TP n° X

R3 : Exercices supplémentaires

Exercice 1 [Blacklisting d'adresses IP]

Les opérations d'authentification sont journalisées dans le répertoire `/var/log`. Les plus récentes sont dans le fichier `auth.log`, les plus anciennes sont dans les autres fichiers `auth.log.*`.

On y trouve notamment les échecs de connexion. On remarque par exemple des tentatives répétées utilisant des logins n'existant pas dans le système.

Vous pouvez récupérer des logs à l'emplacement suivant :

`http://www-lipn.univ-paris13.fr/~coti/cours/auth.log.tgz`

1. Repérer les échecs dans le fichier de log
2. Comment peut-on les extraire de ces fichiers (par exemple, avec grep) ?
3. Écrire un script qui permet d'extraire :
   - les logins erronés
   - les adresses IP d'où sont venues ces tentatives

On cherche à interdire la connexion depuis des adresses IP d'où proviennent plusieurs tentatives avec des logins erronés. La commande pour blacklister une adresse IP en utilisant le firewall de Linux est :

```
/sbin/iptables -I INPUT -s <adresse_ip> -j DROP
```

4. Écrire un script qui prend en paramètre un seuil de tolérance et qui interdit l'accès à toute machine ayant trop de tentatives erronées (un nombre d'échecs supérieur à ce seuil de tolérance).

C'est typiquement le genre de tâche que l'on veut effectuer régulièrement, pour réagir à des tentatives d'intrusion. Sous Unix, `cron` est un utilitaire qui exécute des tâches à intervalles régulier. Ces tâches sont définies dans le fichier `/etc/crontab`.

5. Comment définit-on quelles tâches sont lancées toutes les heures ? Tous les jours ? Toutes les semaines ? Comment définit-on à quel moment précisément elles doivent être lancées ?
6. Modifiez la configuration de `cron` pour que votre script de blacklisting d'IP soit lancé toutes les 10 minutes.

Exercice 2 [DeDup]

Écrire un script dont le but est de diminuer l'espace disque utilisé par d'éventuelles copies d'un fichier. Si le fichier est présent en plusieurs exemplaires, on n'en garde qu'un et on remplace les autres par un lien hard.

Pour déterminer si deux fichiers sont identiques, on effectuera les deux tests suivants :

- On teste si la taille des deux fichiers est identique
- Si c'est le cas, on compare les md5sum des deux fichiers

Attention aux fichiers dont le nom contient des espaces.

Exercice 3 [Dessin dans le terminal]

Écrire un script vous permettant de dessinez à l'écran (dans le terminal) un losange dont la largeur est donnée par l'utilisateur :

```
coti@pistache:~$ ./losange.sh 9
```

```
          .
         . .
        . . .
       . . . .
      . . . . .
     . . . . . .
    . . . . . . .
   . . . . . . . .
  . . . . . . . . .
  . . . . . . . . .
   . . . . . . . .
    . . . . . . .
     . . . . . .
      . . . . .
       . . . .
        . . .
         . .
          .
```

Exercice 4 [Figlet]

Figlet permet d'afficher du texte dans un terminal dans des polices "exotiques" ou avec des décorations. Par exemple :

```
coti@pistache:~/soft/figlet-2.2.4$ ./figlet -f fonts/bubble.flf "R3 c'est chouette"
  _   _     _   _   _   _   _     _   _   _   _   _   _   _   _
 / \ / \   / \ / \ / \ / \ / \   / \ / \ / \ / \ / \ / \ / \ / \
( R | 3 ) ( c | ' | e | s | t ) ( c | h | o | u | e | t | t | e )
 \_/ \_/   \_/ \_/ \_/ \_/ \_/   \_/ \_/ \_/ \_/ \_/ \_/ \_/ \_/
```

Téléchargez Figlet à l'emplacement suivant :

```
ftp://ftp.figlet.org/pub/figlet/program/unix/figlet-2.2.4.tar.gz
```

1. Décompressez l'archive et allez dans le répertoire créé lors de la décompression.
2. Compilez Figlet avec la commande `make`.
3. Écrivez un script figlet.sh qui prend un texte en paramètre et qui affiche ce texte avec la décoration de la police bubble.flt *sans faire d'appel à l'exécutable figlet.* Vous devez réaliser la même chose entièrement en bash.

