Travaux Pratiques n°2

Etudes et configuration des clients et serveurs pour le transfert de fichiers

But du TP

Ce TP veut aborder la transmission de fichiers informatiques à travers le réseau. Les protocoles tels que `smb`, `nfs` de « montage » dans le système de fichiers local sont vus ailleurs et ne seront donc pas abordés ici au profit de `ftp`, `sftp`, `scp`, `tftp`, `wget`, `curl`, `rsync`, `tftp`…

Mise en place

Cette manipulation peut être effectuée sur chacun des postes des salles 410, 405 voire 102 sans notion de poste client et poste serveur. Tous les PC peuvent être à la fois clients et serveurs. Comme vous devez être administrateur sur le système, vous devrez travailler sur une machine virtuelle. Ce TP suppose l'utilisation de la distribution Debian/v11.02 telle qu'elle est installée sur le disque virtuel Debian1102-20220323-pm...`vdi`. Il faut donc créer une machine virtuelle comme indiqué en adaptant le nom de votre vm.

```
createvm
la commande sans paramètre vous donne la syntaxe à utiliser.
createvm Debian1102-20220323-pm….vdi ma_vm
```

Ne pas changer l'adresse MAC, cela empêcherait la VM d'obtenir une adresse IP prévisible. On suppose aussi qu'un câble droit relie directement l'interface `en1` de votre PC au switch de la salle (SW102, SW410 ou SW405) : voir la figure n°1.

Paillasse 1

PC Gauche n° 1			SW405		RT4331 haut	SW2960 haut	SW2960 bas	SW3560	RT2811	RT4331 bas		SW405	PC Droit n° 2		
COM	EN 1	EN 0	Baie A1	Baie B1	COM	COM	COM	COM	COM	COM	Baie B2	Baie A2	EN 0	EN 1	COM
1	Câble fixé	3	4	5	6	7	8	9	10	11	12	13	14	Câble fixé	16

Illustration 1: Bandeau de la paillasse 1, salle 405.

Faites démarrer votre VM et logez-vous en root ou en rt. Dans tout le TP il va être question d'adresses IP et de noms de machines. Les exemples utilisent soit la salle 405

soit la 410, soit la 102 : il convient de tout ramener à la salle dans laquelle vous effectuez la manipulation.

I. File Transfer Protocol (FTP)

La plupart du temps les navigateurs `http` ont des extensions qui permettent de manipuler de manières intuitives le transfert de fichiers avec le protocole `ftp`. Il existe aussi de nombreux utilitaires graphiques qui s'interfacent avec `ftp`. Ici notre propos est de manipuler avec *un client en ligne de commande* pour se retrouver dans la situation de l'administrateur d'un serveur.

1. Le ftp côté serveur

a) Démarrage

Nous installons le serveur `proftpd` sur votre machine virtuelle.

```
# apt-get update ; apt-get install proftpd-basic
```

Si vous avez des questions lors de l'installation, laisser les options par défaut, un mode de fonctionnement indépendant, pas `inetd`, (donc *standalone*), ce sera plus clair pour le TP.

Vérifiez que dans le fichier `/etc/proftpd/proftpd.conf` le ServerType est bien `standalone`.

Modifier le nom du serveur, ce sera plus aisé de distinguer les serveurs quand vous utiliserez le serveur d'un autre binôme.

Un commentaire sur `inetd` : il s'agit d'un « super-serveur ». Normalement les serveurs sont des processus qui tournent sans attache à un terminal, ce que l'on appelle usuellement un « démon » en Unix et qui sont démarrés en même temps que la machine. C'est le cas des services web (apache), samba (partage de fichiers Windows) ou `dhcp` et bien d'autres. Par contre, certains services sont utiles, pratiques sur les serveurs mais utilisés au coup par coup, somme toute, rarement. Plutôt que de les lancer systématiquement au démarrage, ils ne sont démarrés « qu'à la demande » par le processus `inetd` qui, lui, écoute sur les ports configurés. C'est ce que nous verrons avec le serveur `tftp` en fin de manipulation.

