Utilisation de LXC en mode utilisateur non-privilégié

- Objet : Utiliser LXC sans être root, et lancer des instances non-root
- Niveau requis :
  avisé
- Commentaires : *Des conteneurs sécurisés.*
- Suivi :
  - Création par captnfab 09/12/2015
  - Testé par captnfab le 09/12/2015
  - Testé et mis à jour par captnfab le 17/06/2018
  - Testé et mis à jour par jeremyp3 le 18/11/2020
- Commentaires sur le forum : ici [1)]

Nota :

Contributeurs, les Fix Me! sont là pour vous aider, supprimez-les une fois le problème corrigé ou le champ rempli !

Introduction

Si vos conteneurs LXC doivent contenir des serveurs sensibles aux attaques, vous ne voudriez pas qu'un serveur compromis compromette tout le reste du système. Ne pas donner les privilèges super-utilisateur au conteneur permet de limiter ce risque.

Installation

Les outils de gestion lxc :

```
apt-get install lxc uidmap
```

D'autres outils normalement présents par défaut :

```
apt-get install libpam-systemd cgroup-bin bridge-utils
```

Et des outils de débuggage :

```
apt-get install less
```

Préparation

Nous allons créer un conteneur Test, qui appartiendra à l'utilisateur lxcuser-test, et accédera au réseau via le pont lxcbr0. Cette opération sera à répéter pour chaque utilisateur devant pouvoir créer

un ou plusieurs conteneurs. C'est la seule partie du tutoriel qui doit nécessairement être effectuée en root.

Création et configuration de l'utilisateur

Création de l'utilisateur

```
adduser --disabled-password lxcuser-test
```

L'option `--disabled-password` n'est pas obligatoire, elle permet simplement de s'assurer que la connexion par mot de passe est impossible via ssh.

Comme vous devrez vous connecter à cet utilisateur via ssh, je vous conseille de créer dès maintenant une paire de clefs via ssh-keygen, et de rajouter la clef publique aux clefs autorisées pour cet utilisateur.

Nous devons ensuite nous assurer qu'une plage de sous-uid/sous-gid a été affectée à l'utilisateur.

Identification et affectation d'une plage d'uid/gid libre

```
cat /etc/subuid
```

Exemple :

```
nm-openvpn:100000:65536
pulse:165536:65536
mpd:231072:65536
geoclue:296608:65536
systemd-timesync:362144:65536
systemd-network:427680:65536
systemd-resolve:493216:65536
systemd-bus-proxy:558752:65536
```

Chaque ligne du fichier commence par un nom d'utilisateur, suivi par un numéro d'uid et d'un nombre d'uid. Par exemple, ici, il faut comprendre que les uid 100000 à 165535 sont des sous-uid dépendant de l'utilisateur nm-openvpn. Voir le man

```
man subuid
```

Si lxcuser-test **n'apparaît pas dans la liste**, alors on va l'ajouter, sinon, passer directement au paragraphe suivant. Dans notre exemple, le premier uid de libre est le 558752+65536=624288, donc notre utilisateur utilisera la plage d'uuid 624288 à 689823 (624288+65535). Nous faisons ensuite la même opération pour les gid, en consultant /etc/subgid. Dans notre exemple, on considèrera que le fichier /etc/subgid est identique au fichier /etc/subuid, ce qui est en général le cas.

Nous allouons alors les sous-uid et sous-gid à l'utilisateur test via la commande (**à modifier en fonction de vos propres `/etc/sub{u,g}id` !**) :

```
usermod lxcuser-test --add-subuids 624288-689823 --add-subgids 624288-689823
```

Création et configuration de l'accès au pont réseau

Création du pont

```
brctl addbr lxcbr0
```

(Penser à automatiser sa création, par exemple via le fichier `/e/n/interfaces` ou via Network-Manager)

Accès

On autorise l'utilisateur lxcuser-test à connecter une interface *ethernet* virtuelle au port (vous pouvez augmenter ce nombre selon vos besoins, fichier à créer) :

/etc/lxc/lxc-usernet

```
lxcuser-test veth lxcbr0 1
```

Configuration cgroups

Il faut activer certaines fonctionnalités des cgroups pour que l'on puisse lancer les conteneurs utilisateurs.

