Sécurité des infrastructures de virtualisation

Conteneurs et sécurité - Solutions

3e année cycle ingénieur STI, option 2SU, 2017 – 2018

- **Question 1 :** Sous quel utilisateur tourne ce shell ?
- **Réponse :** La commande id lancée dans le conteneur nous indique que ce shell tourne en root :

```
[fedora@fedora-dev ~]# docker run -it --rm fedora bash
[root@644ee121dac7 /]# id
uid=0(root) gid=0(root) groups=0(root)
```

- **Question 2 :** Est-ce le même à l'intérieur et à l'extérieur du conteneur ?
- **Réponse :** En cherchant l'identifiant du conteneur dans la sortie de la commande `ps auxf`, on peut trouver les informations sur le processus `bash` s'exécutant dans le conteneur :

```
# ps auxf | less
root 9693 0.0 0.4 287592 9244 ? Ssl 13:31 0:00 /usr/libexec/docker/docker-containerd-current --listen ↙
    ↳ unix:///run/containerd.sock --shim /usr/libexec/docker/docker-containerd-shim-current -- ↙
    ↳ start-timeout 2m
root 10036 0.0 0.1 190208 2952 ? Sl 13:34 0:00 \_ /usr/libexec/docker/docker-containerd-shim-current ↙
    ↳  644ee121dac78d572f7bbd2064cf212d0a01949946b6de3dbf77e7cd5898d61b /var/run/docker/ ↙
    ↳ libcontainerd/644ee121dac78d572f7bbd2064cf212d0a01949946b6de3dbf77e7cd5898d61b /usr/ ↙
    ↳ libexec/docker/docker-runc-current
root 10050 0.0 0.1 12588 3800 pts/1 Ss+ 13:34 0:00 \_ bash
```

L'utilisateur sous lequel tourne le shell est aussi root lorsqu'on l'observe depuis l'extérieur du conteneur.

- **Question 3 :** Sous quel contexte SELinux et avec quelles catégories tourne ce processus ?
- **Réponse :** A l'intérieur du conteneur, les processus ont l'impression que le système fonctionne sans SELinux. Vu de l'extérieur, on peut trouver un contexte de la forme suivante :

```
# ps axfZ | less
system_u:system_r:container_runtime_t:s0 9693 ? Ssl 0:00 /usr/libexec/docker/docker-containerd- ↙
    ↳ current --listen unix:///run/containerd.sock --shim /usr/libexec/docker/docker-containerd- ↙
    ↳ shim-current --start-timeout 2m
system_u:system_r:container_runtime_t:s0 10036 ? Sl 0:00 \_ /usr/libexec/docker/docker- ↙
    ↳ containerd-shim-current 644 ↙
    ↳ ee121dac78d572f7bbd2064cf212d0a01949946b6de3dbf77e7cd5898d61b /var/run/docker/ ↙
    ↳ libcontainerd/644ee121dac78d572f7bbd2064cf212d0a01949946b6de3dbf77e7cd5898d61b /usr/ ↙
    ↳ libexec/docker/docker-runc-current
system_u:system_r:container_t:s0:c212,c687 10050 pts/1 Ss+ 0:00 \_ bash
```

Les processus à l'intérieur du conteneur tournent donc sous le contexte : `system_u:system_r:container_t:s0:c212,c687`.

- **Question 4 :** Sous quel utilisateur, contexte SELinux et avec quelles catégories tourne ce processus ?
- **Réponse :** Il tourne sous le même utilisateur et le même contexte SELinux mais il ne partage pas les même catégories SELinux. Les processus de conteneurs différents ne peuvent donc pas interagir entre eux.