Côté client (sur l'autre pc ou sur un autre terminal), utilisez la commande `ftp` pour vous connecter au serveur.

```
# ftp adresseIP_de_votre_machine_serveur
```

Choisissez l'utilisateur `rt` et faites des `cd` pour visiter la machine serveur. Normalement c'est possible. Il est peut-être inutile d'offrir ces fonctionnalités à tout le monde : il va falloir les limiter.

Testez les commandes « `?` », `status`, `ascii`, `binary`… (*que font ces commandes* ?)

Sans rien changer, déconnectez-vous et reconnectez-vous sur le compte `root`.

Si vous y arrivez c'est que vous êtes un pirate ! `Root` comme d'autres utilisateurs privilégiés ne peut pas se loger par `ftp`, `car` son nom de *login* est présent dans le fichier `/etc/ftpusers`.

b) Les ports : mode normal

Ftp n'utilise pas toujours les mêmes ports tcp.

Faisons un essai classique : sans le mode passif.

- Logez-vous sur **un client** autre que la machine serveur, par exemple sur votre autre PC de la paillasse ou sur une seconde VM.
  - Attention le client `ftp` n'est pas installé sur la machine physique. Sinon allez en `rt/rt` sur `centrex`.
- Sur le serveur, lancer une acquisition `wireshark` sur l'interface `en1`.
- Avec le client « `ftp` » connectez-vous sur votre serveur.
  - En faisant plusieurs fois la commande « `passive` », assurez-vous d'être dans le mode actif.
  - Faire la suite de commandes :

    ```
    cd /etc
    get passwd
    bye
    ```

  - qu'avez-vous fait ? Le fichier est en texte : quel mode avez-vous utilisé ? Quel est l'autre ? Aurait-ce été mieux ou moins bien ?

Arrêtez `wireshark`.

- Dans `wireshark` limitez l'affichage en mettant un filtre du genre « `ip.addr==addrIpClient and ip.addr==addrIpServeur` ».
- Dans le menu « `Statistiques` » choisir l'entrée « `Graphique des flux` »
  - Dans les différents onglets de la nouvelle fenêtre, limitez l'affichage à
    - Filtre d'affichage
    - Adresses réseaux
    - Flux TCP
- Suivez les différentes étapes du protocole tout en vérifiant les numéros de ports utilisés par le serveur.
  - Combien voit-on de couleurs dans le graphique ? Qu'est que cela représente ?
    - Combien de connexions TCP ont-elles eu lieu ?
    - Qui les ouvrent ?

— Qu'y a-t-il dans les gros paquets en orange[1] du serveur vers le client ?

— Comment est fait le choix du port sur le client ? (Indication : regardez le contenu de certains paquets qui précèdent l'ouverture de la nouvelle connexion TCP).

- Est-ce que les pare-feu vont apprécier ? Pourquoi ?
- Sauvegardez l'acquisition, il sera bon d'y revenir.

c) Les ports : mode passif

Reprenez l'ensemble de ce que vous avez fait au paragraphe ci-dessus (I. 1. b)) mais avant de faire la commande « `get passwd` », passez en mode passif.

- En examinant les résultats sur `wireshark`, en particulier dans la fenêtre de flux, on voit encore deux couleurs : Quelle est la différence ?
  - Qu'est-ce qui est transmis dans la partie orange ?
  - Quels sont les ports mis-en jeu ?
  - Qui ouvre la communication orange ?
  - Comment le numéro du port du serveur est-il déterminé ?

Déduisez l'utilité de cette commande. D'après vous, dans quel cas est-ce particulièrement intéressant ?

d) Limitation des accès

Nous allons modifier le mode d'accès au site en bloquant l'utilisateur dans son répertoire.

Parcourez le fichier `/etc/proftpd/proftpd.conf`. Par contre vous devez modifier ce fichier pour qu'il reflète le fonctionnement souhaité.

