Sandboxing / KB mars 2013

CAHIER DE CHARGE

Description

Dans le cadre de la recherche appliquée sur les solutions de virtualisation et leur sécurisation, le laboratoire de transmission de données mène l'étude sur Linux-KVM et SELinux.

L'objectif est d'acquérir des connaissances poussées afin de définir une best practice et des SOP qui sécurisent, à l'aide de politique de contrôle obligatoire d'accès et de bac à sable, un serveur virtualisé.

Au final, nous souhaitons proposer un SaaS avec une sécurisation présente à tous les niveaux.

Public cible

- Développeurs
- Administrateurs

Logiciels utilisés

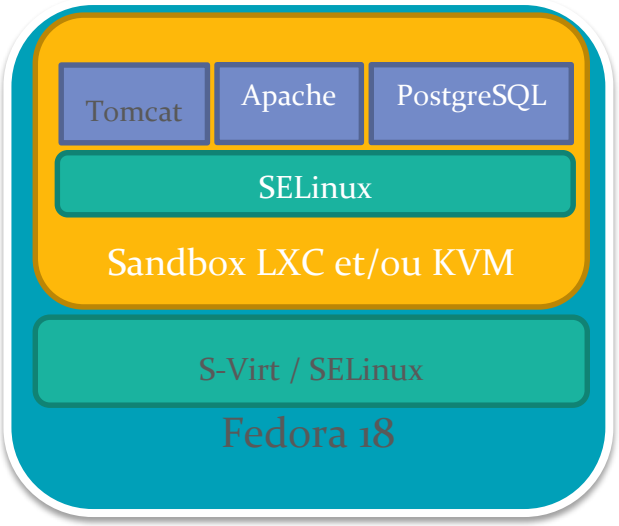
- Fedora 18 64bit
- Dernière version KVM
- Dernière version de sandbox
- Dernière version de tomcat
- Dernière version de PostgreSQL

Démarche

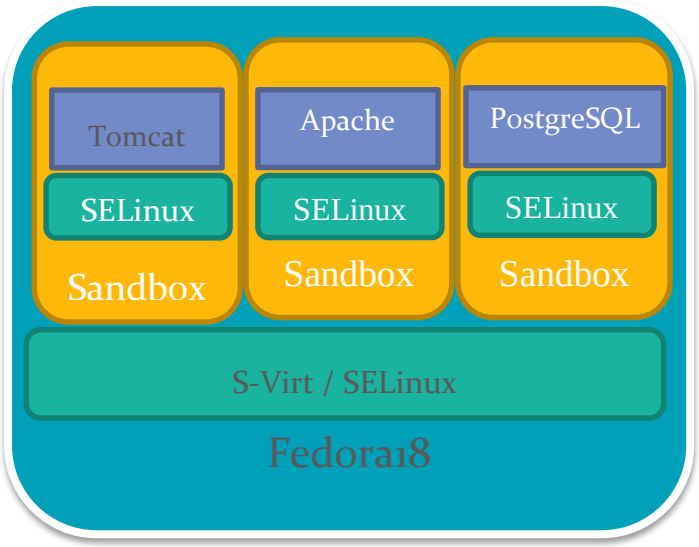
- Étude détaillée du fonctionnement de l'outil virt-sandbox et de virt-sandbox-service
- Définition des cas de tests
    - Injection SQL
    - Tests de pénétration
    - Tests de performance
- Confiné un serveur web Tomcat et une base de données PostgreSQL
- Expérimenter les différentes variantes

Variantes :

A :



B :



KVM

LINUX CONTAINERS

LXC est la contraction de l'anglais Linux Containers est un système de virtualisation au niveau système d'exploitation utilisé pour faire fonctionner de multiples environnements Linux isolés les uns des autres sur un seul et même système hôte. Le conteneur LXC n'est pas une machine virtuelle mais uniquement un environnement virtuel qui dispose de ses propres processus et de son propre réseau (isolés du système physique hôte).

