Sécurité des infrastructures de virtualisation

systemd, journald, cgroups et durcissement système - Solutions

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

- **Question 4 :** Comment peut on afficher tous les logs qui ont été émis par le service Apache (`httpd`) entre 10h30 et 11h00 ce matin ?
- **Réponse :** La commande suivante réponds à la question :

```
$ journalctl -u httpd --since 10:30 --until 11:00
```

- **Question 5 :** Que fait la commande suivante ?
- **Réponse :** La commande se place en écoute des connexions TCP entrantes sur le port 9999, sur toutes les interfaces. Pour chaque nouvelle connexion, un processus fils est créé et les entrées/sorties associées à cette connexion sont redirigées vers les entrées/sorties standards de ce nouveau processus.
- **Question 6 :** Où allons nous placer notre fichier d'unit ?
- **Réponse :** Tous les fichiers d'unit créés par un administrateur doivent être placés dans le dossier `/etc/systemd/system`.
- **Question 7 :** Créer un `service unit` qui lance la commande précédente.
- **Réponse :** Exemple le plus simple :

```
[Unit]
Description=Exemple de test de systemd
# Optionnel : Indique qu'il est nécessaire que le réseau soit démarré avant que
# notre service soit lancé
After=network-online.target
Wants=network-online.target

[Service]
# Note : Il n'est pas nécessaire de préciser le type puisque celui qui
# conrrespond à notre unit est "Simple" et c'est le type par défaut.
ExecStart=/usr/bin/nc -lke /bin/bash 0.0.0.0 9999

[Install]
# Optionnel : Indique que le service est une dépendance la target multi-user.
# Il sera lancé au démarrage comme un service classique.
WantedBy=multi-user.target
```

- **Question 8 :** Confiner maintenant ce service en définissant :
  - ⋆ un utilisateur et un groupe à choisir judicieusement (à créer si nécessaire) ;
  - ⋆ des limitations d'accès sur les répertoires du système (`ProtectSystem, etc.`) ;
  - ⋆ des limitations sur les appels systèmes disponibles :
    - — liste noire pour supprimer les fonctions de debug ou les appels systèmes obsolètes ;
    - — ou liste blanche avec uniquement les appels systèmes nécessaires.
  - ⋆ le redémarrage automatique en cas d'erreur non prévue.

- **Réponse :** Suggestion : Nous allons créer un utilisateur et un groupe système sans privilèges pour notre service :

```
# useradd -d / -M -r -s /sbin/nologin netcat
```

```
[Unit]
Description=Exemple de test de systemd
After=network-online.target
Wants=network-online.target

[Service]
User=netcat
Group=netcat

ExecStart=/usr/bin/nc -lke /bin/bash 0.0.0.0 9999

# Il n'aura pas accès au contenu de /etc
InaccessiblePaths=/etc

# Il ne sera pas possible de créer des fichiers spéciaux, par exemple des tubes nommés
SystemCallFilter=~mknod

# Alternative : le filtre suivant correspond à une bonne blacklist minimale
SystemCallFilter=~@debug @obsolete

# Alternative : un début de liste blanche, très stricte (à compléter)
SystemCallFilter=@basic-io @file-system @network-io @resources @process @io-event @ipc
SystemCallFilter=brk mprotect set_tid_address set_robust_list rt_sigaction rt_sigprocmask futex

# Redémarrage en cas de plantage
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

- **Question 9 :** Vérifier que toutes les options fonctionnent correctement.
- **Réponse :** Les commandes suivantes permettent de tester l'efficacité des mesures de protections misent en place à la question précédente :

```
$ echo 'id' | nc localhost 9999
uid=992 gid=992 groups=992

$ echo 'ls /etc' | nc localhost 9999
<vide>

$ echo 'ls -l /' | nc localhost 9999 | grep etc
d---------. 2 0 0 40 Feb 4 12:50 etc

$ echo 'mkfifo /tmp/test' | nc localhost 9999

$ echo 'ls /tmp' | nc localhost 9999
<vide>

$ sudo systemctl kill --signal=SIGSEGV test.service

$ sudo systemctl status test.service
```