



LINUX

Prise en main et aspects Réseaux

Outils d'analyse réseau :

nmap

1 Exploration du réseau

Nmap est un scanner de port. Les options les plus importantes sont:

- p <liste de ports> : scanne une liste de ports
- sL : lister les machines
- sO : detecteter les protocoles IP supportés
- sP : tester la disponiblité des machines
- sV : detecteter les services.

exemples:

Ports ouverts sur votre machine : nmap localhost

Sur une autre : nmap 192.168.0.5

Recherche d'une machine active sur le réseau

nmap -sP www.univ-orleans.fr

nmap -sP www.ibm.com

nmap -P0 www.ibm.com

nmap -sP 192.168.0.0/24

trouver les logiciels associé aux services :

nmap -sV localhost

nmap -sV www.tf1.fr

nmap -sV -P0 -p 25 smtp.neuf.fr

nmap -O www.openbsd.org

Outils d'analyse matérielle :

**dmidecode - lspci - hdparm - lshw - smart –
hddtemp - /proc**

(Installer kernel-utils si non installé)

1 Configuration matérielle

1.1 DMI (Desktop Management Interface)

DMI est un standard décrivant les ressources d'une machine (intégré au bios): il y a actuellement 39 catégories d'informations.

À l'aide de dmidecode (**dmidecode | more**) répondre aux questions:

Modèle et marque du système et de la carte mère ?

Combien de barrettes mémoires sont installées ?

Combien de slots PCI sont utilisés ?

Version du BIOS ?

1.2 Bus PCI

Un bus PCI primaire (00) est généralement équipé de ponts (bridges).

À l'aide des commandes lspci et lspci -tv (ou lspci -vv pour des informations détaillées)

établir l'arborescence PCI de la machine.

Combien de slots sont utilisés ?

verifier les informations en visualisant pci.ids

cat /usr/share/misc/pci.ids

1.3 Disques durs

Les disques ide sont capables de fournir un certain nombre d'informations.

A l'aide de hdparm -i /dev/hda

Marque, modele, N° de série ?

Performances maximales du disque ?

hdparm -t /dev/hda

1.4 Lshw

Cette commande permet de faire un inventaire complet. Il peut fournir les résultat sous forme xml ou html:

lshw

lshw -html > toto.htm

/usr/sbin/lshw-gtk

1.5 SMART

La technologie SMART (Self-Monitoring Analysis and Reporting Technology) permet d'analyser les disques durs

Installer le paquet smartmontools

smartctl -a /dev/hda

combien d'heures de fonctionnement ?

nombre de mise sous tensions ?

nombre d'erreurs répertoriées ?

http://lea-linux.org/cached/index/Hardware-ha4rd_plus-smart.html

Une liste plus exhaustive des attributs est disponible sur ce site :

<http://smartlinux.sourceforge.net/smart/attributes.php>

1.6 HDDTEMP

HDDTEMP permet de connaître la température actuelle des disques durs. Il

fonctionne comme un serveur à l'écoute sur le port 7634.

installer hddtemp. IL faut également mettre à jour la base de données des disques reconnus:

wget <http://www.guzu.net/linux/hddtemp.db> -O /etc/hddtemp.db

On lance ensuite le serveur: **hddtemp -d /dev/hda**

et enfin on interroge avec un client (netcat) sur le port 7634: **nc 127.0.0.1 7634**

Interroger une machine distante sur lequel hddtemp est actif.

2. Pseudo fichiers

Pour cette partie vous aurez besoin des commandes de visualisation **cat, less, more**

..

/proc

Le répertoire /proc contient des « pseudo-fichiers » qui représentent l'état du noyau.

Ces fichiers sont chargés en mémoire et ne sont pas sur le disque dur.

Ouvrir un terminal et allez dans le répertoire /proc (**cd /proc**)

avec la commande ls -l visualiser le contenu du répertoire.

Certain sous-répertoires ont un nom explicite comme cpuinfo; d'autres sont nommés par un nombre : il s'agit du PID des processus en cours et le répertoire contient des informations sur ce processus.