Excellent guide de Red Hat sur la gestion des ressources à l'aide des Cgroups :

Fichier liens : Red Hat Enterprise Linux 6 - Resource Management Guide

https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Resource_Management_Guide/ch01.html

s-Virt → libvirt security driver

virt-sandbox → libvirt-sandbox

LIBVIRT-SANDBOX

Disponible à partir de Fedora 17

Supporte uniquement LXC et QEMU/KVM. Le démon libvirt comprend un pilote LXC, un pilote KVM et encore d'autres.

API et ligne de commande avec des fonctionalités similaire à la commande sandbox de SELinux, mais qui utilise une techinique différente de sandboxing.

Info :

La commande sandbox de SELinux restreint et isole les applications dans un contexte de sécurité SELinux. Cette commande peut aussi utiliser optionnelement les namespace du système du fichier pour fournir un vue personnalisée de celui-ci.

Libvirt-sandbox exécute directement les applications dans une sandbox construite avec un ou plusieurs pilotes de virtualisation libvirt. L'application sera sandboxée, confinée dans un contenur LXC ou une machine virtuelle KVM. L'environement confiné pourrait avoir une configuration réseau et une vue personnalisée du système de fichier.

Libvirt-sandbox avec KVM

Récemment, KVM s'est vu octroyé la permission d'accès à l'OS host. Grâce à cette permission, il devient possible de sandboxer une application dans une machine virtuelle complète sans avoir besoin à maintenir une image additionnelle d'un OS. Le démarrage d'une application dans une sandbox commence par le chargment de initramfs et par le démarrage du noyau dans une VM. Une fois le noyau lancé, l'application à sandboxée démarre. Avec le système de fichier plan9fs, le noyau de la VM est capable de lire des zones du système de fichier du host, rendant ainsi l'installation d'un OS dans la VM non obligatoire! Pour échapper à la sandbox, les applications devront casser la sandbox guest Linux kernel, l'hyperviseur KVM, et sVirt (SELinux MCS).

Libvirt-sandbox avec LXC

Sandbox individuels à l'intérieur d'un conteneur. Pour échapper à la sandbox, les applications devront casser le conteneur LXC et la politique SELinux.

Impact sur les performances

L'impact sur les performances est perçu au démarrage de l'application dans la sandbox.

KVM : Le délai de démarrage de la VM retarde le démarrage de l'application d'environ 3 secondes. En ce qui concerne les pertes en termes de puissance CPU, elles ne sont que d'environ 10% par rapport à une application qui tourne sur le système host directement.

LXC : l'application est exécutée sur le noyaud du host dans une zone séparée. La mise en place d'un conteneur prend moins de 200 ms et l'application sandboxée tourne sans aucune perte de performance.

Étude du code source de libvirt-sandbox

Informations générales

Git actif : http://libvirt.org/git/?p=libvirt-sandbox.git;a=summary

Verison étudiée : libvirt-sandbox-0.1.1-1.fc18

Package Name libvirt-sandbox

Version 0.1.1

Release 1.fc18

Summary libvirt application sandbox framework

Description This package provides a command for running applications within a sandbox using libvirt.

Built by berrange

State complete

Started Mon, 10 Dec 2012 12:30:41 UTC

Completed Mon, 10 Dec 2012 12:39:04 UTC

Task build (f18-candidate, /libvirt-sandbox:d119dbfba6e1ae044dc56ee9092b4d87917086e0)

The following mandatory dependancies are required in order to

build libvirt-sandbox

libvirt-glib	>= 0.0.9
libvirt	>= 0.9.13
glib2	>= 2.28.0

And either the libvirt LXC or QEMU/KVM drivers.

The libvirt-sandbox library is built using GObject (GLib Object System, Ajoute la POO au language C) to enable it to be accessible to **any programming** language via GObject Introspection.