- **Question 5 :** Sous quel utilisateur, contexte SELinux et avec quelles catégories tourne ce processus ?
- **Réponse :** De façon similaire aux questions précédentes, on peut déterminer que les processus du conteneur tournent sous le contexte : `system_u:system_r:container_runtime_t:s0`.
- **Question 6 :** Quels sont les risques présentés par cette commande ?
- **Réponse :** Cette commande met à disposition des programmes à l'intérieur du conteneur l'ensemble du système de fichier de l'hôte. Les processus à l'intérieur du conteneur peuvent alors interagir avec le système hôte, ce qui pose des problèmes de sécurité.
- **Question Dockerfile :** Créer un conteneur Docker qui :
  — est basé sur l'image de confiance fournie par le projet Fedora ;
  — ne tourne pas sous l'utilisateur `root` ;
  — lance un serveur Apache sur le port 8080.
- **Réponse :** Suggestion de Dockerfile :

```
FROM fedora
MAINTAINER Timothée Ravier

# Mise à jour de l'ensemble de l'image et installation de Apache
RUN dnf -y update ; dnf -y install httpd ; dnf clean all

# Modification de la configuration de Apache pour qu'il ne soit plus nécessaire
# de l'exécuter en root puisque le port 8080 est non privilégié
RUN sed -i 's|Listen 80|Listen 8080|g' /etc/httpd/conf/httpd.conf

# Export du port 8080 à l'extérieur du conteneur
EXPOSE 8080

# Il n'est plus nécessaire de changer d'utilisateur
RUN sed -i 's|User apache|#User apache|g' /etc/httpd/conf/httpd.conf
RUN sed -i 's|Group apache|#Group apache|g' /etc/httpd/conf/httpd.conf

# L'installation du paquet httpd a créé l'utilisateur et le groupe apache que
# nous pouvons utiliser comme utilisateur non privilégié. Sinon, nous aurions
# pu créer un utilisateur à l'aide de la commande :
# RUN useradd -d /var/lib/http -M -r -s /sbin/nologin http

# Il n'est plus nécessaire d'être root pour ouvrir les fichiers de log ou pour
# accéder au dossier /run/httpd
RUN chown -R apache:apache /var/log/httpd /run/httpd

# Les commandes qui suivent seront désormais lancées en tant qu'utilisateur
# apache
USER apache
CMD /usr/sbin/httpd -e info -DFOREGROUND
```

Construction d'une image à partir de ce Dockerfile :

```
# ls
Dockerfile

# Docker build .
Sending build context to Docker daemon 3.072 kB
Step 1 : FROM fedora
```

```
 ---> f5c7d31d33dd
Step 2 : MAINTAINER Timothée Ravier
 ---> Using cache
 ---> be56ca58cb93
Step 3 : RUN dnf -y update ; dnf -y install httpd ; dnf clean all
 ---> Using cache
 ---> b8755c0e24cc
Step 4 : RUN sed -i 's|Listen 80|Listen 8080|g' /etc/httpd/conf/httpd.conf
 ---> Using cache
 ---> 845dbe1fc35a
Step 5 : EXPOSE 8080
 ---> Using cache
 ---> 429bc5ffa123
Step 6 : RUN sed -i 's|User apache|#User apache|g' /etc/httpd/conf/httpd.conf
 ---> Using cache
 ---> 97ba7aa0d5b3
Step 7 : RUN sed -i 's|Group apache|#Group apache|g' /etc/httpd/conf/httpd.conf
 ---> Using cache
 ---> 171cf8174f6c
Step 8 : RUN chown -R apache:apache /var/log/httpd /run/httpd
 ---> Using cache
 ---> e2118551ecd1
Step 9 : USER apache
 ---> Using cache
 ---> 585354f7726c
Step 10 : CMD /usr/sbin/httpd -e info -DFOREGROUND
 ---> Using cache
 ---> 22053762c59c
Successfully built 22053762c59c

# docker tag 22053762c59c apache

# docker run -d -p 8080 apache
64a5160e29e0fc2db13942b069975c32397b7c7f010b91543b37de61e0ccf330

# curl 171.17.0.2:8080
...
```