Exercice 5 [BibTeX2HTML]

Un format "standard" pour stocker les informations bibliographiques est le format BibTeX. Il permet de donner un nombre variable d'informations sur une publication (ouvrage, article, rapport...).

Vous trouverez un fichier BibTeX à l'emplacement suivant :

`http://www-lipn.univ-paris13.fr/~coti/myBib.bib`

Le but de cet exercice est d'écrire un script bash `bibtex2html.sh` qui prend un fichier BibTeX tel que celui qui vous est fourni ici en argument et produit une page html sur la sortie standard.

Chaque entrée du fichier est comme suit (les valeurs des champs peuvent être entre guillemets ou entre accolades) :

```
@typeDeDocument{ reference,
   champ1 = "contenu du champ",
   champ2 = {contenu de ce champ},
   champ3 = "contenu d'un troisième champ",
}
```

Les champs présents dépendent du type de document. Par exemple, un article publié dans une revue (`article`) aura le titre de la revue dans son champ `journal`. Un article publié dans les actes d'une conférence (`inproceedings`) aura le titre des actes dans son champ `booktitle`. Par ailleurs, les champs sont optionnels : il n'y a aucune obligation à ce qu'ils soient tous présents.

On veut que chaque entrée dans le fichier soit présenté sous la forme : auteurs: *titre*, nom de la revue ou titre des actes, lieu, mois année, *note s'il y en a une*. Si ces champs sont présents, on peut aussi mettre le volume, l'issue et les pages. Par exemple, la première entrée du fichier qui vous est fourni sera :

Franck Butelle and Camille Coti, *A Model for Coherent Distributed Memory For Race Condition Detection*, in proceedings of the 13th Workshop on Advances in Parallel and Distributed Computational Models (APDCM'11), Anchorage, Ak, May 2011, *to appear*.

Les entrées doivent être formatées sous forme d'une liste. En HTML, une liste est réalisée de la façon suivante :

```
<ul>
  <li>Premier élément de la liste</li>
  <li>Deuxième élément de la liste</li>
</ul>
```

1. Écrire un script `bibtex2html.sh` qui prend toutes les entrées d'un fichier BibTeX (en ignorant les autres lignes) et produit sur la sortie standard une page HTML contenant la liste des entrées.

Le document HTML a la structure suivante :

```
<html>
  <head>
    <title>Titre de la page</title>
  </head>
  <body>

  </body>
</html>
```

2. Modifiez le script `bibtex2html.sh` de façon à inclure le titre du fichier BibTeX entre les balises `<title>` et `</title>` et à mettre la liste des entrées entre les balises `<body>` et `</body>`.

Vous remarquerez que dans le fichier BibTeX fourni, les entrées sont classées par catégories séparées par :

```
----------------------------
--- Titre de la categorie ---
----------------------------
```

On veut avoir non plus une liste globale, mais une liste par catégorie. Chaque liste doit être précédée par le titre de la catégorie. On met le titre de la liste sous forme d'un paragraphe précédent le début de la liste :

```
<p>
    Titre de la categorie
</p>
```

3. Modifiez le script `bibtex2html.sh` de façon à avoir une liste par catégorie d'entrées, en donnant le titre de la catégorie avant le début de la liste.

Exercice 6 [Compilation du noyau]

Le noyau est le cœur d'un système d'exploitation : c'est lui qui ordonnance les processus, gère la mémoire... Dans le cas d'un système GNU/Linux, le noyau utilisé est Linux.

Les sources du noyau sont distribuées sur le site :

`http://www.kernel.org`

De nombreuses possibilités sont fournies dans le noyau : une large gamme de pilotes matériels, d'options pour diverses plate-formes (embarqué, radio amateur...). Tout n'est pas utile pour un système donné. C'est pourquoi la configuration du noyau est une étape cruciale : un noyau plus petit prendra moins de place en mémoire et sera chargé plus rapidement au démarrage.

Certaines fonctionnalités peuvent être mises en place sous forme de *modules* : un module n'est chargé que si les fonctionnalités qu'il implémente sont appelées. Cependant, il n'est pas forcément possible d'utiliser des modules pour tout. Par exemple, pour charger un module il faut le lire sur le disque dur. Par conséquent, les pilotes du système de fichier ne peuvent pas être dans des modules : il faut les avoir chargé pour lire les modules.