Libvirt-sandbox

GVirSandboxBuilder — Sandbox construction base class

GVirSandboxBuilderContainer — Sandbox container construction

GVirSandboxBuilderInitrd — Kernel ramdisk construction

GVirSandboxBuilderMachine — Sandbox virtual machine construction

GVirSandboxCleaner — Sandbox context cleanup tasks

GVirSandboxConfig — Basic sandbox configuration details

GVirSandboxConfigGraphical — Graphical sandbox configuration details

GVirSandboxConfigInitrd — Kernel ramdisk configuration details

GVirSandboxConfigMount — Filesystem attachment configuration details

GVirSandboxConsole — A text mode console

GVirSandboxContext — Application sandbox context

GVirSandboxContextGraphical — Desktop application sandbox context

Object Hierarchy

```
GObject
    GVirSandboxBuilder
        GVirSandboxBuilderContainer
        GVirSandboxBuilderMachine
    GVirSandboxBuilderInitrd
    GVirSandboxCleaner
    GVirSandboxConfig
        GVirSandboxConfigGraphical
        GVirSandboxConfigInteractive
        GVirSandboxConfigService
    GVirSandboxConfigInitrd
    GVirSandboxConfigMount
        GVirSandboxConfigMountFile
            GVirSandboxConfigMountGuestBind
            GVirSandboxConfigMountHostBind
            GVirSandboxConfigMountHostImage
        GVirSandboxConfigMountRam
    GVirSandboxConfigNetworkAddress
    GVirSandboxConfigNetwork
```

GVirSandboxConfigNetworkRoute

GVirSandboxConsole

GVirSandboxConsoleRaw

GVirSandboxConsoleRpc

GVirSandboxContext

GVirSandboxContextGraphical

GVirSandboxContextInteractive

GVirSandboxContextService

GBoxed

GVirSandboxConfigGraphicalSize

Compilation :

Installation dépendance :

```
yum install glib2-devel
yum install libvirt-devel
yum install libvirt-glib-devel
yum install libvirt-gobject-devel
yum install libselinux-devel
yum install gcc
## uniqement pour compilation avec git #
yum install git
yum install autoconf
yum install automake
yum install gtk-doc
```

Installation depuis l'archive source :

Télécharger le code source et l'extraire de l'archive

Ce placer dans le dossier du package libvirt-sandbox-0.1.1 qui contient le fichier INSTALL

Configuration de la compilation en mode débug et désactivé l'optimisation :

```
./configure –prefix=$HOME/libvirt-sandbox CFLAGS=-g
CXXFLAGS=-g
```

Installation avec les source depuis un git chekckout :

```
mkdir -p /root/libvirt/libvirt-sandbox
mkdir /root/libvirt/libvirt-sandbox/libvirt-sandbox-0.1.1_test
cd /root/libvirt/libvirt-sandbox
git clone git://libvirt.org/libvirt-sandbox.git
git checkout v0.1.1
./autogen.sh -prefix=/root/libvirt/libvirt-sandbox-0.1.1_test/
make
make install
```

Installation (inutile si compilé depuis le code source)

Installé :

libvirt-sandbox.x86_64 0:0.1.0-1.fc18

Dépendances installées :

glib2-devel.x86_64 0:2.34.2-2.fc18

libvirt.x86_64 0:0.10.2.3-1.fc18

libvirt-daemon-config-network.x86_64 0:0.10.2.3-1.fc18

libvirt-daemon-config-nwfilter.x86_64 0:0.10.2.3-1.fc18

libvirt-daemon-driver-libxl.x86_64 0:0.10.2.3-1.fc18

libvirt-daemon-driver-lxc.x86_64 0:0.10.2.3-1.fc18

libvirt-daemon-driver-uml.x86_64 0:0.10.2.3-1.fc18

libvirt-daemon-driver-xen.x86_64 0:0.10.2.3-1.fc18

libvirt-sandbox-libs.x86_64 0:0.1.0-1.fc18

xen-libs.x86_64 0:4.2.1-7.fc18

xen-licenses.x86_64 0:4.2.1-7.fc18

Décortication

https://www.berrange.com/tags/libvirt-gobject/

Utilise libvirt

Utilise un langage bindings libvirt-glib

(Convention GIO pour l'exécution async sans création de thread explicite)

Complexe à étudié trop d'inconnus : langage C Gobject / script bash extrêmement avancé / libvirt / libselinux /

———

Pour analyser ce code je vais partir d'un exemple d'utilisation simple de la commande virt-sandbox (sans configuration d'interface réseau, ni copie de fichier dans la sandbox, etc.)

La commande ci-dessous exécute le binaire date dans une vm QEMU/KVM qui fait office de bac à sable :

```
# ./virt-sandbox -d -v -c qemu:///session /bin/date
```

Problème actuel :

- SELinux bloque le lancement de la sandbox
- Résolution, désactivation de SELinux. Pour ne pas perdre du temps sur ce blocage, je reviendrai plus tard sur la résolution de problème (setenforce 0)
- Sandbox KVM ne se lance pas (erreur Virtio-9p Failed to initialize fs-driver with id :fsdev-fs1 and export path:/root/libvirt/libvirt-sandbox-0.1.1_test/var/run/libvirt-sandbox/sandbox/config

VIRT-SANDBOX-SERVICE

virt-sandbox-service vs libvirt-sandbox

D. Walsh à aider à la création de virt-sandbox-service, cet outil permet à un administrateur de créer un conteneur LXC facilement afin qu'il contient des applications serveur avec virt-sandbox et SELinux qui le vérouille. Cet outil à été introduit dans Fedora 17.

TEST VIRT-SANDBOX-SERVICE

La commande virt-sandbox-service ne fonctionne pas dans Fedora 18. Beaucoup de temps perdu et au final dans un ticket assigné à Daniel Berrange sur BugZilla[1], une réponse de Daniel Walsh qui indique que la résolution du problème sera faite dans le paquetage libvirt-sandbox-0.1.2-1.fc19.x86_64 sous Fedora Core 19. Pour info, la dernière version disponible sous Fedora Core 18 est la libvirt-sandbox-0.1.0-1.fc18.x86_64.

```
[root ~]# virt-sandbox-service create -C -l s0:c1,c2 -u httpd.service container1
Created sandbox container dir /var/lib/libvirt/filesystems/container1
Created sandbox config /etc/libvirt-sandbox/services/container1.sandbox
Created unit file /etc/systemd/system/httpd@container1.service
```

[1] Command virt-sandbox-service bug : https://bugzilla.redhat.com/show_bug.cgi?id=921967

```
[root ~]# virt-sandbox-service start container1
systemd 197 running in system mode. (+PAM +LIBWRAP +AUDIT +SELINUX +IMA
+SYSVINIT +LIBCRYPTSETUP +GCRYPT +ACL +XZ)
Detected virtualization 'lxc-libvirt'.
Welcome to Fedora 18 (Spherical Cow)!
Set hostname to <localhost.localdomain>.
Initializing machine ID from container UUID.
ln: impossible de supprimer
« /lib/systemd/system/anaconda.target.wants/anaconda-
tmux@tty1.service »: Système de fichiers accessible en lecture seulement
ln: impossible de supprimer
« /lib/systemd/system/anaconda.target.wants/anaconda-
shell@tty2.service »: Système de fichiers accessible en lecture
seulement
[  OK  ] Listening on Delayed Shutdown Socket.
[  OK  ] Reached target Swap.
[  OK  ] Reached target Local File Systems.
[  OK  ] Listening on Journal Socket.
         Starting Recreate Volatile Files and Directories...
[  OK  ] Started Recreate Volatile Files and Directories.
[  OK  ] Reached target System Initialization.
[  OK  ] Listening on D-Bus System Message Bus Socket.
[  OK  ] Reached target Sockets.
[  OK  ] Reached target Basic System.
         Starting The Apache HTTP Server...
         Starting Journal Service...
[  OK  ] Started Journal Service.
         Starting D-Bus System Message Bus...
[  OK  ] Started D-Bus System Message Bus.
[  OK  ] Started The Apache HTTP Server.
[  OK  ] Reached target Sandbox target.
Failed to issue method call: Unit chronyd.service is not loaded.
```

Daniel Walsh 2013-04-01 13:53:30 EDT

```
The problem here is the container was running with the svirt_t label
rather then the svirt_lxc_net_t label.

THis should be fixed in

libvirt-sandbox-0.1.2-1.fc19.x86_64
```

REFERENCES

https://www.berrange.com/posts/2012/01/17/building-application-sandboxes-with-libvirt-lxc-kvm/

http://people.redhat.com/berrange/fosdem-2012/libvirt-sandbox-fosdem-2012.pdf

http://www.h-online.com/open/news/item/Sandbox-applications-quickly-with-KVM-or-LXC-1429268.html

http://blog.bodhizazen.net/linux/lxc-linux-containers/

https://fedoraproject.org/wiki/Features/VirtSandbox