Modifier le nom du serveur pour voir ce que ça fait.

Dé-commenter la ligne

```
DefaultRoot                     ~
```

Redémarrer le serveur

```
# systemctl restart proftpd.service
```

Vérifiez que cette fois on ne peut plus s'écarter de son répertoire de `login`.

e) Accès administrateur

C'est bien pour les gens qui ne doivent gérer que leur compte, mais pour les administrateurs c'est très gênant ! On va autoriser les utilisateurs du groupe `staff` à aller où bon leur semble.

Changer la directive `DefaultRoot` du fichier `proftpd.conf` en

1 Les couleurs peuvent être différentes suivant le nombre de connexions TCP de votre acquisition.

```
DefaultRoot                     ~ !staff
```

et redémarrez le service.

Sur le serveur, déclarer un utilisateur `cisco` et le mettre dans le groupe staff.

```
# adduser cisco
```

Répondre aux questions

```
# adduser cisco staff
```

Vérifiez les groupes auxquels appartiennent `cisco` et `rt` :

```
# groups cisco rt
```

Vérifiez maintenant la différence entre les deux `logins` en vous connectant par `ftp`.

f) Un serveur ftp sur un autre port (10021)

Nous allons créer un serveur « virtuel » sur un autre port. Nous avons vu que le serveur utilisait le port 21 (ftp) pour les commandes.

On veut tester un serveur général sur le port 10021 qui soit authentifié avec les comptes utilisateur déjà vus mais sur lequel personne ne peut écrire.

Ajoutez à la fin du fichier `/etc/proftpd/proftpd.conf` quelque chose d'inspiré du fichier `/etc/proftpd/virtuals.conf` (cf. listing 1 page 5).

```
<VirtualHost adresseIP_duServeur>
ServerAdmin             root@localhost
ServerName              "Big FTP Archive"
TransferLog             /var/log/proftpd/nomDuServeurA
RequireValidShell       no
Port                    10021
DefaultRoot             /srv/ftp_aroot
AllowOverwrite          yes
DisplayLogin            /etc/proftpd/welcome.msg
</VirtualHost>
```

Listing 1: Définition d'un serveur ftp sur un autre port.

Après avoir redémarré le serveur, la commande

```
ftp adresseIP_du serveur 10021
```

ne donne rien de bon.

Allez voir dans le fichier `/var/log/proftpd/proftpd.log`. Que ce passe-t-il ? Le corriger et refaire l'expérience. Côté serveur avec un « cp », mettre un fichier quelconque dans le bon répertoire pour voir que c'est vraiment le bon, vu du côté ftp !

Mais tout le monde y a accès.

On limite l'accès à certaines machines clientes, en ajoutant quelques directives .

```
<Limit LOGIN>
  Order allow,deny
  Allow from adresseIP_duClient/28
  Deny from all
</Limit>
```

Listing 2: Limitation d'accès

On va limiter l'accès à votre client et donc interdire au reste du monde. Il faut ajouter le contenu du listing n°2 à la définition déjà faite, c'est-à-dire entre les balises « VirtualHost » du listing 1.

Vérifiez que vous vous connectez toujours du client mais pas d'une autre machine de la salle en dehors des 16 adresses autorisées.

g) Un message

On peut, et souvent on doit, avertir l'usager des risques juridiques qu'il encourt.

La ligne « `DisplayLogin` » du listing n°1 ne correspond à rien, car le fichier désigné n'existe pas :

Copier le fichier `/usr/share/proftpd/templates/welcome.msg` dans le bon répertoire pour qu'il soit vu dans sur le serveur virtuel (port 10021) et le personnaliser un peu.

Vérifiez le fonctionnement.