1. Téléchargez les sources du noyau Linux
2. Décompressez-les dans `/usr/src`
3. Créez un lien symbolique `/usr/src/linux` (ou modifiez le lien existant s'il y en a déjà un) vers les sources que vous venez de décompresser
4. Placez-vous dans le répertoire `/usr/src/linux`

Vous êtes maintenant dans les sources du noyau. Il existe plusieurs façons de choisir les options et les réglages que vous allez compiler. Vous pouvez écrire directement dans le fichier de configuration ou utiliser une interface. Trois interfaces sont disponibles : en mode graphique, en mode ncurses (dans le terminal) et en mode texte. On vous propose ici d'utiliser l'interface ncurses, qui est disponible sur à peu près tous les systèmes (un serveur X n'est pas forcément utilisable dans toutes les situation) et plus agréable à utiliser que le mode texte.

Il manque quelques fichiers sur les machines sur lesquelles vous travaillerez. En effet, vous aurez besoin de la bibliothèque `libncurses` et de ses fichiers d'en-tête. Or, si la bibliothèque est bien installée, les fichiers d'en-tête ne sont pas présents. Vous devez donc les installer au préalable. Cette opération doit être effectuée en tant que super-utilisateur.

5. Téléchargez l'archive contenant `ncurses` en vous situant dans le répertoire `/root` avec la commande :
   `wget http://ftp.gnu.org/pub/gnu/ncurses/ncurses-5.7.tar.gz`
6. Décompressez l'archive et placez-vous dans le répertoire extrait.

7. Générez les options de configuration et de configuration en exécutant le script `configure` présent dans le répertoire. Cette commande teste la présence des dépendances ainsi que quelques paramètres du système, puis génère les fichiers de compilation (`Makefiles`).

8. Compilez avec la commande `make`. La compilation va échouer : ce n'est pas grave, nous n'avons pas besoin de tout compiler.

9. Installez ce qui a été compilé avec la commance `make install`. Cette commande va également échouer, toutefois ce dont on a besoin aura été installé. En effet, c'est la compilation de la bibliothèque, déjà présente sur le système, qui échoute. La copie des fichiers d'en-tête a été effectuée avant.

Vous pouvez à présent passer à la configuration de votre noyau. Retournez dans le répertoire `/usr/src/lunix`.

10. Appelez l'interface de sélection des options avec la commande `make menuconfig`

11. Allez dans toutes les catégories et sélectionnez les options de votre choix. Le but ici est d'obtenir un noyau qui soit le plus petit possible tout en étant fonctionnel. Lisez les descriptions des options disponibles et faites dans un premier temps un choix "large" : assurez un noyau fonctionnel, vous réduirez sa taille plus tard. Les choix sont sauvegardés dans le fichier `.config`.

12. Compilez le noyau avec la commande `make bzImage`

13. Compilez les modules avec la commande `make modules`

14. Installez ces modules avec la commande `make modules_install`. Les modules et leurs fichiers de configuration sont copiés dans le répertoire `/lib/modules/<numéro de version>`. Cette commande doit être effectuée en tant que super-utilisateur (root).

Vous trouverez l'image ainsi créée dans le répertoire `arch/i386/boot`. Il y a deux façons de l'installer : automatiquement (`make install`) ou manuellement, en copiant les fichiers dans le répertoire où sont stockés les outils de démarrage (généralement `/boot`).

15. Installez le noyau que vous venez de compiler avec la commande `make install`. Quels fichiers viennent d'être créés dans `/boot` ? À quoi correspondent-ils ?

Au démarrage du système, un système minimal est chargé en mémoire afin de pouvoir charger le reste du système (à commencer par le noyau). C'est ce système qui est contenu dans le fichier `initrd`, et qui est ensuite monté sous forme d'un pseudo système de fichiers au démarrage.

16. Dans le répertoire `/boot`, crééz un `initrd` avec la commande `mkinitrd -o initrd.img-<version> <version>`.

Il faut maintenant modifier la configuration du gestionnaire de démarrage (*bootloader*). Celui-ci est situé, sur les architectures PC traditionnelles, dans une zone spéciale du disque dur principal appelé *Master Boot Record* (MBR). Si plusieurs systèmes sont installés (Windows, plusieurs version de Linux...), il permet de choisir sur lequel démarrer. Sa configuration doit donc indiquer le chemin vers l'initrd à charger et le noyau à démarrer, ainsi que sur quel disque et quelle partition ils se trouvent, ainsi que les options à passer au noyau.

