Sécurité des infrastructures de virtualisation

SELinux - Solutions

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

- **Question 1 :** Sous quel contexte l'utilisateur fedora s'exécute-il ?
- **Réponse :** `unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023`

- **Question 2 :** Quel sont les contextes associés aux fichiers ou répoertoires suivants :
  — /usr/bin : system_u :object_r :bin_t :s0
  — /etc/shadow : system_u :object_r :shadow_t :s0
  — /etc/password : system_u :object_r :passwd_file_t :s0
  — /home : system_u :object_r :home_root_t :s0
  — /home/fedora : unconfined_u :object_r :user_home_dir_t :s0
  — /root : system_u :object_r :admin_home_t :s0
  — /var/www : system_u :object_r :httpd_sys_content_t :s0
  — /tmp : system_u :object_r :tmp_t :s0

- **Question 3 :** Quel sont les contextes associés aux processus suivants :
  — démon sshd : system_u :system_r :sshd_t :s0-s0 :c0.c1023
  — démon httpd : system_u :system_r :httpd_t :s0
  — init : system_u :system_r :init_t :s0

- **Question 4 :** Sous quel contexte tourne le shell obtenu ?
- **Réponse :** user_u :user_r :user_t :s0

- **Question 5 :** Essayer de passer root. Que se passe-t-il ?
- **Réponse :** Les commandes sudo et su refusent de s'exécuter ou ne permettent pas de passer root même si le mot de passe est connu.

- **Question 6 :** Section 5 : Politique SELinux pour pyserver.py
- **Réponse :** Exemple : Autorise uniquement l'accès au répertoire /tmp et /var/www/html : pyserver.te :

```
policy_module(pyserver, 1.0.0)

require {
        type bin_t;
        type node_t;
        type tmp_t;
        type passwd_file_t;
        type soundd_port_t;
        type httpd_sys_content_t;
};

########################################
#
# Declarations
#

type pyserver_t;
```

```
type pyserver_exec_t;
init_daemon_domain(pyserver_t, pyserver_exec_t)

# A supprimer une fois la politique complétée !
# permissive pyserver_t;

########################################
#
# pyserver local policy
#
allow pyserver_t self:fifo_file rw_fifo_file_perms;
allow pyserver_t self:unix_stream_socket create_stream_socket_perms;

domain_use_interactive_fds(pyserver_t)

files_read_etc_files(pyserver_t)

miscfiles_read_localization(pyserver_t)

allow pyserver_t bin_t:file { execute execute_no_trans map };

# Autorise la lecture de /etc/passwd
allow pyserver_t passwd_file_t:file { getattr open read };

# Autorise l'utilisation du port 8000 et l'écoute avec une socket TCP
allow pyserver_t self:tcp_socket { accept bind create getattr listen shutdown read write };
allow pyserver_t node_t:tcp_socket node_bind;
allow pyserver_t soundd_port_t:tcp_socket name_bind;

# Autorise la lecture du contenu de /var/www/html
allow pyserver_t httpd_sys_content_t:dir read;
allow pyserver_t httpd_sys_content_t:file { getattr ioctl open read };

# Autorise le listage du contenu de /tmp
allow pyserver_t tmp_t:dir read;
```

pyserver.fc :

```
/usr/local/bin/pyserver --gen_context(system_u:object_r:pyserver_exec_t,s0)
```

pyserver.if :

```
<vide>
```

Les commandes suivantes permettent de tester le bon fonctionnement de la politique :

```
$ sudo mkdir -p /var/www/html/test
$ echo test | sudo tee /var/www/html/test/a
$ restorecon -RFv /var/www/html

$ curl localhost:8000/var/www/html/test/a
test
$ curl localhost:8000/tmp/
...
$ curl localhost:8000/home/
<404>
$ curl localhost:8000/etc/shadow
<404>
```