

SELinux

Pour afficher toutes les règles fcontext du système :

```
semanage fcontext -l
```

Pour donner à un dossier et tout ses sous-dossiers/fichiers les même contexte qu'un autre, utilisez "l'équivalence"

```
semanage fcontext -a -e /home /mnt/nextcloud-lamal
```

Comprendre un message d'AVC

Dans l'exemple suivant d'AVC de qbittorrent

```
type=AVC msg=audit(1685394119.816:61526): avc: denied { name_bind } for pid=157511  
comm="qbittorrent-nox" src=8080 scontext=system_u:system_r:qbittorrent.process  
:s0:c557,c831 tcontext=system_u:object_r:http_cache_port_t:s0 tclass=tcp_socket  
permissive=0
```

`name_bind` est l'action

Web

Autoriser un dossier à être lu par le serveur Web.

```
chcon -Rt httpd_sys_content_t /path/to/www
```

Conteneurs

Autoriser un conteneur à accéder à un dossier mappé sur l'hôte (flag `Z`).

```
podman run --rm -v /path/to/volume:/data:Z debian
```

Creer une policy custom (podman):

Extraire les settings du conteneur et run udica dessus :

```
podman inspect [id/nom du pod] > conteneur.json
```

```
udica -j conteneur.json mon_conteneur
```

Editer le fichier .cil créé, et ajouter/modifier selon ce que vous voulez autoriser

Exemple pour qbittorrent :

```
(block qbittorrent
  (blockinherit container)
  (blockinherit restricted_net_container)
  (allow process process ( capability ( chown dac_override fowner fsetid kill net_bind_service setfcap setgid
setpcap setuid sys_chroot )))

  # Autorise a se bind et a contacter tous les ports
  (allow process port_type ( tcp_socket ( name_bind name_connect )))
  (allow process port_type ( udp_socket ( name_bind )))

  # Autorise a acceder aux fichier avec le contexte httpd_user_ra_content_t
  (allow process httpd_user_ra_content_t ( dir ( add_name create getattr ioctl lock open read remove_name
rmdir search setattr write )))
  (allow process httpd_user_ra_content_t ( file ( append create getattr ioctl lock map open read rename setattr
link unlink write )))
  (allow process httpd_user_ra_content_t ( fifo_file ( getattr read write append ioctl lock open )))
  (allow process httpd_user_ra_content_t ( sock_file ( append getattr open read write link create setattr unlink
execute rename )))

  (allow process default_t ( dir ( add_name create getattr ioctl lock open read remove_name rmdir search
setattr write )))
  (allow process default_t ( file ( append create getattr ioctl lock map open read rename setattr unlink write )))
  (allow process default_t ( fifo_file ( getattr read write append ioctl lock open )))
  (allow process default_t ( sock_file ( append getattr open read write )))
  (allow process var_t ( dir ( add_name create getattr ioctl lock open read remove_name rmdir search setattr
write )))
  (allow process var_t ( file ( append create getattr ioctl lock map open read rename setattr unlink write )))
  (allow process var_t ( fifo_file ( getattr read write append ioctl lock open )))
```

```
(allow process var_t ( sock_file ( append getattr open read write )))  
)
```

Définir le contexte (httpd_user_ra_content_t ici) des dossiers qui sont accédés par le conteneur (et d'autres process éventuels)

```
semanage fcontext -a -t httpd_user_ra_content_t "/var/opt/qbittorrent/config(/.*)"?"  
semanage fcontext -a -t httpd_user_ra_content_t "/data/downloads(/.*)"?"  
restorecon -Rv /var/opt/qbittorrent/config/ # Ajouter l'option -F si ça marche pas  
restorecon -Rv /data/downloads/ # Pareil
```

importer la policy :

```
semodule -i qbittorrent.cil /usr/share/udica/templates/{base_container.cil,net_container.cil}
```

relancer le conteneur, verifier les alertes selinux, autoriser ce qui génère les alertes, supprimer et reimporter la policy, relancer le conteneur, repeat until it works...

```
semodule -r qbittorrent # Supprime la policy  
tail -f /var/log/audit/audit.log # Lookout for les AVC here
```

Révision #19

Créé 28 mai 2023 17:15:10 par blacksponge

Mis à jour 1 janvier 2025 13:54:47 par sinistag