ESIEE Paris
Unité 3R-IN3 2020-2021

TP noté - Durée 3h
11 juin 2021

Les 3 exercices du TP sont indépendants. Le travail doit être fait en binôme, conformément à la table d'affectation communiquée par mail. Vous devez rendre au plus tard en fin de séance :

- un rapport unique, contenant vos réponses aux questions suivantes, et au moins une capture d'écran chaque fois qu'un test est demandé
- les scripts que vous avez écrits, ou à défaut une copie des lignes de commandes que vous aurez entrées sur votre terminal

sous forme d'une archive ZIP unique à déposer via le formulaire suivant, à l'exclusion de tout autre moyen : https://mvx1.esiee.fr/tp_drop.html (vous devez accepter l'exception de sécurité, le certificat SSL a expiré). Indiquez bien les deux logins formant votre binôme, sauf si vous travaillez seul. Pour créer l'archive ZIP : `zip -r archive.zip fichiers...`

Vous devez aller jsuqu'au bout de la procédure, qui se terminera par l'envoi d'un mail intitulé « Remise de TP : accuse de reception » sur votre boîte mail : pas d'accusé de réception, pas de contestation possible.

1. Fichiers de mauvaise extension

Les extensions de fichiers sont parfois trompeuses. Il est en effet parfaitement possible qu'un nom de fichier se termine par « .txt » alors qu'il s'agit en fait d'un fichier MP4, par exemple.
La commande `file fichiers ...` tente d'identifier la nature du ou des fichiers qu'elle reçoit en argument d'après leur signature d'entête, ou leur contenu.

Travail demandé :

1) Le répertoire courant contient des fichiers dont certains peuvent avoir l'extension « .jpeg » ou « .jpg » , mais dont on soupçonne qu'ils ne sont pas au format JPEG. Ecrire un script, ou donner directement du code qui permet de les lister. Testez le résultat sur les fichiers que vous trouverez dans `/nfs/opt/bdr/3R-IN3`

2) Complétez le script ou les commandes précédentes pour que la liste des fichiers trouvés sorte dans un fichier `trouves` plutôt qu'à l'écran

3) Créez un répertoire `mauvais .` Puis, en réutilisant le fichier `trouves` , et seulement lui, placez une copie de chaque mauvais fichier de `/nfs/opt/bdr/3R-IN3` dans le répertoire mauvais.

2. Annuaire et login

Un fichier texte vous est livré, qui contient des informations sur des personnes organisées selon le format suivant : Prénom NOM[-NOM]|Localisation|Telephone|Email

Un exemple vous est donné dans `/nfs/opt/bdr/3R-IN3/annuaire`

On voudrait générer automatiquement des propositions de logins pour chacun de ces utilisateurs, en utilisant les 8 premières lettres au plus de son nom, toutes converties en minuscules. Par exemple, pour l'extrait suivant :

```
Xavier HILAIRE|Bureau 5352c|926708|x.hilaire@esiee.fr
Laurence LARAUD-MELVEL|Site de Cergy||laurence.laraud@esiee.fr
Claude DEBLANGY|||cdeblang@sfr.fr
```

on devrait obtenir :

```
hilaire
laraudme
deblangy
```

Proposer un script, ou directement des commandes, qui permettent de traiter un tel fichier et produisent le résultat voulu.
Aide : `tr 'A-Z' 'a-z'` remplace les caractères majuscules trouvés sur l'entrée standard par des minuscules.

