Sécurité des infrastructures de virtualisation

SELinux

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

Objectifs

- Découvrir SELinux ;
- Gestion des contextes et des booléens ;
- Confinement des utilisateurs ;
- Ecriture d'un module pour un démon système.

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 httpd bash-completion selinux-policy-devel setools-console
$ sudo systemctl enable --now httpd
$ sudo reboot
```

1 État de SELinux

Afficher l'état de SELinux et de la politique actuellement chargée :

```
$ sestatus -v
$ cat /etc/selinux/config
$ getenforce
$ seinfo
```

2 Contextes fichiers et processus

- **Question 1:** Sous quel contexte l'utilisateur fedora s'exécute-il ?
- **Question 2:** Quel sont les contextes associés aux fichiers ou répoertoires suivants :
  - /usr/bin
  - /etc/shadow
  - /etc/password
  - /home
  - /home/fedora
  - /root
  - /var/www

— /tmp
- **Question 3:** Quel sont les contextes associés aux processus suivants :
  — démon sshd ;
  — démon httpd ;
  — init.

3 Gestion des booléens

Afficher l'état des booléens

```
$ getsebool -a
```

— Créer le dossier `/home/fedora/public_html` et un ficher dans ce dossier ;
— Donner le droit à tout le monde de traverser le dossier `/home/fedora` ;
— Commenter la directive `UserDir disable` et décommenter la directive `UserDir public_html` dans la configuration d'Apache (`/etc/httpd/conf.d/userdir.conf`) ;
— Redémarrer Apache.

Essayer de récupérer le contenu du fichier précedement créé avec la commande `curl` :

```
$ curl http://localhost/~fedora/test
```

Ne pas oublier de consulter les logs pour obtenir des informations :

```
$ sudo journalctl -e
$ sudo tail /var/log/httpd/error_log
$ sudo tail /var/log/audit/audit.log
```

Chercher si une règle n'existe pas déjà dans la politique pour autoriser cet accès :

```
$ ls -alZ ~/public_html/
$ sesearch -v --allow -s httpd_t -t httpd_user_content_t
```

Afficher toutes les règles ajoutées par ce booléen :

```
$ sesearch --allow -b httpd_enable_homedirs
```

Activer le booléen pour autoriser Apache à lire le contenu du dossier public_html dans les répertoires utilisateurs :

```
$ sudo setsebool httpd_enable_homedirs on
$ curl http://localhost/~fedora/test
```

4 Confinement d'un utilisateur

Ajouter un utilisateur toto et lui associer un utilisateur SELinux confiné.

```
$ sudo adduser toto
$ sudo passwd toto
$ sudo semanage user --list
$ sudo semanage login --list
$ sudo semanage login --add -s user_u -r s0 toto
```

Se loguer sous cet utilisateur (à distance avec SSH, pas avec `su`).

- **Question 4:** Sous quel contexte tourne le shell obtenu ?
- **Question 5:** Essayer de passer `root`. Que se passe-t-il ?

5 Écrire un module pour un démon système

— Importer le script python `pyserver.py` dans le dossier `/usr/local/bin/`
— Importer l'unit systemd `pyserver.service` dans le dossier `/etc/systemd/system/`
— Recharger la configuration de `systemd` :

```
$ sudo systemctl --daemon-relad
```

— Récupérer les sources de la politique et les recompiler avec :

```
$ dnf download --source selinux-policy
$ rpm --install selinux-policy-3.13.1-283.19.fc27.src.rpm 2>/dev/null # erreurs non importantes
$ sudo dnf install -y rpmdevtools
$ sudo dnf builddep -y selinux-policy
$ sudo dnf install -y @development-tools
$ cd ~/rpmbuild/SPECS
$ rpmbuild -bp selinux-policy.spec # erreurs non importantes
$ cd ~/rpmbuild/BUILD/serefpolicy-3.13.1
```

— Installer le démon `setroubleshoot` (à ne pas utiliser en production) pour s'aider à comprendre les erreurs SELinux :

```
$ sudo dnf install -y setroubleshoot-server
```

Utiliser sepolgen pour générer un caneva de politique SELinux pour notre démon :

```
$ sepolgen --init /usr/local/bin/pyserver # erreurs non importantes
```

Lancer le démon avec systemd et mettre à jour la politique en s'aidant du script `pyserver.sh`. Penser à désactiver les régles `dontaudit` pour obtenir tous les messages d'erreurs :

```
$ sudo semodule -DB
$ sudo ./pyserver.sh
$ sudo systemctl start pyserver
$ curl http://localhost:8000/
$ sudo journalct -e -u audit
$ sudo tail -fn 10 /var/log/audit/audit.log
$ sudo ./pyserver.sh --update
$ vim pyserver.te
```

Une fois la politique complétée, il faut supprimer la directive rendant le domaine permissif pour rendre effective la protection avec SELinux.

6 Références

- SELinux project Wiki : https://selinuxproject.org/page/Main_Page
- SELinux Notebook : http://freecomputerbooks.com/The-SELinux-Notebook-The-Foundations.html
- BLog de Dan Walsh : http://danwalsh.livejournal.com/