17. Les deux gestionnaires de démarrage les plus répandus sont Grub et LILO. Lequel est utilisé sur votre machine ?

18. Modifiez le fichier de configuration correspondant au gestionnaire de démarrage installé sur votre machine : `/boot/grub/menu.lst` ou `/etc/lilo.conf`. Ajoutez une entrée pour pouvoir démarrer sur votre noyau *mais ne retirez pas la possibilité de démarrer sur l'ancien noyau !*

19. Actualisez le gestionnaire de démarrage pour prendre en compte ces modifications dans le MBR : `update-grub` ou `lilo`.

20. Si tout va bien, redémarrez et sélectionnez votre nouveau noyau au démarrage.

Si votre noyau contient toutes les fonctionnalités nécessaires à votre système et que vous l'avez configuré correctement, votre système va redémarrer. Sinon, redémarrez sur l'ancien noyau et modifiez la configuration du noyau.

Si vous modifiez la configuration du noyau, vous n'avez pas à refaire toutes ces étapes. Recompilez simplement le noyau et les modules (sans oublier de les installer) et copiez l'image créée (`vmlinuz`) à la place de celui que vous aviez précédemment copié dans `/boot`.

21. Une fois que vous avez réussi à compiler un noyau fonctionnel, modifiez les réglages afin d'obtenir un noyau le plus petit possible tout en restant fonctionnel. Quelle taille obtenez-vous ? Quel est le temps de démarrage de ce noyau ?

Exercice 7 [Multi-utilisateurs d'un ensemble de machines]

NB : cet exercice est particulièrement intéressant s'il est réalisé sur plusieurs machines. Vous êtes par conséquent encouragé à travailler avec un autre membre du groupe et à utiliser vos deux machines.

La gestion des utilisateurs sur plusieurs machines pose plusieurs problèmes au niveau de la gestion des utilisateurs et de leurs données. Dans cet exercice nous allons en aborder deux aspects : l'authentification des utilisateurs, et la disponibilité de leurs fichiers.

Admettons qu'on souhaite ajouter un nouvel utilisateur dans le système. La méthode simple consiste à l'ajouter sur une machine avec la commande `adduser`. Cependant, cette commande ne l'ajoute que sur la machine locale. Si on souhaite ajouter cet utilisateur sur plusieurs machines, il faut effectuer cette commande sur *toutes* les machines auxquelles on souhaite lui donner accès.

Une méthode simple et naïve consiste à synchroniser entre les machines les fichiers contenant les informations sur les utilisateurs.

1. Quels sont ces fichiers ?

2. À quels autres moments cette synchronisation doit-elle être effectuée ? Comment peut-on automatiser cette synchronisation ?

3. Quel problèmes sont posés par cette méthode ?

Il existe un système de gestion des utilisateurs appelé *NIS* (Network Information Service). NIS permet d'ajouter des utilisateurs "simplement", à l'aide de la commande habituelle `adduser`, et de maintenir une vision cohérente des utilisateurs autorisés à utiliser ces machines.

L'organisation est la suivante :

- Un serveur maintient la liste des utilisateurs et la propage aux clients
- Les clients interrogent le serveur

Lorsqu'un utilisateur cherche à se connecter sur une machine cliente, celle-ci interroge le serveur et lui demande si cet utilisateur est bien autorisé à se connecter au système en utilisant le mot de passe qu'il a fourni.

Les commandes et les démons NIS ont des noms commençant par `yp` (le nom d'origine de NIS était *Yellow Pages*, mais il était déjà déposé).

4. Sur la machine serveur, installez un serveur NIS (`ypserv`) en utilisant le gestionnaire de paquets de Mandriva `urpmi`.

5. Sur les machines clientes, installez les outils clients NIS (`yp-tools, portmap et ypbind`) en utilisant `urpmi`.

Vous devez maintenant définir le domaine NIS dans lequel se trouvent les machines. Le fichier à éditer se trouve à l'emplacement `/etc/sysconfig/network`. Ajoutez le nom du domaine dans lequel vous voulez mettre vos machines : `NISDOMAIN=lenom`

NFS