Service

- Docker pivilege

Docker pivilege

Bonjour, à tous nous nous retrouvons aujourd'hui pour un nouveau tutoriel concernant l'escalade de privilège avec Docker. Vous l'ignorez peut-être, mais Docker souffre d'une faille de sécurité si d'autres personnes que l'user root sont autorisées à utiliser cette commande. C'est pour ça qu'il faut garder à l'esprit d'utiliser Docker dans le respect des bonnes pratiques. Je vous laisse le soins de parcourir ce petit guide □□

Recommandations de sécurité relatives aux déploiements de conteneurs docker :

https://www.ssi.gouv.fr/uploads/2020/12/docker_fiche_technique.pdf

Sommaire

I. Envirronement de mise en place

- Ubuntu 20.04.x – Accès à internet, avec le service SSH (pas obligatoire si-vous utilisez Ubuntu avec une GUI). Toutes les commandes ci-dessous seront exécutées via l'utilisateur root de de cette machine

II. Installation de docker, et scénario d'attaque

Si-vous ne savez pas comment installer docker, je vous laisse consulter le lien suivant : https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04-fr

Dès lors que docker est installé, imaginez le contexte suivant (déjà vécu pour ma part)

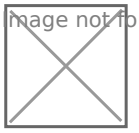
Vous avez installé docker sur un serveur, pour que des développeurs puisse intéragir avec des conteneurs et réaliser différentes actions de test/build sur leur application. Afin d'éviter de donner le mot de passe root du système à tout le monde, vous décidez d'ajouter chaque utilisateur concerné à un groupe docker qui aura la possibilité de pouvoir utiliser la commande `docker` *en ce passant de sudo.*

Dans mon cas, je vais créer l'utilisateur suivant

```
adduser tintin
```

Puis, je vais ajouter tintin au groupe suivant : *(Le groupe n'est pas forcément obligé d'être créé en amont, car sur Linux dès qu'un groupe est affilié à un utilisateur, si celui-ci n'existe pas, il est automatiquement créé.)*

```
newgrp docker
usermod -G docker tintin
```



Capture d'écran reflétant les actions efféctuées précédemment

Créez un fichier flag, dans le dossier /root/ via l'utilisateur root avec la commande suivante :

```
echo "you are into the /root/ folder of your docker host !!! congretulations" >> /root/flag
```

Imaginez qu'à la suite d'un brute force de mot de passe sur le service SSH, vous arrivez à trouver les credentials du compte tintin. À partir de ce moment, vous pouvez commencer à chercher si le compte tintin dispose de privilège, ou à accès à des commandes qui pourraient nous permettre de réaliser notre privesc. ***Ce scénario est totalement fictif, mais possible, et déjà rencontré à titre personnel. Il vous permettra de mieux conceptualiser cette attaque de bout en bout****. Ici la privesc est le sésame qui va permettre de devenir root sur l'hôte docker.*

Par la suite, connectez vous avec le compte Tintin d'une des manières suivantes :

- GUI
- SSH

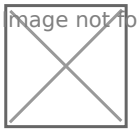
Si-vous n'avez pas de GUI, connectez vous en SSH à votre hôte local de la manière suivante :

```
ssh tintin@127.0.0.1
```

Dans mon cas, je vais utiliser la GUI de ma machine virtuelle donc je me déconnecte, puis je me connecte avec l'utilisateur tintin nouvellement créé, puis ouvrez un terminal.

III. Exploitation de la faille

Comme vous pouvez le constater ci-dessous, **depuis mon compte Tintin, je ne dispose pas de privilège administrateur, j'ai simplement la possibilité d'exécuter les commandes « standard » ainsi que le lot de commandes docker.**



flag, est un fichier que j'ai créé dans le dossier /root/ à titre d'exemple pour ce tutoriel (depuis le compte root)

Je teste si j'ai accès à la commande `docker version`

Bingo ! Notre utilisateur tintin peut accéder à la commande `docker` Comme prévu ^^ je peux donc réaliser ma privesc.

Afin de bypasser cette restriction, je vais dans un premier temps utiliser la commande ci-dessous qui va permettre de pouvoir **afficher le contenu du dossier /root de mon hôte docker**, par l'intermédiaire d'un conteneur se basant sur une image alpine. *(De base je ne peux pas naviguer à l'intérieur du dossier super-utilisateur root (cf capture un peu plus haut))*.

Le contenu du dossier /root/ de l'hôte docker sera monté dans le dossier du conteneur /mnt/

```
docker run -v /root:/mnt -it alpine
```

Cela est déjà pas mal, **mais vous êtes sûrement entrain de vous dire une chose…** Vous n'avez pas le contrôle total de votre machine hôte docker ici. Vous pouvez simplement explorer les fichiers du dossier de root. Heureusement (ou pas ^^), en modifiant légèrement notre précédente commande, vous allez pouvoir prendre le contrôle « total » de votre hôte docker, afin d'obtenir un shell root avec une vue « racine » du système hôte.

Sortez du conteneur.

```
exit
```

Exécutez la commande suivante :

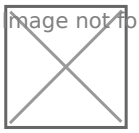
```
docker run -it -v /:/mnt alpine chroot /mnt
```

Comme vous pouvez le constater, j'ai la vue racine du système et je peux exécuter n'importe quelle commande n'importe où. À titre d'exemple, je change le mot de passe du compte user, via l'instruction `passwd`.

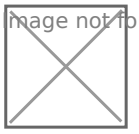
Vous avez pris le contrôle de votre machine hôte docker !

Ci-dessous, j'exécute un ensemble de commandes sur le système hôte depuis le conteneur, qui nécessite d'être root :

```
docker ps # Je trouve plutôt fun de vérifier que mon conteneur fonctionne bien depuis mon conteneur :)
cat /etc/passwd
cat /etc/shadow
cat /etc/sudoers
```



Vous pouvez faire ce que vous voulez, vous avez accès à tous les dossiers que vous souhaitez. **Vous êtes le capitaine du navire** ^^ (ce qui n'était pas le cas avant)



J'espère que cet article vous aura fait prendre conscience qu'il ne faut pas ajouter d'autres users que root (par défaut) au groupe docker, car cela peut représenter une faille de sécurité réelle.

++

Geoffrey