Sécurité des infrastructures de virtualisation

systemd, journald, cgroups et durcissement système

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

Objectifs

- Découvrir systemd et journald ;
- Gérer des processus avec les cgroups ;
- Durcir la configuration d'un système à l'aide de systemd.

Introduction et recommandations

La version PDF de ce TD est disponible à l'adresse suivante : https://tim.siosm.fr/cours.

N'hésitez pas à garder une trace des différentes commandes que vous utilisez et des configurations que vous allez produire.

Ce TD est prévu pour être réalisé dans une machine virtuelle Fedora 27 ou supérieure.

Pour préparer une machine virtuelle en partant de zéro :

```
$ sudo dnf update -y
$ sudo dnf install -y vim tree bash-completion
$ sudo reboot
```

1 systemd et journald

1.1 État des services

Lister l'état des services sur la machine :

```
# systemctl
# systemctl status
```

Inspecter l'état précis de quelques services :

```
# systemctl status <service>
```

Tuer un service au hasard avec la commande `kill` et constater les changements de status. Il est possible de filtrer les résultats en n'affichant que les services disponibles (ou lancés ou en échec) :

```
# systemctl --state=running
# systemctl --state=active
# systemctl --state=exited
# systemctl --state=failed
```

Désactiver et arrêter tous les services dont on n'a pas besoin :

```
# systemctl disable <service>
# systemctl stop <service>
```

1.2 Démarrage du système

Vérifier que la target par défaut est bien multi-user.target :

```
# systemctl get-default <target>
# systemctl set-default <target>
```

Étudier la procédure de démarrage du système avec :

```
# systemd-analyze plot > bootchart.svg
```

Le fichier `bootchart.svg` peut être affiché dans un navigateur.

- **Question 1 :** Dans quel ordre démarrent les services ?
- **Question 2 :** Trouver quatre services avec des dépendances et retrouvez cette information dans leur fichier d'unit.

1.3 Modifier certains paramètres pour un service précis

Modifier quelques services sans modifier directement leur unit en utilisant :

```
# systemctl edit <service>
```

Constater les changements dans le dossier :

```
/etc/systemd/system/<service>.service.d/*..conf
```

Indiquer à systemd de recharger la configuration, redémarrer le service et vérifier que la modification a bien été appliquée :

```
# systemctl daemon-reload
# systemctl restart <service>(.service)
# systemctl show service
```

- **Question 3 :** Indiquer qu'un service devra être relancé en cas d'interruption imprévue (chercher l'option Restart dans systemd.service(5)). Tester que cela fonctionne correctement.

Modifier quelques services en copiant leur fichier d'unit par défaut dans le dossier :

```
$ cp /usr/lib/systemd/system/<unit>.service /etc/systemd/system/
```

Les modifications peuvent alors être effectuée directement dans l'unit. Attention, les modification dans le dossier `*.service.d/*.conf` sont toujours prises en compte. Pour afficher l'ensemble des configurations prises en compte pour un service par systemd :

```
# systemctl cat <service>
```

1.4 Naviguer dans le journal

Essayer et expliquer les commandes suivantes :

```
journalctl -e
journalctl -b -1
journalctl -b -0 _PID=333
journalctl -k
journalctl -u nginx
journalctl -f -n 50
journalctl /usr/bin/dbus-daemon
```

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

1.5 Créer son propre service unit

Nous allons prendre l'utilitaire `nc` et en faire un démon géré par `systemd`.

- **Question 5 :** Que fait la commande suivante ?

```
# nc -lke /bin/bash 0.0.0.0 9999
```

- **Question 6 :** Où allons nous placer notre fichier d'unit ?
- **Question 7 :** Créer un service unit qui lance la commande précédente.

Ne pas oublier que systemd ne prend pas en compte les nouveaux fichiers d'unit automatiquement. Pour lui indiquer de recharger la configuration :

```
# systemctl daemon-reload
```

Tester le bon fonctionnement de ce nouveau service

```
$ nc localhost 9999
ls
...
```

- **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.
- **Question 9 :** Vérifier que toutes les options fonctionnent correctement.

2 Gestion avec les cgroups

2.1 Gestion manuelle (cgroups v1)

Le `PID controller` permet de limiter le nombre de PID disponibles dans un cgroup. Pour utiliser ce `controller`, il faut utiliser la commande `mount` :

```
# mount -t cgroup -o pids none /sys/fs/cgroups/pids
```

Dans la plupart des cas, systemd aura déjà monté cette hiérarchie au démarrage du système :

```
# mount | grep cgroup
```

Créer un nouveau groupe dans la hiérarchie du `PID controller` et y déplacer le `shell` courant :

```
# mkdir /sys/fs/cgroup/pids/toto
# echo $$
# echo $$ > /sys/fs/cgroup/pids/toto/cgroup.procs
```

- **Question 10 :** Expliquer le résultat de la commande suivante :

```
# cat /sys/fs/cgroup/pids/cgroup.procs
24150
27463
```

Mettre en place et tester une limite à l'aide du `controller` :

```
# echo 20 > /sys/fs/cgroup/pids/toto/pids.max
# for i in $(seq 1 20); do sleep 20 & done
```

Avant de supprimer un groupe, il faut que tous les processus qui lui sont associés se terminent. La suppression du dossier supprimera ensuite le groupe.

2.2 cgroups et systemd

systemd et configure et exploite une partie des cgroups disponibles sur un système. Pour lister l'organisation déjà crée par systemd :

```
# systemd-cgls
```

- **Question 11 :** Que constatez vous sur l'organisation des services avec les cgroups ?

Les variables (liées aux `cgroups`) modifiables à la volée sont listées dans systemd.resource-control(5).

Pour réduire dynamiquement et de façon permanente la valeur de CPUShares d'un service :

```
# systemctl set-property <service> CPUShares=512
```

- **Question 12 :** Retrouver les modifications effectuées par `systemd` dans l'arbre des `cgroups` (`/sys/fs/cgroup/`).

Pour effectuer une modification temporaire qui persistera uniquement jusqu'au prochain reboot :

```
# systemctl --runtime set-property ...
```

Activer l'accounting pour quelques services :

```
# systemctl set-property <service> CPUAccounting=true MemoryAccounting=true ...
```

Surveiller l'utilisation des ressources par ces services avec :

```
# systemd-cgtop
```

3 Références

- Wiki de systemd sur FreeDesktop.org : http://freedesktop.org/wiki/Software/systemd/
- Pages de man de systemd : http://www.freedesktop.org/software/systemd/man/
- The New Control Group Interfaces : http://www.freedesktop.org/wiki/Software/systemd/ControlGroupInterface/
- Red Hat Summit 2013 - Getting Ready for Systemd : https://access.redhat.com/site/videos/403833
- Durcissement système à l'aide de systemd, Timothée Ravier, SSTIC 2017 : https://www.sstic.org/2017/presentation/durcissement_systeme_avec_systemd/