2. Le ftp côté client

La commande ftp que nous utilisons est un peu simple.

a) Ftp de base

Commencez à vous loger de la machine cliente sur votre serveur

```
ftp adresseIP_duServeur
```

au prompt « ftp> » tapez un « ? » suivi d'« entrée » vous obtenez l'ensemble des commandes disponibles. Il y en a très peu qui s'adresse au répertoire local. (seul lcd existe).

b) `Ncftp`

Nous disposons en ligne de commande de nombreux autres outils, par exemple `ncftp`.

Il est déjà installé. Tapez et expliquez le résultat :

```
ncftp ftp://rt:rt@serveurIP
ncftp / > cd /usr/bin
```

Oh c'est vrai cela ne fonctionne pas ! Que faut-il changer ? Modifier la commande en conséquence et récupérer le fichier `/usr/bin/xzless`.

On teste d'autres commandes : expliquez ce qu'elles font.

```
ncftpls ftp://rt:rt@IpDuServeur:10021/
ncftpget -u rt -R IpDuServeur   /tmp/rt/ .
```

II. SCP

L'usage de `scp` suppose que l'on dispose d'un serveur `ssh`. En effet `scp` est une manière extrêmement simple d'utiliser un tunnel ssh pour copier des fichiers.

Nous n'aborderons pas ici l'installation et la configuration d'un serveur `ssh`, car vous le verrez plus tard. Nous supposerons seulement que le serveur `ssh` a été installé correctement. Nous n'étudierons d'ailleurs que le côté client.

Le serveur `ssh` est installé par défaut sur chacune des machines virtuelles. Nous allons copier un fichier en libre accès qui existe sur toutes les machines (`/etc/passwd`).

Sur la machine cliente

```
scp /etc/passwd rt@adresseIP_serveurSSH:
```

Cela copie le fichier `passwd` sur le serveur sur le compte de `rt`, dans son répertoire de `login`. Il vous est demandé le mot de passe de `rt`.

On peut aussi le faire dans l'autre sens avec la même facilité.

```
scp rt@adresseIP_serveurSSH:/etc/passwd ./
```

Cela utilise le compte `rt` sur le serveur pour copier le fichier `/etc/passwd` dans votre répertoire courant.

La grande force de cette commande c'est qu'elle travaille sur le réseau de manière cryptée, n'est pas limitée à des comptes standards, *a priori* `root` peut aussi l'utiliser, et tous les modes d'authentification de `ssh` sont possibles.

Les deux options à mon avis les plus intéressantes sont « -p » et « -r ».

Que changent-elles dans le fonctionnement ? Vérifions !

```
# scp -pr /var/spool rt@adresseIP_duServeurSSH:
```

La copie s'effectue de manière hiérarchique mais certains fichiers ne sont pas copiés ! Lesquels ?

Vérifiez les dates des fichiers copiés. Cela correspond-il ?

III. SFTP

Comme son nom l'indique, nous avons affaire à un ftp sécurisé.

En pratique c'est un module particulier du serveur ssh qu'il faut activer (il l'est par défaut) dans le fichier /etc/ssh/sshd_config : voir la ligne

```
subsystem sftp /usr/lib/openssh/sftp-server
```

On peut alors utiliser sur un serveur ssh, des commandes proches de celles du client ftp.

```
sftp rt@addresseIP_duServeurSSH
```

On a été identifié comme « rt ». Le « ? » permet de voir les commandes disponibles. Expérimentons un peu !

```
sftp> cd /etc
sftp> get passwd
sftp> lls -l
```

On voit que l'on a récupéré le fichier dans le répertoire courant (/root).

Essayons dans l'autre sens, pour le déposer dans le répertoire de login de rt :

```
sftp> put /etc/passwd
```

Il semble qu'il y ait un problème ! Lequel ?

```
sftp> put /etc/passwd ~/
```

Est-ce mieux ? Pourquoi ?

```
sftp> put /etc/passwd /home/rt/
```

Pourquoi cela résout-il le problème ?

IV. Wget et curl

Ces deux logiciels servent à transférer le contenu d'url *(Universal Resource Locator)*. L'usage de wget et de curl est très proche. Nous allons plus nous pencher sur l'usage de curl.

