Architecture d'un système d'exploitation (UE S3)

TP4

Introduction à la sécurité informatique en environnement HPC

Etudiant : Romain Pereira *Encadrant :* M. F. Combeau

17/11/2018

Table des matières

1 Réponses aux questions 1 à 22 (faites en cours)

4. Pour se connecter, on a utilisé l'authentification système, à savoir, les **Pluggable Authentication Modules (PAM)**

5. Le mot de passe *root* est stocké dans le fichier '/etc/shadow' sous forme d'un hachage. Le mot de passe est haché avant d'être stocké car il est stocké localement (chaque machine connecté au réseau possède le fichier '/etc/shadow' localement).

6. Je ne peux pas déterminer facilement le mot de passe de *Bob*, car il est également stocké sous forme de hash. Cependant, à l'aide d'une méthode de brute force, on pourrait eventuellement déterminer des chaîne de caractères qui ont le même hash que celui de *Bob*. Le mot de passe de *Bob* serait donc potentiellement l'une de ces chaînes de caractères, ou pas.

On ne peut pas déterminer le mot de passe d'*Alice*, car elle n'en a pas. En effet, le champ devant correspondre à son mot de passe dans le fichier '/etc/shadow' est ***** (ou **!!**). Les fonctions de hachages utilisées par les PAM ne donneront jamais des hashs avec ces caractères ('*' et '!'). En spécifiant des caractères impossibles à obtenir par hachage, on désactive de façon indirecte l'authentification par mot de passe pour l'utilisateur : jamais un mot de passe ne donnera ce hash. L'utilisateur est en quelque sorte 'banni'.

7. Si l'on passe cette ligne de **sufficient** à **required**, alors *root* ne pourra plus se logger s'authentifier. En effet, l'identifiant utilisateur (*uid*) de *root* vaut 0. A la ligne suivante, on refuse l'authentification des utilisateurs dont l'uid est inférieur à 1000, le test ne passera donc pas. est donc inférieur à 1000 : les tests suivant ne passeront pas.

Finalement, *root* ne pourra plus se logger (à cause de la ligne *pam_ deny.so*)

8. Oui, on arrive à se connecter sur le compte de *Bob* à partir du compte *root*

```
> less /var/log/secure
[...]
Accepted password for bob form ::1 port 47404 ssh2
pam_unix(sshd:session) : session opened for user bob by (uid=0)
[...]
```

La connexion sur le compte *bob* à partir de l'utilisateur *root* (uid=0) est bien tracée dans les logs.

10. En passant de **required** à **requisite**, rien ne va changer (dans les 2 cas, tous les modules utilisant ce contrôle doivent passer avec succès pour que la vérification soit accordée)

En passant de **requisite** à **sufficient**, ...

11. Alice n'a pas de mot de passe (cf '!!' et '*')

```
> ssh alice@localhost
Enter passphrase for key '/root/.ssh/id_rsa'
```

Donc ssh utilise son propre système d'authentification, et non pas les PAM systèmes. La clef privé ssh (RSA) a été chiffré à l'aide d'un mot de passe (pour éviter le vol de clef)

Une fois connecté sous Alice, on a la clef dans '.ssh/authorized_keys'

Logs :

```
sshd : Accepted publickey for alice from ::1 port 47410 ssh2: RSA SHA256:/JdhKRDYT.
sshd : pam_unix(sshd:session): session opened for user alice by (uid=0)
```

12. Protection '775' ('002' est le complément à 8 du masque octal)

```
> drwxrwxr-x 2 bob 4096 19 nov. 15:22
> umask
'0002'
```

13. Oui on a le droit de créer, car on est sous root (root à le droit d'écrire partout)

```
> -rw-r--r--. 1 root    0 19 nov. 15:52 toto
```

Ce ficher appartient à root Linux applique le masque octal (umask), mais par défaut, il est non executable par mesure de sécurité.

14. Le système notifie qu'on a pas les droits d'écriture sur le fichier (entant que 'bob') Mais le fichier a été supprimé, car 'bob' à le droit d'écriture dans le dossier

```
> rm toto
'rm : supprimer fichier vide (prot g   en   criture ) "␣toto␣" ? y
```

15. Il suffit d'utiliser l'utilitaire 'chmod'

```
> chmod 1775 test
> drwsr-sr-t. 1 bob    0 19 nov. 15:52 test
```

16. Oui, on a toujours le droit, et le fichier appartient à 'root'

```
> -rw-r--r--. 1 root    0 19 nov. 15:52 toto
```

17. Oui on a toujours le droit.

18. Il suffit d'utilliser l'utilitaire 'chown'

```
 > chown root test
 > drwsr-sr-t. 1 root    0 19 nov. 15:52 test
```

19. Non **bob** ne peut pas l'effacer, car il y a le sticky bit (le 't') sur le repertoire. Il faut être propriétaire du dossier pour supprimer un fichier. Dans ce cas, **root** est le propriétaire du dossier, donc **bob** ne peut pas effacer de fichiers.

20. */bin/passwd* : modifie des fichiers auxquelles seul root à accès (ex : */etc/shadow*)

21. **Bob** peut partager son fichier en utilisant les ACL.

Pour pouvoir parcourir le dossier

```
> setfacl -m u:alice:rx dossier
```

Pour pouvoir ecrire dans le fichier du dossier

```
> setfacl -m u:alice:rw pour_alice
```

22. On remarque que l'utilisateur 'sftp' est ch-rooté dans '/home/sftp'

Lors d'une connection ssh :

```
This service allows sftp connections only.
Connection to localhost closed.
```

(le service n'autorise que des connections sftp)

Lors d'une connection sftp :

```
'Connected to localhost'
```

Avec l'utilisateur **bob**, le chroot est '/', alors qu'avec **sftp**, le chroot avec '/home/stfp'

2 Réponses aux questions XX à YY (bonus)