- à l'aide de **ps -ef | grep terminal** déterminez le PID de votre terminal.
- Placer vous dans son répertoire : quel est le contenu ?

- Que contient le fichier cmdline ((**cat ./cmdline; echo**) | **tr '000' '\n'**) ?
(rep : les arguments de la ligne de commande qui a permis de lancer le processus.)

- Que contient le fichier environ ((**cat ./environ; echo**) | **tr '000' '\n'**) ?

- exe ?

(rep : lien symbolique vers le binaire exécuté.)

- statm ?

(rep : état de la mémoire utilisée par le processus.)

- status ?

(rep : n° utilisateur, n° du groupe, n° du processus parent, etat courant: sleeping ou Running.)

Se repositionner dans /proc : analyser le contenu de driver, ide, scsi, net, cpuinfo, mounts, partitions, informant, modules.

Le répertoire proc/sys permet à l'utilisateur d'agir sur la configuration du noyau :
que fait la commande suivante: **echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all**
?

verifier en realisant un **ping** sur la machine avant et après ...

Pour voir tous les paramètres configurables du répertoire /proc/sys vous pouvez lancer **/sbin/sysctl -a**

que signifie les lignes suivantes

dev.hpet.max-user-freq = 64

(HPET= High Precision Event Timer ->

<http://www.intel.com/hardwaredesign/hpetspec.htm>)

dev.rtc.max-user-freq = 64

(rep :fixe une limite à la frequence des interruptions pour les processus utilisateurs.)

Quelle information fournis **cat /proc/sys/vm/swappiness** ?

(rep :le pourcentage de données qui sont mises en cache: par défaut 60%)

Quelques astuces pour optimiser son système :

http://lea-linux.org/cached/index/Admin-admin_plus-optimise.html

Ouvrage en francais (1ère édition épuisée) :

/Proc et /SYS 2ème édition [Daudel](#) (28 avril 2006)



Connexion à distance :

vino - x2vnc - ssh

1. Administration à distance

1.1. VINO

Dans l'environnement GNOME, l'outil de partage de bureau s'appelle Vino (serveur vnc):
Environnement de bureau -> Préférences -> Bureau à distance :
Cocher les options pertinentes...

Faites un essai de bon fonctionnement sur la même machine :

`vncviewer localhost:0`

Pourquoi plusieurs fenêtres s'ouvrent ?

dans un terminal (en root) lancer la commande `startx -- :1.`

Vous avez alors à votre disposition deux « bureaux » accessibles par **Ctrl+Alt+F7** et **Ctrl+Alt+F8**.

Le premier écran a pour identifiant localhost:0 et le deuxième localhost:1.

Pour lancer un processus graphique dans l'une ou l'autre des fenêtres à partir d'une console il faut le préciser dans la ligne de commande ou dans la variable DISPLAY:

```
$export DISPLAY=localhost:1
```

```
$ nautilus&
```

Visualiser votre bureau 1 **Ctrl+Alt+F8**).

Essayer maintenant à partir d'un ordinateur distant:

`vncviewer 192.168.0.X:0` ou `vncviewer 192.168.0.X:1`

En fonction des options choisies sur le serveur (Vino) vous pouvez voir et/ou contrôler le bureau distant.

Attention: dans cette configuration, les échanges ne sont pas sécurisés !

1.2. Couplages d'écrans

On utilise x2vnc

`x2vnc 192.168.0.y:0 -east` (ou west ou north ou south)

Lorsque vous sortez votre pointeur de souris par la droite de l'écran vous contrôlez l'écran distant (souris+clavier).

1.3. Accès sécurisés

Pour sécuriser les échanges on peut utiliser ssh:



- créer vous un compte utilisateur sur la machine distante
 - autoriser le « forwarding » des échanges X11:
 - dans `ssh_config` (de la machine cliente) :
`ForwardX11 yes`
 - dans `sshd_config` (du serveur) :
`X11Forwarding yes`
 - `ssh -X user@192.168.0.X`
 - une fois le terminal distant activé, vous pouvez lancer nautilus ou konqueror.
- Pour le transfert des fichiers : `scp` ou `sftp`.