1. Installation

Ce logiciel n'est pas installé sur votre machine. Nous allons commencer par cela.

```
# apt-get update
```

Si vous avez des erreurs ne pas hésiter à relancer la commande plusieurs fois.

```
# apt-get install curl
```

2. Récupération d'un fichier

Pour récupérer un fichier, il faut savoir où il est, quel protocole peut me donner le meilleur accès. Fréquemment il n'est pas nécessaire de s'authentifier auprès d'un serveur pour accéder au fichier. C'est le principe du serveur anonyme. Il y a un serveur ftp anonyme sur la machine centre, nous allons l'utiliser.

Il y a plusieurs méthodes pour récupérer un fichier par ftp ou http. Nous allons tester la récupération d'un gros fichier interrompu, au moyen du protocole http.

```
# curl -O http://centre.gtr.tp/Medias/Sintel.2010.720p.mkv
```

Interrompre le transfert et le reprendre avec la commande modifiée

```
# curl -C - -O  http://centre.gtr.tp/Medias/Sintel.2010.720p.mkv
```

Vous voyez qu'il reprend la transmission là où il en était. Vous pouvez la ré-interrompre et la reprendre plusieurs fois. C'est pratique !

On peut faire mieux : récupérer tous les fichiers mkv du site.

Avec wget :

```
# wget ftp://ftp@centre.gtr.tp/*.mkv
```

l'interrompre et reprendre…

Avec ncftp :

```
# ncftpget centre.gtr.tp ./ /*.mkv
```

On voit que ces deux logiciels reprennent le transfert même avec plusieurs fichiers.

Pour ftp le meilleur, c'est ncftp, il peut aussi transmettre des hiérarchies de répertoires. Pour http, wget est très simple d'emploi.

V. RSYNC

Nous ne ferons donc ici qu'une expérimentation élémentaire des possibilités de rsync.

Il faut commencer par installer rsync sur vos deux machines au moyen d'un apt-get.

Ensuite effectuons les commandes suivantes sur vos deux machines :

```
# mkdir /var/back-log
# rsync -va/var/log root@IpMachineDistante:/var/back-log/
```

Normalement ça ne fonctionne pas : en effet rsync ici utilise ssh et l'utilisateur root ne peut se loger par ssh avec un mot de passe. Nous allons l'autoriser. Cela introduit un défaut de sécurité qui ici n'est pas grave, mais nous verrons au TP n°5 une méthode plus sécurisée.

Sur la machine vers laquelle vous copiez (la machine distante de la commande précédente) dans le fichier /etc/ssh/sshd_config, ajoutez la ligne

```
PermitRootLogin yes
```

puis redémarrez le serveur ssh :

```
service ssh restart
```

Ré-exécutez la commande précédente

```
# rsync -va /var/log root@IpMachineDistante:/var/back-log/
```

Cette fois-ci elle accepte le mot de passe. Expliquez ce que l'on voit et ce qui se passe et analysez les différentes options de la commande.

Rappeler la commande de synchronisation :

```
# rsync -va /var/log root@IpMachineDistante:/var/back-log/
```

Expliquez ce que l'on voit (ou ne voit pas) et ce qui se passe.

On modifie très légèrement la commande de synchronisation :

```
# rsync -va /var/log/ root@IpMachineDistante:/var/back-log/
```

Expliquez ce que l'on voit et ce qui se passe

Rappeler la commande de synchronisation :

```
# rsync -va /var/log/ root@IpMachineDistante:/var/back-log/
```

Expliquez ce que l'on voit (ou ne voit pas) et ce qui se passe

On crée un événement

```
# su rt
```

Au prompt de `rt`, vous délogez aussitôt, et rappeler la commande de synchronisation :

```
# rsync -va /var/log/ root@IpMachineDistante:/var/back-log/
```

Qu'observe-t-on et pourquoi ?

On ajoute l'option `--delete` à la commande. Tester et expliquez !

