Travaux pratiques

1. Configuration TCP/IP de Linux

But : configurer un agrégat de cartes réseau sur une distribution de type Red Hat.

1. Il est possible, sous Linux, de créer des agrégats de cartes réseau. En cas de défaillance d'une carte, une autre peut prendre le relais, ou encore n cartes peuvent fonctionner de manière simultanée. Les cartes se nomment bond0, bond1, etc.
   Créez un fichier ifcfg-bond0 dans /etc/sysconfig/network-scripts. Remplissez-le comme tout autre fichier de configuration réseau :

   ```
   DEVICE=bond0
   ONBOOT=yes
   NETMASK=255.255.255.0
   IPADDR=192.168.1.25
   BOOTPROTO=static
   ```

2. L'agrégat, ou bonding, nécessite le chargement du module « bonding ». Les paramètres de ce module déterminent le mode de fonctionnement de l'agrégat. L'option « mode » du module prend la valeur 1 pour un backup : si la première carte cesse de fonctionner, la seconde prend le relais, et vice versa. Modifiez le fichier de configuration /etc/modprobe.conf :

   ```
   alias bond0 bonding
   options bond0 mode=1
   ```

3. Les deux cartes eth0 et eth1 vont faire partie de l'agrégat. Modifiez les fichiers de configuration de chacune des cartes en conséquence. Par exemple, pour la carte eth0, le fichier ifcfg-eth0 :

   ```
   DEVICE=eth2
   HWADDR=00:1B:2C:3D:4E:5F
   ONBOOT=yes
   TYPE=Ethernet
   MASTER=bond0
   SLAVE=yes
   ```

4. Activez l'agrégat avec la commande ifup :

   ```
   # ifup bond0
   ```

5. Vérifiez l'état du bonding :

   ```
   # cat /proc/net/bonding/bond0
   Bonding Mode: fault-tolerance (active-backup)
   Primary Slave: None
   Currently Active Slave: eth0
   ...

   Slave Interface: eth0
   MII Status: up
   Link Failure Count: 0
   Permanent HW addr: xx:xx:xx:xx:xx:xx

   Slave Interface: eth1
   MII Status: up
   Link Failure Count: 0
   ```

```
Permanent HW addr: xx:xx:xx:xx:xx:xx
```

2. Quelques commandes réseau

But : manipuler les commandes ping et traceroute pour déterminer le fonctionnement du réseau.

1. Émettez un seul ping vers l'adresse de votre choix, par exemple linuxfr.org, et vérifiez le code retour de la commande.

```
# ping -c 1 linuxfr.org
PING linuxfr.org (88.191.250.104) 56(84) bytes of data.
64 bytes from linuxfr4 (88.191.250.104): icmp_seq=1 ttl=59
time=6.43 ms

--- linuxfr.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 6.439/6.439/6.439/0.000 ms
# echo $?
0
```

2. Le code retour 0 indique que la machine distante a répondu. Écrivez une ligne de commande permettant de tester si un serveur répond, et si non de sortir du programme avec un message d'erreur :

```
# ping -c 1 linuxfr.org >/dev/null 2>&1 || { echo "Le serveur ne
répond pas"; exit 1; }
```

3. Deux serveurs de données sont situés sur des sites différents. Un serveur quelconque, où qu'il soit, veut déterminer le serveur le plus proche pour limiter le trafic réseau et le nombre de passerelles à passer. Il va pour cela utiliser traceroute et les options `-q 1` (une requête par saut) et `-n` (affichage des « hops » au format simple).

```
# traceroute -q 1 -n server1
1  10.x.y.247  0.619 ms
2  10.x.y.2  0.956 ms
3  10.x.y.8  1.028 ms
4  172.x.y.101  1.979 ms
```

4. Pour déterminer le plus court trajet, il suffit de compter le nombre de sauts :

```
S1=$(traceroute -q 1 -n $SERVER1 2>/dev/null | wc -l)
S2=$(traceroute -q 1 -n $SERVER2 2>/dev/null | wc -l)
```

5. Comparez les deux valeurs et déterminez le serveur le plus proche à placer dans la variable SERVER. C'est donc le serveur dans cette variable qu'il faudra utiliser pour optimiser les accès.

```
[ $SERVER1 -ge $SERVER2 ] && SERVER=$SERVER2 || SERVER=$SERVER1
```

3. Le resolver

But : gérer la résolution de nom.

1. Votre machine fait partie d'un grand réseau disposant de deux serveurs de noms DNS et de plusieurs domaines et sous-domaines déclarés. Vous faites partie du domaine subnet1.slynet.org. Les IPs des serveurs DNS sont 10.17.23.127 et 10.17.23.247. Construisez le fichier /etc/resolv.conf :

```
domain subnet1.slynet.org
nameserver 10.17.23.127
nameserver 10.17.23.247
```

2. Votre machine se nomme s14p23 (salle 14 poste 23). Elle doit pouvoir accéder via leur nom court (hostname) à toutes les machines, y compris celles des autres sous-domaines. Par exemple s19p01 est dans le sous-domaine subnet2.slynet.org. Modifiez /etc/resolv.conf en conséquence :

```
domain subnet1.slynet.org
search subnet1.slynet.org subnet2.slynet.org
nameserver 10.17.23.127
nameserver 10.17.23.247
```

3. Certains services récupèrent le nom de votre machine via une résolution locale passant par /etc/hosts. Votre IP est 10.17.35.168. Or sous Linux le nom d'hôte est souvent associé au localhost 127.0.0.1. Rectifiez ceci dans /etc/hosts :

```
127.0.0.1        localhost
10.17.35.168        s14p23 s14p23.subnet1.slynet.org
```

4. La résolution locale passe d'abord par /etc/hosts puis par le DNS. Mais le DNS est généralement plus à jour que /etc/hosts. Modifiez le fichier /etc/nsswitch en conséquence : modifiez l'ordre de la ligne hosts :

```
hosts : dns files
```

4. Services réseau

But : contrôler divers services réseau.