Plannification synchrone et asynchrone:

At - Cron -

1. Plannification synchrone

1.1. CRON

l'aide Cron realiser un script qui verifie toutes les minutes la temperautre des disques de trois machines distantes et l'enregistre dans un fichier.

Cron est un outil qui permet l'exécution de tâches planifiées et récurrentes (pour les taches à n'executer qu'une fois on utilise at).

Cron est composé de plusieurs entités :

- **crond** : demon verifiant toutes les minutes la présence de nouveaux fichiers de configuration (crontab).

-**fichier de configuration crontab** (dans /var/sppol/cron) : contient les travaux programmés. Chaque utilisateur autorisé à un fichier distinct. le format de chaque ligne dans ce fichier est le suivant:

```
*      *      *      *      *      Command to be executed
-      -      -      -      -
|      |      |      |      |
|      |      |      |      +----- Day of week (0-7)
|      |      |      +----- Month (1 - 12)
|      |      +----- Day of month (1 - 31)
|      +----- Hour (0 - 23)
+----- Min (0 - 59)
```

(* remplace n'importe qu'elle valeur)

exemples:

execution toutes les heures de la commande /bin/lis:

0 * * * * /bin/lis >/dev/null 2>1

executer la commande à minuit 10 tous les jours

10 0 * * * /bin/nightly

le lundi à 2h du matin

0 2 * * 1 /usr/local/bin/monday

à 1h,2h,3h,4h:

*** 1-4 * * * /bin/some-hourly**

-**commande crontab** avec les options suivantes :

-**l** : permet de visualiser le contenu crontab de l'utilisateur courant.

-**e** : permet d'éditer le fichier crontab et éventuellement de le modifier. La commande ouvre le fichier crontab dans un éditeur (vi par défaut !!!)

-**r** : supprime la crontab dans /var/spool/cron

-fichier log : dans /var/log/cron.

Il existe également un fichier crontab du système (/etc/crontab) pour l'exécution des tâches récurrentes système définis par:

```
/etc/cron.d
/etc/cron.daily
/etc/cron.hourly
/etc/cron.monthly
/etc/cron.weekly
```

avant de planifier nos propres taches on change d'éditeur :

export EDITOR=/usr/bin/gedit

on peut maintenant éditer crontab : **crontab -e** et taper ses lignes:

se faire réveiller toutes les minutes:

***/* * * * * (xclock)**

(pour tuer ce processus: **ps** et **kill**)

Pour arrêter définitivement cette horreur **crontab -r**.

On va maintenant verifier toutes les minutes que firefox est actif, et lance le navigateur le cas échéant.

créer le script ci-dessous qui surveille firefox

```
#!/bin/sh
# script surveillance_firefox
/bin/pidof /usr/lib/firefox/firefox-bin > /dev/null
if [ $? = 1 ]
then
firefox
echo "Redémarrage de firefox" | mail -s "[firefox]"
cadorel@debianpresario.cresitt.com
fi
```

et exécuter par crontab.

finalement réaliser un script qui vérifie toutes les 5 minutes la température des disques de trois machines distantes et l'enregistre dans un fichier.

1.2.AT

récupérer une image d'une webcam à une heure donnée:

```
$ at 15:30
at>wget http://invite:invite@82.229.118.71:15100/jpg/image.jpg
at> ^D
$
```

Sauvegardes et synchronisation de contenus:

scp – Rsync - Unison

1 SSH pour sécuriser les échanges

Nous avons déjà utilisé ssh pour se connecter de manière interactive à un PC distant. Ici nous souhaitons réaliser les échanges sans intervention de l'utilisateur : on ne peut donc pas lui demander de s'authentifier de manière interactive !
On va utiliser un système d'échange de clés.

1.1 Authentification par clé publique

Créer votre compte utilisateur (pas root !!!) sur le PC local et distant.
Sur le PC local ouvrez une session sur ce compte.
On va maintenant générer une clé privé et une clé publique avec la commande suivante:

```
ssh-keygen -t dsa
```

Le programme va vous demander une « passphrase »: Pour simplifier ne rien mettre (en production il faudra rentrer cette phrase et on automatisera les connexion grâce à ssh-agent qui gère les clés).

