\_TP2\_Centos : utilisation d'une distribution CentOS Linux 7 dotée d'un compte utilisateur nommé *tp* (mot de passe *tp*) ayant l'autorisation de lancer des commandes à l'aide de *sudo*.

### 3. Contrôle d'accès en environnement Linux

#### **Exercice 3-1**

Objectif : manipuler les permissions UNIX et les ACL Posix

La VM à utiliser pour cet exercice est la VM Debian.

La commande *umask* permet de modifier les permissions par défaut des fichiers créés par un utilisateur. Modifier le fichier *.bashrc* de l'utilisateur *tp* pour que les fichiers qu'il crée en ligne de commande soit en lecture/écriture pour lui seul.

\_\_\_\_\_

\_\_\_\_\_

Ce mécanisme fonctionne en tout ou rien, il est possible d'obtenir un résultat plus fin, répertoire par répertoire, avec la commande *setfacl*. Créer un répertoire */tmp/privatebox* dans lequel tous les utilisateurs peuvent créer des fichiers privés par défaut (i.e. qui sont visibles et modifiables uniquement par le propriétaire).

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Les fichiers créés peuvent-ils être supprimés par d'autres utilisateurs (en dehors de root) ?

---

---

## **Exercice 3-2**

Objectif : Rechercher les utilitaires SUID/SGID root

Sur la VM Debian, taper la commande suivante

```
find / -type f \( -perm -2000 -a -gid 0 -o -perm -4000 -a -uid 0 \) -print 2>/dev/null
```

En principe, la commande ping fait partie de la liste. Taper la commande suivante pour vérifier son bon fonctionnement :

```
ping 127.0.0.1
```

Taper les commandes suivantes en testant à chaque fois l'impact de la commande taper sur le comportement de ping :

```
sudo chmod u-s /bin/ping  
sudo setcap cap_net_raw=p /bin/ping
```

---

---

Les opérations réalisés sur les droits du fichier /bin/ping présentent-elles un intérêt ? Si oui, lequel ? Si besoin, la description des capacités Linux s'obtient avec `man capabilities`.

---

---

Pour restaurer les droits et capacités d'origine, il faut taper les commandes suivantes :

```
sudo setcap -r /bin/ping  
sudo chmod u+s /bin/ping
```

Refaites les mêmes tests sur la machine virtuelle CentOS et comparer les approches.

### **Exercice 3-3**

Objectif : identifier les processus utilisant des capacités

La VM à utiliser pour cet exercice est la VM Debian.

Consulter la page de manuel de la commande `getpcaps` (située dans `/sbin/getpcaps`) et écrire un script permettant d'identifier les processus privilégiés. Dans un premier temps, ce script doit se limiter à remonter tous les processus utilisant au moins une capacité. Dans un deuxième temps, ce script doit permettre de cibler une ou plusieurs capacités données.

### **Exercice 3-4**

Objectif : évaluer les conséquences de certaines options de montage d'un système de fichiers

La VM à utiliser pour cet exercice est la VM Debian.

Les commandes suivantes servent à créer un fichier qui va héberger un système de fichiers.

```
sudo dd if=/dev/zero of=/var/fichier.part bs=512 count=100k
sudo losetup /dev/loop0 /var/fichier.part
sudo mkfs -t ext4 /dev/loop0
sudo mount -t ext4 /dev/loop0 /mnt
```

Lancer la commande `mount` sans argument pour vérifier les options de montage utilisées par défaut pour `/mnt`.

Les commandes suivantes servent à simuler la préparation d'un attaquant :

```
sudo cp /usr/bin/sudo /mnt
sudo chmod 4755 /mnt/sudo
```

Nota : il s'agit ici de la commande `sudo` ; standard pour des questions de simplicité, il faut imaginer une version modifiée qui ne fasse pas de contrôles basés sur le fichier `sudoers`.

```
/mnt/sudo id
sudo cp /bin/nc /mnt
sudo setcap cap_net_bind_service=ep /mnt/nc
sudo cp /bin/sh /mnt/su_sh
sudo chmod 4755 /mnt/su_sh
```

Ecrire un script `/mnt/evil-script.sh` qui utilise l'interpréteur `/mnt/su_sh` pour copier le fichier `/etc/shadow` sur `/mnt` ou afficher son contenu. Puis exécuter le.

## UR1 M2 - SSE - Sécurité des systèmes d'exploitation

Tester la mise en écoute du port 80 avec netcat :

```
nc -l 80
```

puis sa version doté d'une capacité :

```
/mnt/nc -l 80
```

Tester si le port est bien en écoute, par exemple avec la commande suivante :

```
wget http://127.0.0.1
```

Changer les options de montage en retirant les permissions suid

```
sudo mount -t ext4 /dev/loop0 /mnt -o remount,nosuid
```

Est-ce que ce changement est directement visible sur le système de fichiers ?