VI. `Tftp`

Le protocole `tftp` n'est pas un protocole sécurisé du tout car il ne permet pas d'authentifier le client. Il sert pourtant beaucoup aux machines simples pour récupérer sur le réseau leur noyau, leur configuration… C'est ce protocole qu'utilisent les *switches* les routeurs, les téléphones IP… Malgré son nom il n'est pas proche du protocole ftp. Il utilise le transport UDP.

Nous allons utiliser le logiciel `atftp` qui est un peu plus avancé que les autres. (*Advanced Trivial File Transfer Protocol*). On pourrait n'installer le client `tftp` que sur la VM cliente et le serveur `tftpd` que sur la VM serveur.

Par souci de simplicité, effectuez l'installation des deux paquets serveur et client sur les deux machines virtuelles :

```
# apt-get install atftpd atftp
```

Reconfigurez le serveur pour qu'il fonctionne avec `inetd`.

```
# dpkg-reconfigure atftpd
```

Le laisser démarrer par inetd, et valider des options par défaut sauf la « multi-diffusion » où vous mettrez « non ». Ne pas oublier de redémarrer le démon inetd.

```
# service openbsd-inetd restart
```

Vous avez vu donc que le serveur utilisera le répertoire /srv/tftp comme racine.

On peut avoir deux modes au niveau du client : un mode interactif et un mode ligne de commande. Si la syntaxe n'est pas la même, on peut choisir exactement les mêmes options. J'indique dans la suite les commandes, vous pouvez bien-sûr utiliser la méthode interactive.

On récupère un fichier binaire de centre :

```
# atftp -g -r 2950/c2950-i6k2l2q4-mz.121-22.EA11.bin centre
```

Normalement il fait une erreur d'écriture : atftp a la même faiblesse que le client simple ftp, il récupère le même chemin en distant et en local. En clair, l'erreur vient de ce que le répertoire 2950 n'existe pas en local. On vérifie :

```
# mkdir 2950
# atftp -g -r 2950/c2950-i6k2l2q4-mz.121-22.EA11.bin centre
```

cette fois cela a dû fonctionner.

Envoyer au moyen de tftp, le fichier obtenu (c2950-i6k2l2q4-mz.121-22.EA11.bin) sur la machine distante de votre paillasse, c'est-à-dire l'autre PC de votre paillasse où vous avez installé atftpd.

Dans quel répertoire, sur la machine distante, le fichier a-t-il été copié ?

Tftp peut être très pratique mais il a une importante limitation.

Copier le film Sintel de centre sur votre serveur tftp.

```
serveur# cd /srv/tftp/
serveur# curl -C - -O  http://centre.gtr.tp/Medias/Sintel.2010.720p.mkv
```

Maintenant essayez de copier par tftp le fichier Sintel.2010.720p.mkv sur votre machine cliente tftp à partir de la machine cliente.

Il y a quelque temps, vous auriez dû avoir une erreur. TFTP a été amélioré et s'adapte à la taille des fichiers qu'il doit transmettre. Difficilement, mais il y arrive. Par contre vous avez dû noter le temps très important qu'a pris la transmission, par rapport à la précédente par scp.

Le problème est simple : les paquets sont numérotés et transmis l'un après l'autre. Le champ où est inscrit le numéro est de taille 16 bits. Combien aurons-nous de

paquets au maximum ? La taille utile transmise par défaut en `tftp` est de 512 octets. Quelle est alors la taille maximale du fichier transmis ? On peut augmenter la taille du bloc jusqu'à 1428 pour Ethernet. Quelle serait alors la taille maximale du fichier ? Lors de l'installation d'`atftpd` vous avez validé des « options », la variation de la taille et le « timeout » font partie de ces options qui permettent la transmission de grands fichiers. La durée de la transmission (27 minutes dans mon test) montre que ce protocole, n'est efficace que pour de petits fichiers.

Fichier *TP-2_fileTransfert-2223.odt* imprimé le 22 mars 2023