1. Un service rlast

La commande `last` permet d'afficher l'historique des derniers utilisateurs qui se sont connectés à la machine qui l'exécute. Voici un extrait produit sur une machine de l'ESIEE :

```
hilairex pts/1        90.3.43.7        Fri Jun  4 21:12   still logged in
hilairex pts/0        90.3.43.7        Fri Jun  4 19:23   still logged in
cevaerc  pts/1        147.215.150.51   Fri Jun  4 14:31 - 19:15  (04:44)
cevaerc  pts/0        92.167.153.21    Fri Jun  4 14:05 - 19:15  (05:10)
cevaerc  pts/0        92.167.153.21    Fri Jun  4 12:19 - 12:19  (00:00)
cevaerc  pts/0        92.167.153.21    Fri Jun  4 12:16 - 12:16  (00:00)
cevaerc  pts/4        147.215.150.51   Fri Jun  4 12:15 - 12:15  (00:00)
dupalutb pts/4        147.215.200.10   Fri Jun  4 11:48 - 11:52  (00:03)
cevaerc  pts/1        147.215.150.51   Fri Jun  4 11:43 - 12:15  (00:31)
cevaerc  pts/3        147.215.150.51   Fri Jun  4 11:09 - 12:15  (01:06)
cevaerc  pts/2        147.215.150.51   Fri Jun  4 10:33 - 12:15  (01:41)
pagazanj pts/2        91.173.164.130   Fri Jun  4 10:30 - 10:31  (00:00)
cevaerc  pts/0        92.167.153.21    Fri Jun  4 10:05 - 12:15  (02:10)
cevaerc  pts/2        147.215.150.51   Fri Jun  4 10:03 - 10:05  (00:01)
pintoc   pts/1        147.215.40.31    Fri Jun  4 08:26 - 11:22  (02:55)
cevaerc  pts/0        92.167.153.21    Fri Jun  4 07:43 - 10:05  (02:21)
reboot   system boot  4.9.0-15-amd64   Fri Jun  4 00:02   still running
```

On voudrait proposer un service de consultation à distance de `last`, c'est-à-dire un service qui renvoie les informations produites par last sans pour autant devoir se connecter à la machine concernée ni même y détenir un compte. Pour limiter les indiscrétions, on recourra à un fichier de mots de passe dont chaque ligne comporte un mot de passe, suivi d'un espace au moins, puis d'une liste d'utilisateurs séparés par des virgules. En voici un exemple :

```
ESIEE2022!!     hilairex
+sudoko+        ALL
WeakPasswd      dupalutb,cevaerc,pintoc
```

Ce fichier autorise le serveur à renvoyer les informations de connexion concernant l'utilisateur hilairex à condition que le mot de passe « ESIEE2022!! » lui soit fourni ; celles concernant dupalutb, cevaerc, et pintoc s'il s'agit de « WeakPasswd » ; et TOUTES les connexions s'il s'agit de « +sudoko+ ».

Travail demandé :

1) Ecrivez un script `serveur1.sh` qui implémente ce service. Il prendra en unique argument un fichier de mots de passe que vous créerez, puis devra effectuer en boucle infinie les opérations suivantes :
- lecture du mot de passe envoyé par le client sur une ligne
- si ce mot de passe est invalide, le serveur doit répondre au client par une ligne FAILED
- s'il est valide, alors il devra lister les connexions concernant les utilisateurs appropriés, puis terminer par une ligne OK

Testez votre serveur directement sur le terminal. Vous pourrez simuler ce que doit faire le client en entrant le mot de passe directement, et n'avez pas besoin d'une mise en réseau pour le moment.

2) Mettez votre script précédent en service réseau, sur le port TCP de votre choix. Vous avez bien entendu le droit de réutiliser ce qui a été fait lors du dernier TP, et le script précédent pourra être appelé à partir d'un script `serveur.sh`
Refaites un test en simulant ce que devra faire le client via netcat là encore.

3) Ecrivez le client sous la forme d'un script `client.sh` , acceptant en arguments le nom de la machine à interroger, et le mot de passe à utiliser. Ce script pourra lui-même appeler un autre script `client1.sh`. Le client ne devra afficher que les lignes de connexions de last renvoyées par le serveur s'il en renvoie, mais son code de sortie devra refléter le résultat de sa demande : 0 si le serveur a renvoyé OK, 1 s'il a renvoyé FAILED.