Configuration du noyau

Vérifier que le fichier `/sys/fs/cgroup/cpuset/cgroup.clone_children` est bien à 1 par défaut sur le système.

```
cat /sys/fs/cgroup/cpuset/cgroup.clone_children
```

Créer le fichier de configuration de sysctl suivant :

/etc/sysctl.d/40-lxc-userns.conf

```
kernel.unprivileged_userns_clone=1
net.bridge.bridge-nf-call-arptables=0
net.bridge.bridge-nf-call-iptables=0
```

```
net.bridge.bridge-nf-call-ip6tables=0
```

Et le charger dans le système :

```
sysctl --sys
```

Création des cgroups

Pour l'instant, les cgroups nécessaires ne sont pas créés par défaut. Créer les fichiers suivant :

/usr/local/sbin/prepare-lxc-cgroups

```
#!/bin/sh
LXC_USERS=$(cat /etc/lxc/lxc-users)
for d in /sys/fs/cgroup/*
do
  f=$(basename $d)
  if [ "$f" = "cpuset" ]
  then
    echo 1 > $d/cgroup.clone_children;
  elif [ "$f" = "memory" ]
  then
    echo 1 > $d/memory.use_hierarchy;
  fi
  for u in $LXC_USERS
  do
    mkdir -p $d/$u
    chown -R $u $d/$u
  done
done
```

et

/usr/local/bin/move-pid-to-cgroup

```
#!/bin/sh
PID=$1
for d in /sys/fs/cgroup/*
do
        echo $PID > $d/$USER/tasks
done
```

à rendre exécutables :

```
chmod a+x /usr/local/sbin/prepare-lxc-cgroups
```

```
chmod +x /usr/local/bin/move-pid-to-cgroup
```

Accès

Et rajouter `lxcuser-test` dans le fichier `/etc/lxc/lxc-users`. S'il y a plusieurs utilisateurs, ils doivent être séparés par un espace.

/etc/lxc/lxc-users

```
lxcuser-test
```

Automatisation

Que l'on lancera automatiquement via systemd en créant ce service :

/etc/systemd/system/lxc-cgroups.service

```
[Unit]
Description=Préparation cgroups pour lxc
After=local-fs.target

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/local/sbin/prepare-lxc-cgroups

[Install]
WantedBy=multi-user.target
```

Et en l'activant :

```
systemctl enable /etc/systemd/system/lxc-cgroups.service
systemctl daemon-reload
systemctl start lxc-cgroups
```

Création du conteneur

Le code à faire en tant qu'utilisateur est à faire en tant que `lxcuser-test`.

Configuration des options par défaut

Dans le dossier ~`lxcuser-test/.config/lxc/` (probablement à créer), créer un fichier

`default.conf` contenant la configuration suivante :

depuis Debian buster :

~lxcuser-test/.config/lxc/default.conf

```
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
# À compléter pour définir une adresse mac
#lxc.net.0.hwaddr = 00:16:3e:xx:xx:xx
lxc.idmap = u 0 624288 65536
lxc.idmap = g 0 624288 65536
```

Remplacer *624288* sur les deux dernières lignes par les valeurs du premier sous-uid/gid alloué à l'utilisateur.

Pour les anciennes version de Debian ou de LXC:

~lxcuser-test/.config/lxc/default.conf

```
lxc.network.type = veth
lxc.network.link = lxcbr0
lxc.network.flags = up
# À compléter pour définir une adresse mac
#lxc.network.hwaddr = 00:16:3e:xx:xx:xx
lxc.id_map = u 0 624288 65536
lxc.id_map = g 0 624288 65536
```

Remplacer *624288* sur les deux dernières lignes par les valeurs du premier sous-uid/gid alloué à l'utilisateur.

```
lxc-create -n test -t download -- -d debian -r buster -a amd64
```

Petit résumé des options :

- -n: Nom du conteneur
- -t : template a utiliser, ici download

après les – on donne les paramètres du template :

- -d : Nom de la distribution dans notre cas: Debian.
- -r: le nom de la release par exemple : buster, stretch, sid, testing ...
- -a : architecture de la distribution qui doit être la même que le système dans notre cas: amd64.