Vous devez obtenir dans le **répertoire .ssh** un fichier **id_dsa** contenant votre clé privé , et **id_dsa.pub** contenant votre clé publique.

Il faut maintenant fournir la clé publique au PC distant : ajoutez le contenu de votre id_dsa.pub au fichier **authorized_keys** du compte sur le pc distant.

Essayer maintenant de vous connecter par ssh. Si vous obtenez toujours la demande de mots de passe, l'authentification par clés ne fonctionne toujours pas .

Ce problème est typiquement causé par des permissions plus larges que celles permises par sshd par défaut pour les répertoires \$HOME, \$HOME/.ssh ou le fichier \$HOME/.ssh/authorized_keys.

Dans ce cas, l'exécution des commandes suivantes sur le serveur résoudra le problème :

chmod 600 \$HOME/.ssh/authorized_keys

\$chmod go-w \$HOME \$HOME/.ssh

Maintenant, si vous vous connectez par ssh on ne doit plus vous demander de mot de passe.

Vérifier que vous pouvez exécuter des commandes à distance (sans mot de passe) :

par exemple ssh 192.168.X.Y date (ou ce que vous voulez...)

2 Rsync et UNISON

UNISON est un logiciel libre (disponible sur presque toutes les plate formes) de synchronisation bidirectionnel : il répercute les modifications qui apparaissent dans l'un ou l'autre des répertoire synchronisé. Il utilise Rsync qui est un logiciel de synchronisation unidirectionnel.

Installer unison sur **les deux machines**.

Vérifier le bon fonctionnement en affichant la version :
unison -v

créer deux repertoires locaux par exemple upload et download.
mettez un fichier toto.txt dans upoad.
Puis **unison upload download**

vous devez avoir toto.txt dans l'autre répertoire. Supprimez le et relancez:
unison upload download

Les deux répertoires doivent être vide.

Vous avez peut être le mode graphique par défaut: ceci est bien commode en interactif, mais inutilisable pour une fonction de backup automatique.
Pour lancer unison en mode texte ajoutez l'option **-ui text (il faudra par la suite ajouter également -batch et -silent)**.

Maintenant, nous allons synchroniser un répertoire synchro (de votre compte) sur le PC distant avec ssh :

unison -ui text /home/moncpte/synchro
ssh://192.168.X.Y//home/moncpte/synchro

Vérifier le bon fonctionnement en ajoutant/supprimant des fichiers dans les répertoires, créer des sous-répertoires etc..

Que se passe t-il si on s'amuse à changer l'heure sur l'un des deux PC ? Conclusion?

Pour éviter des lignes de commandes interminables on va placer dans le répertoire .unison de votre compte un fichier de configuration mesynchro.prf qui contiendra par exemple:

```
# Unison preferences file
root= /home/moncpte/
root= ssh://192.168.x.y//home/moncpte/
path= synchro
```

vous pouvez ajouter autant de path que vous volulez pour synchroniser d'autres répertoires (par rapport à la racine /home/moncompte).

maintenant la ligne de commande devient **unison mesynchro**
vous pouvez ajouter d'autres directives(ignore, ui, etc ...) dans le fichier prf.

3 Cron

On souhaite maintenant synchroniser toutes les minutes des répertoires sur le réseau.
Une option d'unison pourrait permettre de le faire (REPEAT) mais elle est rarement compilée pour les distributions courantes. On utilise donc le service standard de gestion temporelle des tâches : CRON.

Modifier le fichier /etc/crontab pour qu'il exécute cette synchronisation toutes les minutes (avec votre profil et pas celui de root sinon ssh demandera un mot de passe...)
sans rien demander à l'utilisateur.

En éditant régulièrement le fichier unison.log vérifier le bon fonctionnement.

Essayez de synchroniser deux PC distants sur lesquels vous avez des comptes, à partir de votre cron local.