Compiler un kernel pour LXC

Un noyau LXC sur serveur dédié OVH

jeudi 20 juin 2013, par Pierrox

Récupérer les sources du noyau, le patch grsec et la config ovh qui va bien

Se rendre dans un dossier de travail

1. **cd** /usr/src

Téléchargement du kernel

1. **wget** ftp://ftp.kernel.org/pub/linux/kernel/v3.x/linux-3.9.6.tar.xz

Téléchargement du patch grsecurity [1] correspondant à la version du noyau (3.9.6 dans l'article)

1. **wget** https://raw.github.com/slashbeast/grsecurity-scrape/master/test/grsecurity-2.9.1-3.9.6-201306182033.patch

récuperer la config du noyaux ovh sur leur serveur ftp [2]

1. **wget** ftp://ftp.ovh.net/made-in-ovh/bzImage/latest-production/config-3.8.13-xxxx-grs-ipv6-64

Préparer les sources

décompresser [3] le noyau

1. **tar** -Jxf linux-3.9.6.tar.xz

copier la configuration ovh dans racine du noyau linux décompressé précédemment

1. **cp** config-3.8.13-xxxx-grs-ipv6-64 linux-3.9.6

aller dans le répertoire du noyau fraichement décompressé

1. **cd** linux-3.9.6

appliquer le patch grsec

1. `patch -p1 < ../grsecurity-2.9.1-3.9.6-201306182033.patch`

Configuration des cgroups

dans le repertoire du kernel, lancer l'utilitaire de configuration du noyau `make menuconfig` et configurer lxc cgroup [4] en activant les options suivantes :

```
-> General setup
    [renommer votre kernel] Local version - append to kernel release
    -> Control Group support
        [x] Namespace cgroup subsystem
        [x] Freezer cgroup subsystem
        [x] Cpuset support
        [x] Simple CPU accounting cgroup subsystem
        [x] Resource counters
        [x] Memory resource controllers for Control Groups
        -> Group CPU scheduler
            [x] Basis for grouping tasks (Control Groups) (!)
            [x] Group scheduling for SCHED_OTHER (NEW)
            [x] CPU bandwidth provisioning for FAIR_GROUP_SCHED
            [x] Group scheduling for SCHED_RR/FIFO
        -> Namespaces support
            [x] UTS namespace
            [x] IPC namespace
            [x] User namespace (!)
            [x] Pid namespace
            [x] Network namespace
-> Device Drivers
    -> Character devices
        [x] Support multiple instances of devpts
        [*] Unix98 PTY support
    -> Network device support
        [x] MAC-VLAN support
        [x] Virtual ethernet pair device
-> Networking support
    -> Networking options
        [x] 802.1d Ethernet Bridging
        [x] Network priority cgroup
    -> Security options
        [x] File POSIX Capabilities (!)
```

Télécharger

(!) suivant la version/configuration du kernel certaines options peuvent être déplacées ou inexistantes. Pour activer l'option **usernamespace** je suis tombé sur un bug pour le noyau 3.8. De plus une contrainte oblige a désactiver le système de fichier **XFS** [5] pour pouvoir activer cette option. XFS devrait être supporté à partir des versions 3.10 du kernel linux.

Une fois votre configuration terminée, retourner sur la page d'accueil de menuconfig sélectionner <

`Exit >`, une boite de dialogue vous demande d'enregistrer votre configuration selectionner `< yes >` avant de quitter.

Avant de compiler, On va tester la configuration :

```
CONFIG=/usr/src/linux-3.9.6/.config lxc-checkconfig

 --- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled
Multiple /dev/pts instances: enabled

 --- Control groups ---
Cgroup: enabled
Cgroup clone_children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: missing (!)
Cgroup cpuset: enabled

 --- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
File capabilities: enabled
```

Cgroup memory controller : missing (!) - Debian Wheezy

Sur Debian wheezy, avec le paquet lxc, la configuration de "memory controller" semble manquante. En installant la dernière version de lxc (0.9) la vérification devient OK

Youpi ! la configuration est maintenant terminée, il est temps de lancer la compilation avant de boire un café !

Compilation du noyau

Penser à installer les outils pour la compilation des noyaux. Sur Debian, le paquet s'appelle build-essential c'est un meta-paquet dédié a l'installation de tous les outils de compilation make, gcc, etc..

Lancer la compilation, ou -j X (correspond au nombre de core) :

1. **make** -j 8

Le kernel se retrouve dans le dossier arch/x86/boot. On va Copier/renommer le nouveau kernel dans le dossier /boot du système :

1. **cp** arch/x86/boot/bzImage /boot/bzImage-3.9.6-xxxx-grs-ipv6-64-lxc

Installer le kernel et redémarrer

Je vous invite à consulter mon article précédent intitulé Installer un nouveau kernel Linux en 5 étapes

---

P.-S.

Configuration du kernel réalisé à partir de http://lxc.sourceforge.net/man/lxc.html

---

Notes

[1] tous les patchs Grsec sur github

[2] Tous les noyaux ovh ftp://ftp.ovh.net/made-in-ovh/bzImage

[3] package xz-utils sur debian et consorts

[4] d'aprés la page du wiki de lxc sur le sourceforge officiel

[5] bug 917708 de compatibilité