

SSE - Contrôle d'accès

TP1

Master 2 Cybersécurité

2018 - 2019

Encadré par :
Josselin MARIETTE

Réalisé par :
Manon DEROCLES
Alexis LE MASLE

Table Des Matières

Contrôle d'accès en environnement Linux	2
Exercice 3-1	2
Exercice 3-2	3
Exercice 3-3	5
Exercice 3-4	5
Exercice 3-4	8
Exercice 3-5	10
Contrôle d'accès en environnement Windows	13
Exercice 4-1	13
Exercice 4-2	13
Exercice 4-3	13

Contrôle d'accès en environnement Linux

Exercice 3-1

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.

```
umask 077
```

Il faut ajouter cette ligne dans le fichier .bashrc de l'utilisateur "tp". De cette manière, on paramètre le mode de création de fichiers et répertoires de l'utilisateur avec des droits

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.)

```
mkdir /tmp/privatebox  
cd /tmp/privatebox  
setfacl d:u::rw-,g:---,o:--- .
```

Commande pour créer le répertoire avec des droits précis

drwxr-xr-x	2	tp	tp	4096	nov.	5	2017	Images
-rw-----	1	tp	tp	0	nov.	23	16:51	question1-sse.txt
drwxr-xr-x	2	tp	tp	4096	nov.	5	2017	Téléchargements

Capture des droits qu'a le dossier nouvellement créer

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

Le dossier ayant les droits d'exécution pour tous permet aux utilisateurs de supprimer les fichiers s'y trouvant.

Exercice 3-2

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

```
tp@ses-tp2-debian:/tmp/privatebox$ find / -type f \( -perm -4000 -a -uid 0 \) -print 2> /dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/pkexec
/usr/lib/eject/dmccrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/ntfs-3g
/bin/su
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
```

Résultat de la commande

On peut voir que le binaire “ping” fait partie de la liste.

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
```

```

tp@ses-tp2-debian:/tmp/privatebox$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.045 ms
^C
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.044/0.044/0.045/0.005 ms

```

Ping fonctionnel avant les commandes ci dessus

Avec la première commande, on enlève le bit *suid* du binaire “ping”, ce qui enlève aux utilisateurs le droit d’exécuter cette commande avec les droits du super-utilisateur.

La seconde commande permet d’ajouter la *capacité* “p” pour *permitted*. Cela permet d’utiliser le ping sans les droits root.

```

tp@ses-tp2-debian:/tmp/privatebox$ sudo chmod u-s /bin/ping
[sudo] Mot de passe de tp :
tp@ses-tp2-debian:/tmp/privatebox$ ping 127.0.0.1
ping: socket: Opération non permise

```

Ping devenu inutilisable pour un utilisateur normal après la première commande

```

tp@ses-tp2-debian:/tmp/privatebox$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.037 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.041 ms
^C
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.037/0.039/0.041/0.007 ms

```

Ping redevenu fonctionnel avec la seconde commande

Les opérations réalisées sur les droits du fichier /bin/ping présentent-t-elles un intérêt ? Si oui, lequel ? Si besoin, la description des capacités Linux s’obtient avec `man capabilities`.

On rétablit les droits originel avec les commandes:

```

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

```

On refait les mêmes manipulations sur la machine CentOS:

Cette fois ci, la commande `find` ne nous renvoie aucun résultat. Le ping fonctionne normalement.


```
[tp@sse-tp2-centos ~]$ sudo chmod u-s /bin/ping
[sudo] Mot de passe de tp :
[tp@sse-tp2-centos ~]$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.072 ms
^C
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.072/0.072/0.073/0.008 ms
```

Malgré la suppression du suid root, le ping fonctionne

```
[tp@sse-tp2-centos ~]$ sudo setcap cap_net_raw=p /bin/ping
[tp@sse-tp2-centos ~]$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.073 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.080 ms
^C
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.073/0.076/0.080/0.009 ms
```

Dans tous les cas le ping fonctionne sur les machines CentOS

Exercice 3-3

Objectif: identifier les processus utilisant des capacités.

Exercice 3-4

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

On exécute les commandes:

