



Infrastructures Virtuelles et Conteneurs

Legond-Aubry Fabrice

fabrice.legond-aubry@parisnanterre.fr

TP outils: chroot

- Utiliser fedora sous virtualbox
 - ✓ Créer une VM avec une fedora 31+
 - ✓ https://ftp.lip6.fr/ftp/pub/linux/distributions/fedora/releases/33/Server/x86_64/iso/Fedora-Server-dvd-x86_64-33-1.2.iso
 - ✓ Créer une VM et booter sur le CD, faites l'installation
- **Passer root**
- chroot de la commande **ls**
 - ✓ Utiliser **ldd** sur la commande **ls**
 - ✓ En tant que root:
 - Créer un répertoire **jail** (répertoire de prison)
 - Recréer la structure des bibliothèques utilisées
 - Lancer la commande **chroot** pour **ls** avec "**strace**"
 - Si, c'est ok, vous pouvez tenter d'enlever le strace

TP outils: chroot

- A faire vous-même : chroot de bash, busybox puis vim
 - ✓ Utiliser ensuite "`busybox -install /bin`" pour installer un ensemble de commandes basiques
 - ✓ Ajout des messages localisés (LC_MESSAGES) – `strace/grep ".mo"`
 - ✓ Ajout des informations de terminal pour vi (`dev/tty` et `terminfo`)
 - ✓ Faire un fichier "`.bashrc`" pour fixer les path
- Pour VIM
 - ✓ on peut optimiser un peu en utilisant les packages des distributions
 - ✓ **Note: Cette solution dépend de la distribution !!!**

TP outils: chroot

- Pour VIM, l'autre solution (spécifique fedora)
 - ✓ chercher sur internet les commandes rpm2cpio, cpio
 - ✓ wget
http://ftp.proxad.net/mirrors/fedora.redhat.com/fedora/linux/releases/33/Everything/x86_64/os/Packages/v/vim-enhanced-8.2.1770-1.fc33.x86_64.rpm
 - ✓ wget
http://ftp.proxad.net/mirrors/fedora.redhat.com/fedora/linux/releases/33/Everything/x86_64/os/Packages/v/vim-common-8.2.1770-1.fc33.x86_64.rpm
 - ✓ rpm2cpio vim-common-8.2.1770-1.fc33.x86_64.rpm > vim-common-8.2.1770-1.fc33.x86_64.rpm.cpio
 - ✓ rpm2cpio vim-enhanced-8.2.1770-1.fc33.x86_64.rpm > vim-enhanced-8.2.1770-1.fc33.x86_64.rpm.cpio
 - ✓ cd jail
 - ✓ cpio -v -i -d --no-preserve-owner --no-absolute-filenames < ../vim-enhanced-8.2.1770-1.fc33.x86_64.rpm.cpio
 - ✓ cpio -v -i -d --no-preserve-owner --no-absolute-filenames < ../vim-common-8.2.1770-1.fc33.x86_64.rpm.cpio

TP outils: chroot

- Récupérer la programme "`mon_app.c`"
 - ✓ Compilez-le
 - ✓ Copiez-le dans le répertoire bin de votre "prison"
 - ✓ Faites les montages nécessaires

TP outils: cgroup

- Hiérarchie des cgroup (répertoires des cgroup)
 - ✓ **CPU** : The “cpu” controllers regulates distribution of CPU cycles.
 - ✓ **Memory** : The “memory” controller regulates distribution of memory.
 - ✓ **IO** : The “io” controller regulates the distribution of IO resources.
 - ✓ **PID** : The process number controller is used to allow a cgroup to stop any new tasks from being fork()'d or clone()'d after a specified limit is reached.
 - ✓ **Device controller** : Device controller manages access to device files.
 - ✓ **RDMA** : The “rdma” controller regulates the distribution and accounting of RDMA resources.
- Note : on peut utiliser des fichiers de configuration
- Note: les outils "libcgroup-tools" sont pour les cgroup v1
- Attention : On ne peut associer des processus qu'à des cgroup feuilles
 - ✓ i.e. des cgroup sans sous-éléments (pas de mkdir de sous cgroup)

TP outils: cgroup

- Attention cgroupv1 et cgroupv2 cohabitent mais peuvent se voler
 - ✓ Un gestionnaire en v1 est peut être géré en v2 (mémoire, ...)
 - ✓ Le plus simple : n'utiliser que l'un des 2
- Nous n'utiliserons que la version 2
 - ✓ Sur fedora 33, mount :
 - `"cgroup2 on /sys/fs/cgroup type cgroup2 (...)"`
 - ✓ Ou autre possibilité,
 - `"mount -t cgroup2 none ./cgroupv2"`
- Liste des modules cgroup montés :
 - ✓ `"cat ./cgroupv2/cgroup.controllers"`
- Créer un cgroup nommé mon_conteneur
 - ✓ `"mkdir CGROUP_NAME" / "rmdir CGROUP_NAME"`
 - ✓ Permettant le contrôle de la mémoire, cpu, et io

TP outils: cgroup / namespaces

- Fixer la mémoire maximum
 - ✓ Fichier `memory.high` et `max` du sous répertoire "`mon_conteneur`"
 - ✓ "`cat memory.high`" puis fixer sa valeur à 32mo ($32*1024*1024$)
 - ✓ Interdire le "swap" en mettant "0" dans "`memory.swap.max`"
- Exécuter votre commande chroot avec votre bash
 - ✓ Trouver le pid du processus bash
 - ✓ Associer le processus à votre nouveau cgroup
 - via le fichier "`cgroups.procs`" du répertoire "`mon_conteneur`"
 - ✓ Exécuter le programme "`mon_app`" à nouveau
- Exécuter le bash cette fois-ci avec la commande "`unshare`" au lieu de "`chroot`"
 - ✓ Tester `id/ps/ls`
 - ✓ pid du processus dans le bash
 - ✓ Tenter "`poweroff`"