Configuration du conteneur

Le fichier de configuration `~/.local/share/lxc/test/config` doit ressembler à ça :

Depuis Debian buster :

~/.local/share/lxc/test/config

```
# Template used to create this container: /usr/share/lxc/templates/lxc-
download
# Parameters passed to the template: -d debian -r buster -a amd64
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# À compléter pour définir une adresse mac

# Distribution configuration
lxc.include = /usr/share/lxc/config/common.conf
lxc.include = /usr/share/lxc/config/userns.conf
lxc.arch = linux64

# Container specific configuration
lxc.idmap = u 0 624288 65536
lxc.idmap = g 0 624288 65536
lxc.rootfs.path = dir:/home/lxcuser-test/.local/share/lxc/test/rootfs
lxc.uts.name = test

# Network configuration
lxc.net.0.type = veth
lxc.net.0.link = lxcbr0
lxc.net.0.flags = up
#lxc.network.hwaddr = 00:16:3e:xx:xx:xx
```

Pour les anciennes version de Debian ou de LXC:

~/.local/share/lxc/test/config

```
# Template used to create this container: /usr/share/lxc/templates/lxc-
download
# Parameters passed to the template: -d debian
# Template script checksum (SHA-1):
01d100d3f1129082777c82a0e3a66adcaeb5c37f
# For additional config options, please look at lxc.container.conf(5)

# Uncomment the following line to support nesting containers:
#lxc.include = /usr/share/lxc/config/nesting.conf
# (Be aware this has security implications)

# À compléter pour définir une adresse mac
```

```
# Distribution configuration
lxc.include = /usr/share/lxc/config/debian.common.conf
lxc.include = /usr/share/lxc/config/debian.userns.conf
lxc.arch = linux64

# Container specific configuration
lxc.id_map = u 0 624288 65536
lxc.id_map = g 0 624288 65536
lxc.rootfs = /home/lxcuser-test/.local/share/lxc/test/rootfs
lxc.rootfs.backend = dir
lxc.utsname = test

# Network configuration
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = lxcbr0
#lxc.network.hwaddr = 00:16:3e:xx:xx:xx
```

Les droits des dossiers doivent ressembler à cela :

```
ls -lhd /home{,/lxcuser-test{,/.local{,/share{,/lxc{,/test}}}}}
```

```
drwxr-xr-x 5 root    root    4,0K déc.  17 00:07 /home
drwxr-xr-x 3 lxcuser-test lxcuser-test 4,0K déc.  17 00:56 /home/lxcuser-
test
drwxr-xr-x 3 lxcuser-test lxcuser-test 4,0K déc.  17 00:56 /home/lxcuser-
test/.local
drwxr-xr-x 3 lxcuser-test lxcuser-test 4,0K déc.  19 13:49 /home/lxcuser-
test/.local/share
drwxr-xr-x 3 lxcuser-test lxcuser-test 4,0K déc.  19 13:53 /home/lxcuser-
test/.local/share/lxc
drwxrwx--x 3       624288 lxcuser-test 4,0K déc.  19 13:51 /home/lxcuser-
test/.local/share/lxc/test
```

```
ls -lh /home/lxcuser-test/.local/share/lxc/test
```

```
total 8,0K
-rw-r--r--  1 lxcuser-test lxcuser-test  488 déc.  19 13:51 config
-rw-r--r--  1 lxcuser-test lxcuser-test    0 déc.  19 13:51 fstab
drwxr-xr-x 22       624288       624288 4,0K déc.  19 13:51 rootfs
```

Vérifier les propriétaires suivant ses propres sous-uid/sous-gid. Vérifiez bien les droits de lecture/écriture/exécution pour l'utilisateur, le groupe et les autres.

Utilisation du conteneur

Il faut, en tant que `lxcuser-test`, et dans le shell qui lancera le conteneur, exécuter la commande

suivante (qui ajoutera le shell au cgroup) :

```
move-pid-to-cgroup $$
```

- Lancement

```
lxc-start -n test -d
```

- Shell

```
lxc-attach --set-var
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin -n test
```

Sans l'option `–clear-env`, vous conservez les variables d'environnement de votre utilisateur. Ce n'est en général pas gênant, mais ces variables incluent aussi le PATH… Cependant, clear-env enlève aussi des variables intéressantes comme TERM. Pour ne pas vous retrouver avec un shell root avec un path utilisateur (ne contenant pas sbin) ce qui peut facilement générer des erreurs, ou un terminal mal configuré, utilisez l'option `–set-var`.

