Sécurité des infrastructures de virtualisation

Évaluation

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

Introduction

Cette évaluation a été préparée dans une machine virtuelle Fedora 29, similaire à celles utilisées en TD.

Pour en préparer une de zéro, vous pouvez utiliser les commandes suivantes :

```
$ sudo dnf update -y
$ sudo dnf install -y vim tree bash-completion docker selinux-policy-devel setools-console
$ sudo systemctl enable --now docker
$ sudo reboot
```

Les sources des fichiers `httpserver.py`, `simpleserver.py` et `simpleserver.service` sont fournies dans une archive transmise avec cette évaluation.

Vous devez rendre une archive contenant les solutions des exercices que vous aurez impérativement commentés pour expliquer vos choix et résultats. Vous êtes priés de m'envoyer vos réponses sous forme de texte brut, Markdown ou PDF uniquement. Je ne souhaite pas recevoir de fichier Word, ODT, etc.

1 Service systemd

- **Question 1 :** Créer une unit systemd pour le script Python `httpserver.py`.
  Durcir la configuration de l'unit pour que le service :
  — soit lancé sous un utilisateur système non privilégié ;
  — dispose de son propre dossier /tmp et /dev ;
  — ne puisse pas modifier le contenu de /usr et de /etc ;
  — ne puisse pas utiliser d'appels système obsolètes, liés au debug ou privilégiés ;
  — ne puisse pas utiliser plus de 200Mo de mémoire.
- **Question 2 :** Proposer des commandes pour valider le bon fonctionnement d'au moins deux des mesures de durcissement mises en place.

2 Module SELinux

- **Question 3 :** Écrire un module SELinux pour confiner le script python `simpleserver.py`, qui sera lancé par l'unit systemd `simpleserver.service`. N'oubliez pas de restorer les contextes par défaut pour ces deux fichiers une fois créés dans la machine virtuelle avec la commande :

  ```
  $ restorecon -FRv /etc/systemd/system/simpleserver.service /usr/local/bin/simpleserver.service
  ```

  Ce script doit pouvoir créer des dossiers et fichiers dans le répertoire /var/www/html.
  Les commandes suivantes doivent fonctionner sans erreurs :

  ```
  $ curl -d "Some POST data" localhost:7070/var/www/html/test/toto/foo
  $ curl localhost:7070/var/www/html/test/toto/foo
  ```

  Les commandes suivantes ne doivent pas fonctionner :

  ```
  $ curl -d "Some POST data" localhost:7070/etc/ssl/test
  $ curl localhost:7070/etc/ssl/test
  $ curl localhost:7070/etc/shadow
  ```

3 Conteneur Docker

- **Question 4 :** Créer un conteneur qui lance le script Python `httpserver.py`.
  Ce conteneur :
  — sera basé sur l'image officielle Fedora mise à jour ;
  — utilisera la variable d'environnement pour changer le port d'écoute du script ;
  — exécutera le script sous un utilisateur non privilégié ;
  — aura son système de fichier en lecture seule et utilisera un volume ou point de montage pour partager des données avec l'hôte.
- **Question 5 :** Donner un exemple de commande utilisé pour lancer le conteneur en utilisant la variable d'environnement, un volume ou point de montage et en rendant le port accessible depuis l'hôte.