```
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
```

Création d'un fichier hébergeant un système de fichiers

```

tp@ses-tp2-debian:~/Bureau$ sudo dd if=/dev/zero of=/var/fichier.part bs=512 count=100k
[sudo] Mot de passe de tp :
102400+0 enregistrements lus
102400+0 enregistrements écrits
52428800 bytes (52 MB, 50 MiB) copied, 0,243425 s, 215 MB/s
tp@ses-tp2-debian:~/Bureau$ sudo losetup /dev/loop0 /var/fichier.part
tp@ses-tp2-debian:~/Bureau$ sudo mkfs -t ext4 /dev/loop0
mke2fs 1.43.4 (31-Jan-2017)
Rejet des blocs de périphérique : complété
En train de créer un système de fichiers avec 51200 1k blocs et 12824 i-noeuds.
UUID de système de fichiers=070c73fc-76d1-44ee-981e-769651ce1891
Superblocs de secours stockés sur les blocs :
    8193, 24577, 40961

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (4096 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

tp@ses-tp2-debian:~/Bureau$ sudo mount -t ext4 /dev/loop0 /mnt

```

Résultat des commandes

Après avoir exécuté ces commandes, on a créé un fichier contenant un système de fichiers. Nous pouvons vérifier les droits par défaut attribué à ce fichier /dev/loop0 avec la commande:

```

tp@ses-tp2-debian:~/Bureau$ mount | grep /mnt
/dev/loop0 on /mnt type ext4 (rw,relatime,data=ordered)

```

Commande permettant de vérifier les droit par défaut

Nous avons donc les droits d'écriture et de lecture par défaut sur ce système de fichiers.

```

tp@ses-tp2-debian:~/Bureau$ sudo cp /usr/bin/sudo /mnt
tp@ses-tp2-debian:~/Bureau$ sudo chmod 4755 /mnt/sudo
tp@ses-tp2-debian:~/Bureau$ /mnt/sudo id
uid=0(root) gid=0(root) groupes=0(root)
tp@ses-tp2-debian:~/Bureau$ sudo cp /bin/nc /mnt/
tp@ses-tp2-debian:~/Bureau$ sudo setcap cap_net_bind_service=ep /mnt/nc
tp@ses-tp2-debian:~/Bureau$ sudo cp /bin/sh /mnt/su_sh
tp@ses-tp2-debian:~/Bureau$ sudo chmod 4755 /mnt/su_sh

```

Commandes simulant la préparation d'un attaquant

Nous écrivons un script **/mnt/evil-script.sh** qui utilise l'interpréteur que nous avons créé appelé **/mnt/su_sh** qui nous permet d'afficher le contenu de **/etc/shadow**.

```

#!/mnt/su_sh
cat /etc/shadow

```

Contenu du fichier /mnt/su_sh

Pour l'exécuter nous devons utiliser la commande "**sudo /mnt/su_sh**".

```

nc -lp 80

```

Commande permettant une écoute sur le port 80 avec netcat

Nous exécutons la commande **wget http://127.0.0.1** , aucune réponse ne nous parvient. Au contraire, la commande netcat qui a obtenu la capacité **cap_net_bind_service=ep** est capable de recevoir la requête http.

```
tp@ses-tp2-debian:~/Bureau$ sudo /mnt/nc -lp 80
GET / HTTP/1.1
User-Agent: Wget/1.18 (linux-gnu)
Accept: */*
Accept-Encoding: identity
Host: 127.0.0.1
Connection: Keep-Alive
```

Résultat de l'écoute netcat ayant la capacité

```
tp@ses-tp2-debian:~/Bureau$ wget http://127.0.0.1
--2018-11-26 10:53:59-- http://127.0.0.1/
Connexion à 127.0.0.1:80... connecté.
requête HTTP transmise, en attente de la réponse...
```

Requête émise

```
tp@ses-tp2-debian:/mnt$ sudo mount -t ext4 /dev/loop0 /mnt -o remount,nosuid
mount: /dev/loop0 est déjà monté ou /mnt est occupé
/dev/loop0 est déjà monté sur /mnt
```

Suppression des permissions suid

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

En tapant "mount | grep /mnt" on obtient le même résultat que précédemment avec le mot "nosuid" en plus. Nous obtenons alors un message nous signifiant que nous ne sommes pas vraiment "root".

```
tp@ses-tp2-debian:/mnt$ /mnt/sudo id
sudo: le uid effectif n'est pas 0. Est-ce que /mnt/sudo est sur un système de f
ichiers avec l'option « nosuid » ou un système de fichiers NFS sans privilèges
root ?
```

```
tp@ses-tp2-debian:/mnt$ ./evil-script.sh
sudo: le uid effectif n'est pas 0. Est-ce que /mnt/sudo est sur un système
de fichiers avec l'option « nosuid » ou un système de fichiers NFS sans pri
vilèges root ?
```

Message affiché lors de l'exécution d'une commande

Après avoir exécuté la commande:

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

Commande exécuté

Nous ne pouvons plus exécuter les commandes précédente.


```
tp@ses-tp2-debian:/mnt$ /mnt/sudo id
bash: /mnt/sudo: Permission non accordée
tp@ses-tp2-debian:/mnt$ /mnt/evil-script.sh
bash: /mnt/evil-script.sh: Permission non accordée
tp@ses-tp2-debian:/mnt$ /mnt/nc -l 80
bash: /mnt/nc: Permission non accordée
```

Impossibilité d'exécuter les commandes

Les droits d'exécution ont été supprimé sur système de fichier.

Exercice 3-4

Objectif : appréhender les commandes courantes SELINUX

Nous utilisons la commande “**sestatus**” pour vérifier l'état courant de SELinux

```
[tp@sse-tp2-centos ~]$ sestatus
SELinux status:                enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                    enforcing
Mode from config file:           enforcing
Policy MLS status:               enabled
Policy deny_unknown status:      allowed
Max kernel policy version:       28
```

Commande permettant de vérifier l'état courant de SELinux

La première ligne de résultat nous informe que SELinux est en status “enabled”.

```
[tp@sse-tp2-centos ~]$ ls -Z
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Bureau
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Documents
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Images
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Modèles
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Musique
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Public
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Téléchargements
drwxr-xr-x. tp tp unconfined_u:object_r:user_home_t:s0 Vidéos
```

Contextes de sécurité des fichiers du répertoire courant

```
[tp@sse-tp2-centos ~]$ id
uid=1000(tp) gid=1000(tp) groupes=1000(tp),10(wheel) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[tp@sse-tp2-centos ~]$ sudo id
[sudo] Mot de passe de tp :
uid=0(root) gid=0(root) groupes=0(root) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Vérification de la persistance de l'identité SELinux en changeant d'identité UNIX

En utilisant la commande “id” puis la commande sous une autre identité avec “sudo”, nous vérifions la persistance de l'identité SELinux. Nous pouvons constater que le contexte de sécurité reste le même.

```
[tp@sse-tp2-centos ~]$ sudo semanage login -l
```

Nom pour l'ouverture de session	Identité SELinux	Intervalle MLS/MCS	Service
__default__	unconfined_u	s0-s0:c0.c1023	*
root	unconfined_u	s0-s0:c0.c1023	*
system_u	system_u	s0-s0:c0.c1023	*

```
[tp@sse-tp2-centos ~]$ sudo semanage user -l
```

Identité SELinux	Étiquetage Préfixe	MLS/ Niveau	MLS/ Intervalle MCS	Rôles SELinux
guest_u	user	s0	s0	guest_r
root	user	s0	s0-s0:c0.c1023	staff_r sysadm_r system_r unconfined_r
confined_r				
staff_u	user	s0	s0-s0:c0.c1023	staff_r sysadm_r system_r unconfined_r
confined_r				
sysadm_u	user	s0	s0-s0:c0.c1023	sysadm_r
system_u	user	s0	s0-s0:c0.c1023	system_r unconfined_r
unconfined_u	user	s0	s0-s0:c0.c1023	system_r unconfined_r
user_u	user	s0	s0	user_r
xguest_u	user	s0	s0	xguest_r

Consultation des règles d'attributions des identités SELinux

Nous créons ensuite un nouvel utilisateur nommé “util1” et nous l’ajoutons à SELinux en tant que “user_u”. Nous vérifions son identité SELinux:

```
[util1@sse-tp2-centos tp]$ id
uid=1001(util1) gid=1001(util1) groupes=1001(util1) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

commande “id” connecté en tant que util1

```
[tp@sse-tp2-centos ~]$ sudo -u util1 id
uid=1001(util1) gid=1001(util1) groupes=1001(util1) contexte=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Commande “id” en changeant d'identité sous le nom de “util1”

```
[tp@sse-tp2-centos ~]$ sudo semanage login -l
```

Nom pour l'ouverture de session	Identité SELinux	Intervalle MLS/MCS	Service
__default__	unconfined_u	s0-s0:c0.c1023	*
root	unconfined_u	s0-s0:c0.c1023	*
system_u	system_u	s0-s0:c0.c1023	*
util1	user_u	s0	*

L'identité SELinux de “util1” est user_u

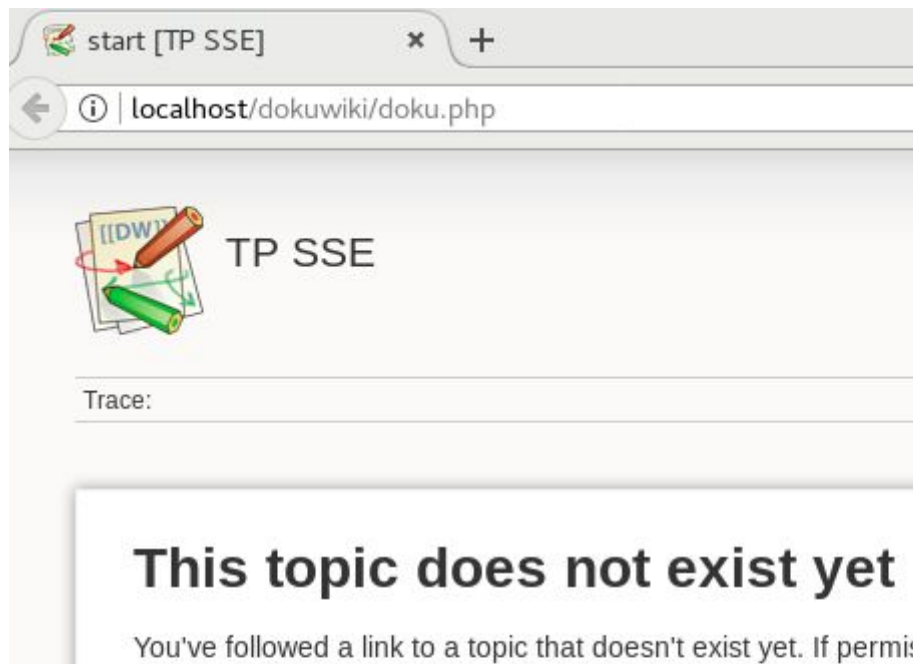
Nous constatons que la commande “id” nous a renvoyé le même résultat pour l’identité SELinux en tant que “util1” ou en tant que “root” exécutant la commande sous le nom de “util1”.

Exercice 3-5

Nous mettons la sécurité SELinux en mode **permissive**, et nous accédons au wiki via l’adresse “localhost/dokuwiki/doku.php”

```
sudo setenforce 0
```

Commande permettant de mettre le mode **permissive**

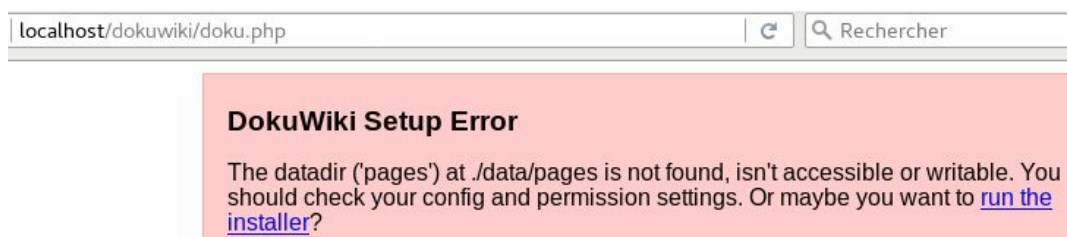


Le wiki fonctionne en mode **permissive**

Le wiki fonctionne bien tant que nous sommes en mode **permissive**. Nous allons maintenant mettre SELinux en mode **enforcing**.

```
sudo setenforce 1
```

Commande permettant de mettre le mode **enforcing**



Le wiki ne fonctionne pas en mode **enforcing**

En mode **enforcing**, le wiki ne fonctionne plus et une erreur apparaît.

Nous vérifions si il existe un module ciblant **dokuwiki** grâce à la commande suivante:

```
sudo semanage module -l  
sudo semodule -l
```

Commande permettant de vérifier si un module par défaut cible dokuwiki

Or dans la liste qui nous est affiché, nous ne voyons aucun module ciblant **dokuwiki**.
Il n'existe pas de module doku ou dokuwiki.

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

Commande affichant les lignes contenant "denied" parmi les 100 dernières lignes de audit.log

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 ?

Nous voyons que le type de requête "denied" est en "httpd". Il va donc falloir créer une règle acceptant ce type de requête. Nous voyons également que l'opération "write" est dans la liste des "denied".

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
```

```
Found 1 semantic av rules:  
allow httpd_t httpd_sys_content_t : file { ioctl read getattr lock open } ;
```

Résultat obtenue

Cette règle affiche la règle d'autorisation liée à la combinaison "httpd_t", "httpd_sys_content_t" et "file", on peut y lire "read" sur l'objet "file". Mais l'opération "write" n'est pas précisée dans la règle.

Commande permettant d'identifier les règles déjà présentes.

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

Nous disposons de 73 options avec cette commande.

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

Nous disposons de 10 options avec la cette commande.

```
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
```

Commandes permettant de changer de contexte du répertoire conf et data

Après avoir exécuté ces commandes, le site web est de nouveau accessible.
Les modifications n'étant pas permanente, en créant un fichier nommé ".autorelabel" à la racine du système de fichier et en redémarrant la machine les configurations se rétablissent.

```
sudo touch /.autorelabel
```

Après redémarrage de la machine, le site web ne fonctionne plus car les règles se sont rétablies.

```
Nothing to do
[tp@sse-tp2-centos ~]$ sudo grep httpd /var/log/audit/audit.log | audit2allow -M mypol
[sudo] Mot de passe de tp :
***** IMPORTANT *****
To make this policy package active, execute:
semodule -i mypol.pp
```

```
[tp@sse-tp2-centos ~]$ sudo semodule -i mypol.pp
```

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

```
[tp@sse-tp2-centos ~]$ sudo grep httpd /var/log/audit/audit.log | audit2allow

#===== httpd_t =====

#!!!! This avc is allowed in the current policy
allow httpd_t httpd_sys_content_t:dir { add_name create remove_name rmdir write };


#!!!! This avc is allowed in the current policy
allow httpd_t httpd_sys_content_t:file write;
```

Contrôle d'accès en environnement Windows

Exercice 4-1

Créer un répertoire D:\Privatebox et mettre en place des autorisations pour répondre au même objectif que l'exercice 3-1. L'utilisation du pseudo-utilisateur « CREATEUR PROPRIÉTAIRE » (CREATOR OWNER en anglais) est conseillé.


Name: D:\Privatebox

Owner: IEUser (MSEDGEWIN10\IEUser)  [Change](#)

Permissions Auditing Effective Access

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Type	Principal	Access	Inherited from	Applies to
 Allow	CREATOR OWNER	Full control	None	This folder, subfolders and files

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

Exercice 4-2

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 ?

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

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

Comparer les permissions et le niveau d'intégrité.