FAQ / Problèmes rencontrés

Messages d'erreur rencontrés en lançant les différentes commandes du tuto, ou encore journalctl -xe

- `lxc_container: Failed to chown /dev/pts/X`, `lxc_start - start.c:lxc_init:445 - Failed to shift tty into container`

Vérifier que le paquet `uidmap` est bien installé Vérifier que `/proc/sys/kernel/unprivileged_userns_clone` et `/sys/fs/cgroup/cpuset/cgroup.clone_children` sont bien à 1

- `lxc_container: failed to clone (0x20000011): Operation not permitted`

Vérifier que le fichier de config `~lxcuser-test/.local/share/lxc/test/config` contient bien des mappings de sous-uid/sous-gid corrects (exemple) :

```
lxc.id_map = u 0 820896 65536
lxc.id_map = g 0 820896 65536
```

- `pam_unix(sudo:auth): conversation failed` ou `pam_unix(sudo:auth): auth could not identify password for [lxcuser-test]` ou `lxc-start: Permission denied - failed to create directory '/run/user/1000/lxc/'`

Vérifier que l'on s'est bien connecté en tant que lxcuser-test via PAM (par exemple via ssh et non-pas par su/sudo)

- `ERROR lxc_cgfs - Permission denied - Could not create cgroup '/test' in '/sys/fs/cgroup/perf_event'.`, `lxc_container: cgfs.c: lxc_cgroupfs_create: 956 Permission denied - Could not create cgroup '/lxc' in`

`'/sys/fs/cgroup/cpuset'.`

Vérifier que nous ne sommes pas à la racine dans les cgroups listés dans `/proc/self/cgroup` Exemple de config problématique :

```
8:perf_event:/
7:blkio:/
…
1:name=systemd:/user.slice/user-1001.slice/session-42.scope
```

Vous devriez plutôt obtenir quelque chose comme

```
8:perf_event:/lxcuser1/
7:blkio:/lxcuser1/
…
1:name=systemd:/user.slice/user-1001.slice/session-42.scope
```

Ce problème survient quand libpam-systemd (qui ajoute automatiquement les nouvelles connexions aux bons cgroups) n'est pas installé, où quand les cgroups users ne sont pas créés dans les différents contrôleurs, ce qui est le comportement par défaut sous Debian, mais pas sous Ubuntu. Reportez-vous à la partie du tuto sur les cgroups si vous avez des soucis.

- `lxc_container: cgmanager.c: lxc_cgmanager_enter: 698 call to cgmanager_move_pid_sync failed: invalid request`

Muh ? Visiblement, problème avec une version cgmanager trop récente (stretch/sid et non jessie)

- `Quota reached`, `lxc_start - failed to create the configured network`

Vérifier que le mode de connexion renseigné dans `/etc/lxc/lxc-usernet` existe bien, et que c'est bien celui indiqué dans `~lxcuser-test/.local/share/lxc/test/config`.

- Vérifier que tous les dossiers parents du dossier `rootfs` sont bien exécutables par Others (ou au moins par le min de la plage de sous-uid correspondante).
- Vérifier que le fichier `~lxcuser-test/.local/share/lxc/test/config` contient bien la ligne `lxc.autodev = 0`

- `lxc_container: No such file or directory - failed to get real path for '/var/lib/lxc/test/rootfs`

Vérifier que le path du rootfs indiqué dans `~lxcuser-test/.local/share/lxc/test/config` est bien celui contenant le rootfs du conteneur, donc a priori `~lxcuser-test/.local/share/lxc/test/rootfs`

- `lxc_container: Error setting devices.deny to a for test` ou `lxc_container: failed to setup the devices cgroup for 'test'`

Vérifier que le fichier `~lxcuser-test/.local/share/lxc/test/config` contient bien la ligne

```
lxc.include = /usr/share/lxc/config/debian.userns.conf
```

- `Set hostname to <test>. Failed to create /init.scope control group: Permission denied Failed to allocate manager object: Permission denied`

```
[!!!!!!] Failed to allocate manager object. Exiting PID 1…
```

cela arrive quand le paquet `libpam-cgfs` n'est pas installé.

Sources

- https://www.mail-archive.com/lxc-devel@lists.linuxcontainers.org/msg01660.html
- https://www.stgraber.org/2014/01/17/lxc-1-0-unprivileged-containers/
- http://unix.stackexchange.com/questions/170998/how-to-create-user-cgroups-with-systemd
- http://www.linuxquestions.org/questions/linux-kernel-70/lxc-unprivileged-container-in-debian-jessie-cgroups-permissions-4175540174/
- http://www.equiscentrico.com.ar/2015/05/unprivileged-lxc-en-debian-jessie.html

[1)]

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !