Sécurité des infrastructures de virtualisation

Conteneurs et sécurité

3e année cycle ingénieur STI, option 2SU, 2018 – 2019

https://tim.siosm.fr/cours

Objectifs

- Découvrir Docker et le confinement avec SELinux sous Fedora ;
- Créer un conteneur à l'aide d'un Dockerfile.

Important : Garder une trace des différentes commandes utilisée et des configurations produites.

1 Récupération d'une image Docker

Pour récupérer directement du registre public Docker l'image officiel de conteneur Fedora, utiliser :

```
$ sudo docker pull fedora
```

Vérifier qu'elle a bien été importée avec la commande :

```
$ sudo docker images
```

La dernière version publiée de l'image est téléchargée. Pour lancer un shell interactif dans un conteneur utilisant cette image :

```
$ sudo docker run -it fedora bash
```

2 Docker & SELinux

Toutes les manipulations qui vont suivre sont à effectuer dans deux machines virtuelles en parallèle (une avec SELinux, une avec SELinux désactivé) pour pouvoir comprendre l'impact de SELinux sur l'isolation des conteneurs avec Docker.

S'aider des commandes suivantes pour répondre aux questions :

```
$ id -Z
$ sudo docker ps
$ systemd-cgls
$ ps afuxZ
```

- **Question 1 :** Sous quel utilisateur tourne ce shell ?
- **Question 2 :** Est-ce le même à l'intérieur et à l'extérieur du conteneur ?
- **Question 3 :** Sous quel contexte SELinux et avec quelles catégories tourne ce processus ?

Lancer un deuxième shell dans un deuxième conteneur en parallèle.

- **Question 4 :** Sous quel utilisateur, contexte SELinux et avec quelles catégories tourne ce processus ?

Lancer un nouveau shell avec la commande :

```
$ sudo docker run -it --rm --privileged -v /:/host fedora bash
```

- **Question 5 :** Sous quel utilisateur, contexte SELinux et avec quelles catégories tourne ce processus ?
- **Question 6 :** Quels sont les risques présentés par cette commande ?

S'inspirer de la commande suivante pour obtenir des informations sur les accès autorisés par la politique SELinux pour le domaine `container_runtime_t` :

```
$ sesearch -A -s container_runtime_t /etc/selinux/targeted/policy/policy.29
```

3 Dockerfile

Récupérer un exemple de Dockerfile à partir du projet https://github.com/fedora-cloud/Fedora-Dockerfiles :

```
$ mkdir docker
$ cd docker
$ curl -O https://raw.githubusercontent.com/fedora-cloud/Fedora-Dockerfiles/master/apache/run-apache.sh
$ curl -O https://raw.githubusercontent.com/fedora-cloud/Fedora-Dockerfiles/master/apache/Dockerfile
```

Construire un conteneur à partir de ce Dockerfile :

```
$ sudo docker build --tag httpd .
```

Lancer le conteneur créé :

```
$ sudo docker run --publish 80 httpd
...
```

Ou pour le lancer en tache de fond :

```
$ sudo docker run --detach=true --publish 80 httpd
19f063bfff5aa37655a...
```

Vérifier que l'on a bien accés au serveur apache :

```
$ sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' 19f063bfff5aa37655a...
172.17.0.2
$ curl 172.17.0.2:80
```

Modifier le Dockerfile pour obtenir une image conteneur qui :
- est basé sur l'image fournie par le projet Fedora ;
- ne tourne pas sous l'utilisateur `root` ;
- lance le serveur Apache sur le port 8080 ;
- l'image du conteneur est en lecture seule (RO) et les données (logs, etc.) sont stockées dans des volumes persistents.

S'aider des guides suivants :
- la référence sur les `Dockerfiles` : https://docs.docker.com/engine/reference/builder/
- les recommandations de l'article : https://www.projectatomic.io/docs/docker-image-author-guidance/
- le guide de bonnes pratiques : https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/

Références

- http://www.projectatomic.io/blog/2015/08/why-we-dont-let-non-root-users-run-docker-in-centos-fedora-or-rhel/
- http://www.projectatomic.io/docs/docker-image-author-guidance/
- http://developerblog.redhat.com/2014/11/06/introducing-a-super-privileged-container-concept/
- https://opensource.com/article/18/3/just-say-no-root-containers
- registry.fedoraproject.org