---

---

Quel est l'impact sur les commandes suivantes :

```
/mnt/sudo id
```

```
/mnt/evil-script.sh
```

```
/mnt/nc -l 80
```

---

---

Changer les options de montage en retirant les permissions suid et exécution

```
sudo mount -t ext4 /dev/loop0 /mnt -o remount,noexec,nosuid
```

Est-ce que ce changement est directement visible sur le système de fichiers ?

---

---

Quel est l'impact sur les commandes suivantes :

```
/mnt/sudo id
```

```
/mnt/evil-script.sh
```

```
/mnt/nc -l 80
```

---

---

Démonter ce système de fichiers

```
sudo umount /mnt
```

```
sudo losetup -d /dev/loop0
```

### **Exercice 3-4**

Objectif : appréhender les commandes courantes SELINUX

La VM à utiliser pour cet exercice est la VM CentOS.

Vérifier l'état de SELinux

```
sestatus
```

Consulter les contextes de sécurité des processus courants et des fichiers du répertoire courant

```
ps -eZ
```

```
ls -Z
```

Consulter le contexte de sécurité de l'utilisateur courant et vérifier la persistance de l'identité SELinux en cas de changement d'identité UNIX.

```
id
```

```
sudo id
```

Consulter les règles d'attributions des identités SELinux.

```
sudo semanage login -l
```

```
sudo semanage user -l
```

Tester le changement de ces règles :

```
sudo adduser util1
```

```
sudo passwd util1
```

```
sudo semanage login -a -s util1 user_u
```

Se connecter sur un autre terminal (CTRL+ALT+F2), puis taper

```
id
```

Revenir sur la session du compte tp (CTRL+ALT+F1)

```
sudo -u util1 id
```

Y a-t-il une différence ?

---

---

La page Debian est un aide-mémoire assez pratique :

<https://debian-handbook.info/browse/en-US/stable/sect.selinux.html#id-1.17.8.5.8>

## Exercice 3-5

Objectif : comprendre des blocages induits par la politique SELINUX

La VM à utiliser pour cet exercice est la VM CentOS.

Un wiki (Dokuwiki) a été installé sans préoccupation de la politique SE Linux (ni de préoccupation de sécurité tout court), il est accessible à l'adresse : <http://localhost/dokuwiki>. Il ne fonctionne pas correctement quand SELinux est activé en mode *enforcing*. Tester que le wiki fonctionne correctement en mode *permissive* :

```
sudo setenforce 0
```

Puis réactiver le mode *enforcing*

```
sudo setenforce 1
```

Vérifier s'il existe un module ciblant dokuwiki dans ceux fournis par défaut :

```
sudo semanage module -l
```

```
Sudo semodule -l
```

En principe, ce n'est pas le cas. Par conséquent, il va donc falloir procéder autrement.

```
Sudo tail -n 100 /var/log/audit/audit.log | grep denied
```

Identifier les contextes source et destination et vérifier qu'ils sont cohérents avec le domaine du sujet (les processus httpd) et le *type* de l'objet (les fichiers dokuwiki). Sur quel type d'opération a lieu le refus ?

---

Au sein de la politique SELinux, identifier les règles d'autorisation liés à cette combinaison, domaine, *type*, objet. L'opération *write* est-elle autorisée ?

```
sesearch -d -allow -s httpd_t -t httpd_sys_content_t -c file
```

---

Plutôt que créer une nouvelle règle dans la politique SELinux, il vaut mieux identifier ce qui est déjà présent et qui peut correspondre au besoin. Tester les 2 commandes suivantes et comparer le nombre d'option disponibles :

```
sesearch -d -allow -s httpd_t -c file -p write
```

```
sesearch -d -R -allow -s httpd_t -t httpd_sys* -c file -p write
```

## UR1 M2 - SSE - Sécurité des systèmes d'exploitation

Le *type* `httpd_sys_rw_content_t` est un candidat générique qui répond au besoin, mais rien n'empêche d'utiliser un autre *type* pour la suite.

Changer le contexte des répertoires *conf* et *data* de Dokuwiki, puis tester que le wiki fonctionne correctement.

```
sudo chcon -Rv --type=httpd_sys_rw_content_t /var/www/html/dokuwiki/conf
sudo chcon -Rv --type=httpd_sys_rw_content_t /var/www/html/dokuwiki/data
```

Cette modification n'est pas réellement permanente. Pour illustrer ce point, taper la commande suivante puis redémarrer la machine.

```
sudo touch /.autorelabel
```

Le wiki ne doit plus fonctionner. Taper les commandes suivantes pour corriger le problème :

```
sudo semanage fcontext -a -t httpd_sys_rw_content_t /var/www/html/dokuwiki/conf/*
sudo semanage fcontext -a -t httpd_sys_rw_content_t /var/www/html/dokuwiki/data/*
sudo restorecon -Rv /var/www/html/dokuwiki/conf
sudo restorecon -Rv /var/www/html/dokuwiki/data
```

La documentation Dokuwiki propose la [procédure](#) suivante pour résoudre le problème rencontré :

```
grep httpd /var/log/audit/audit.log | audit2allow -M mypol
semodule -i mypol.pp
```

Exécuter la commande suivante et analyser les modifications proposées. Que pensez-vous des changements apportés ? Quel pourrait être l'impact ?

```
grep httpd /var/log/audit/audit.log | audit2allow
```

---

---

La commande `sealert` (`sealert -a /var/log/audit/audit.log`) du paquetage *setroubleshoot-server* permet d'analyser les problèmes liés à la politique SELinux. Néanmoins, il vaut mieux comprendre ce que l'on fait avant de suivre les préconisations formulées (elles reposent souvent sur *audit2allow*).

Question bonus : comment sont stockés les mots de passe des utilisateurs Dokuwiki ?

---

---

## 4. Contrôle d'accès en environnement Windows

### Exercice 4-1

Objectif : manipuler les ACL de fichiers Windows

Créer un répertoire D:\Privatebox et mettre en place des autorisations pour répondre au même objectif que l'exercice 3-1.

Créer un répertoire E:\Privatebox et faire de même.

---

---

### Exercice 4-2

Objectif : identifier les privilèges Windows

Dans une session administrateur, lancer une invite de commande et taper la commande

```
whoami /all
```

En parallèle, lancer une invite de commande « élevée » et y taper la même commande. Quelles différences remarquez-vous sur les informations affichées au niveau du *Mandatory Label* et des informations de privilèges ? Cette différence peut-elle avoir un intérêt pour la sécurité ?

---

---

---

---

---

Concernant l'appartenance aux groupes, le compte IEUser est-il membre de groupes qui soient locaux à la machine msedgewin10 ?

---

---

Lancer l'utilitaire secpol.msc et aller dans Local Policies\Users Rights Assignment. Les privilèges accordés au groupe « Administrators » sont-ils nombreux ?

---



Le compte `sshd_server` détient-il des privilèges très sensibles ? Si oui, lesquels ? Cela rend-il ce compte plus sensible ou moins sensible que les utilisateurs membre du groupe *Administrators* ?

---

---

---

---

### **Exercice 4-3**

Objectif : tester un modèle MAC avec Windows avec « Windows Mandatory Integrity Controls »

Taper les commandes suivantes :

```
C:\tp\SysinternalsSuite\accesschk.exe -v -d C:\Users\IEUser\AppData\Local
C:\tp\SysinternalsSuite\accesschk.exe -v -d C:\Users\IEUser\AppData\LocalLow
```

Comparer les permissions et le niveau d'intégrité (*mandatory level*) renvoyé pour les deux répertoires considérés.

---

---

Créer le répertoire `C:\Users\IEUser\AppData\LocalHigh` et lui attribuer un niveau d'intégrité haut (dans une invite de commande « élevée ») :

```
icacls C:\Users\IEUser\AppData\LocalHigh /setintegritylevel H
```

Comparer les permissions et le niveau d'intégrité

---

---

Ouvrir Notepad à partir du menu démarrer et taper du texte dedans. Enregistrer ce texte dans les répertoires :

- `C:\Users\IEUser\AppData\LocalLow`
- `C:\Users\IEUser\AppData\Local`
- `C:\Users\IEUser\AppData\LocalHigh`

Que constatez-vous ? Les permissions positionnées sur chacun de ces répertoires en sont-elles la cause ?

---

---

Dans l'outil Process Explorer, afficher la colonne « Integrity Level » et observer celui du processus notepad.exe. Cela permet-il d'expliquer le problème rencontré sur le répertoire C:\Users\IEUser\AppData\LocalHigh ?

---

---

Sur la base des opérations effectués, effectuer des tests permettant de vérifier si le contrôle d'accès DAC est effectué avant le contrôle d'accès MAC (cas des systèmes Linux) ou vice-versa.

Maintenant, l'affectation d'une étiquette MAC va concerner un programme. Avant de manipuler ses informations de sécurité, une sauvegarde de l'ACL peut s'avérer utile :

```
icacls C:\windows\notepad.exe /save C:\Users\IEUser\acl-notepad.sav
```

Après s'être approprié le fichier C:\Windows\notepad.exe, donner au groupe « Administrators » le contrôle total sur ce fichier. Ces actions peuvent être réalisées par l'interface graphique ou bien en ligne de commande (dans une invite de commande « élevée ») :

```
takeown /F C:\windows\notepad.exe /A  
icacls C:\windows\notepad.exe /grant BUILTIN\Administrators:F
```

Exécuter la commande suivante :

```
icacls C:\windows\notepad.exe /setintegritylevel L
```

Lancer explicitement ce programme (le notepad du menu démarrer est situé dans C:\Winows\system32), taper quelques caractères et essayez d'enregistrer le document dans le répertoire C:\Users\IEUser\AppData\Local puis dans C:\Users\IEUser\AppData\LocalLow. Que se passe-t-il alors ? Quel est le niveau d'intégrité du processus notepad.exe ?

---

---

---

La commande suivante permet de restaurer les informations de sécurité d'origine :

```
icacls C:\windows\ /restore C:\Users\IEUser\acl-notepad.sav
```