Documentation système personnelle

Nicolas Vacelet

27 mai 2015

Résumé

Documentation système à des fins personnelles

Table des matières

1 Introduction

Cette documentation sert à organiser mes différentes recherches et à capitaliser ce savoir pour en faire un référentiel.

2 chroot

http ://karma-lab.net/magie-chroot

Le système fichier d'un *NIX est construit autour d'une racine (le /) sur laquelle les partitions sont ensuite "montées" formant ainsi l'espace de fichier accessible. Cette racine est la référence pour tous les chemins absolus utilisés par un processus lui permettant d'accéder aux fichiers (librairies, configurations, etc.) qui lui sont nécessaires.

Ce que l'on sait moins c'est que cette racine est un paramètre du processus qu'il est parfaitement possible de modifier grace à l'utilitaire chroot. Pour quoi faire me direz-vous ? Tout simplement pour faire croire à ce processus que le dossier que nous lui avons arbitrairement fixé comme racine, est l'origine de tous ses chemins absolus.

Le cadre d'usage de cette technique est vaste. Elle permet par exemple de lancer des processus critiques dans un dossier isolé du reste du système de sorte à rendre plus difficile (mais pas impossible !) la compromission du reste du système de fichier en cas d'exploitation d'une faille de sécurité. C'est ce que l'on appelle mettre en prison le processus (jail).

Elle permet aussi de créer des environnements qui fonctionnent sur des règles différentes du reste du système. Il est ainsi possible de faire tourner une debian au sein d'une mandriva, ou encore un linux en 32bits au sein d'un linux 64 bits. La seule limitation est que le kernel doit être compatible entre les deux distributions.

Mais même si cela sonne terriblement comme de la virtualisation il est important de ne pas confondre les deux principes. Le changement de racine n'émule absolument rien. Ce n'est que l'exploitation d'une propriété des processus unix. Chaque processus chrooté accède donc au même matériel que les processus "normaux". De même ils tournent au sein du même kernel et partagent le même espace mémoire. Plus flagrant, les processus lancés dans le cadre d'un changement de racine sont parfaitement visible si l'on exécute une commande ps à partir d'un shell "normal".

Et c'est là finalement la grande force du changement de racine. C'est un principe limité mais simple, et qui ne souffre d'aucun problème de performance accompagnant généralement la virtualisation.

2.1 Construction d'un dossier chrootable

Il y a d'innombrables manière de créer un dossier utilisable pour un changement de racine. Tout dépend au fond du besoin derrière la manipulation. Si l'on désire chrooter un service FTP par exemple, on va s'attacher à ne mettre dans le dossier que les fichiers strictement nécessaires au processus : l'exécutable, ses librairies et ses fichiers de configurations, placé avec soin dans une arborescence qui mime parfaitement ce que l'on trouverait sur une racine réelle. Il existe des utilitaires qui aident à effectuer ce genre de tâche mais ce n'est pas forcement l'approche la plus didactique.

Pour notre exemple, voyons plutôt comment créer une debian complète dans notre racine de sorte à y placer une pile Apache, PostgreSql et PHP.

La première chose à faire est d'installer dans un dossier de votre choix une installation de la distribution Debian. Il est possible de faire bêtement une copie de la racine principale, cela marcherait très bien, mais ce serait sûrement très volumineux.

Heureusement, il existe un outil debian qui fait des miracles debootstrap. Notez que cet outil n'est pas spécifique à Debian et fonctionne aussi très bien sous Mandriva. Nous allons donc installer la commande et la lancer pour installer une wheezy tout neuve.

```
$ sudo apt-get install debootstrap
$ debootstrap --include=locales-all wheezy ma_racine http://ftp.fr.debian.org/debian
```

Le paramètre –include indique à debootstrap des paquets supplémentaires à installer, ici les locales. Suit la version de debian à installer et le dossier dans lequel effectuer les opérations. Si ce dossier n'existe pas il sera créé. Enfin nous avons l'url des paquets debian en france.

Et voilà, nous sommes dans notre bash chrooté. Mais que c'est il passé exactement ?

```
# on met un petit fichier témoin dans la racine
$ echo "Je suis dans ma racine..." | sudo tee ma_racine/ça_marche.txt > /dev/null

# lancement d'un bash en chroot
$ chroot ma_racine /bin/bash

# On vérifie que ça a bien marche...
$root cat /ça_marche.txt
#Je suis dans ma racine...
```

- La commande chroot engendre un processus qui a pour racine la même que celle de son processus parent, généralement le / (mais rien n'empêche de se la jouer matryoshka ;-).
- En interne, chroot appel de la fonction kernel chroot("/ma_racine"). Le kernel va donc modifier la valeur de la racine pour ce processus et lui associer la valeur /ma_racine.
- Le processus de chroot, exécute la commande passée en 2nd paramètre, /bin/bash. Comme la racine de chroot a été changée, c'est bien le /bin/bash de la nouvelle racine /ma_racine qui va être exécuté.
- Cette exécution débouche sur la création d'un processus fils de celui du chroot qui hérite logiquement de cette nouvelle racine.
- Tout ce que /bin/bash lancera par la suite héritera de cette nouvelle racine jusqu'à ce que l'on tape exit qui mettra fin au processus /bin/bash, et par domino à celui du chroot qui a permis son lancement.

Ok, le bash c'est bien mignon mais voyons comment aller un peu plus loin.

2.2 Dossiers spéciaux

Notre debian a besoin d'un plus que les fichiers installés pour fonctionner. En effet pour accéder au matériel il lui faudra les fameux dossier /proc et /sys. Dans un premier temps, nous allons faire cela à la main.

3 Commandes

- tee : écrit en sortie ce qu'il a récupéré en entrée : echo "hello" | tee out.txt > /dev/null