1. Vous voulez savoir à quel service est associé le port 137. Cherchez l'information dans le fichier /etc/services. Ce port est utilisé pour la résolution de noms Netbios (Microsoft, Samba).

```
# grep "137/" /etc/services
netbios-ns 137/tcp       #NETBIOS Name Service
netbios-ns 137/udp
```

2. Toutes distributions confondues, désactivez, s'il est présent, le service xinetd telnet. Rendez-vous dans /etc/xinetd.d, ouvrez le fichier telnet et modifiez la ligne suivante :

```
disable = yes
```

Relancez xinetd pour relire la configuration.

3. Le service sshd autorise la connexion de root. Vous voulez l'éviter. Modifiez la configuration de sshd en conséquence. Le fichier de configuration est /etc/ssh/sshd_config.
Modifiez le paramètre suivant :

```
PermitRootLogin no
```

Ou commentez la ligne tout simplement, et relancez sshd.

4. Votre machine dispose de divers programmes démarrés qui établissent des connexions réseau. L'une d'elles, vous ne savez pas laquelle, est connectée sur le port distant 1863. Comment trouver le processus associé ?
Une méthode consiste à utiliser netstat pour lister les connexions en cours et les processus associés :

```
# netstat -A inet -p -n | grep 1863
```

```
tcp     0      0 192.x.y.z    207.46.27.19:1863     ESTABLISHED
10505/kopete
```

Une autre solution, qui n'a pas été abordée dans le chapitre, est l'utilisation de lsof avec le paramètre `-i`. Puisque tout est fichier, les connexions réseau le sont aussi :

```
# lsof -i :1863
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
kopete  10505  seb   15u  IPv4 132412  0t0  TCP slyserver:33097
...
```

5. Le fonctionnement par défaut du client DHCP est d'écraser le fichier /etc/resolv.conf à chaque renouvellement d'adresse. Le paramètre `-R` permet d'éviter ceci. Selon la distribution, vous pouvez empêcher ceci.
Par exemple sous openSUSE, le fichier /etc/sysconfig/network/dhcp accepte un paramètre DHCLIENT_MODIFY_RESOLV_CONF="yes" ou "no" pour régler par défaut ce genre de cas.
Recherchez dans l'aide de votre distribution comment modifier ce type de paramètre.

6. L'astuce suivante vous explique comment envoyer un courrier électronique depuis une connexion telnet vers un serveur SMTP. Ouvrez une connexion telnet et suivez le mode d'emploi suivant :

```
# telnet smtp.domaine.net 25
HELO salut
MAIL FROM:joe.dalton@lucky.net
RCPT TO: sheriff@daisytown.net
DATA
subject:Tu ne m'auras jamais

Sheriff tu n'auras jamais les Dalton.
.
QUIT
```

5. Partages de fichiers

But : partager des fichiers et des répertoires via NFS et Samba. Les services NFS et Samba doivent être installés et démarrés.

1. Sur votre machine 192.168.1.25 créez un répertoire /DONNEES que vous allez partager tant via NFS que Samba. Pour NFS, modifiez le fichier /etc/exports afin d'exporter son contenu pour tous les membres du sous-réseau 192.168.1.0/24 en lecture seule.

```
# mkdir /DONNEES
# vi /etc/exports
```

Rajoutez la ligne suivante :

```
/DONNEES        192.168.1.0/255.255.255.0(ro)
```

2. Exportez votre nouveau partage. Vérifiez que le partage est bien actif.
Exécutez la commande exportfs :

```
# exportfs -av
```

Vérifiez si le partage est présent :

```
# showmount -e localhost
```

3. Sur une autre machine du sous-réseau, montez ce partage NFS dans le répertoire /DONNEES local et rendez permanente cette modification :

```
# mount -t nfs 192.168.1.25:/DONNEES /DONNEES
```

Ajoutez une ligne dans /etc/fstab ainsi :

```
192.168.1.25:/DONNEES/DONNEESnfsdefaults00
```

4. Exportez /DONNEES avec Samba, en lecture seule, uniquement pour l'utilisateur joe, sous le nom DONNEES. Le partage est caché. Modifiez le fichier /etc/smb/smb.conf ainsi :

```
[DONNEES]
comment = Dossier partagé DONNEES
path = /DONNEES
browseable = no
public = no
writable = no
printable = no
valid users = joe
```

Testez la syntaxe :

```
# testparm
```

Et relancez samba.

5. Créez l'utilisateur Samba joe associé :

```
# smbpasswd -a joe
```

6. Testez le partage avec smbclient :

```
# smbclient //192.168.1.25/DONNEES -U joe
```