SELinux

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 accedés 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 qbittorent  # Supprime la policy
tail -f /var/log/audit/audit.log  # Lookout for les AVC here
```

Custom Services outside container

Custom policy

- Créer un dossier `/root/SELinux/<name>`
- Dans ce dossier générer le template avec `sepolicy generate --init <path_de_l'executable>`
- Editer les fichiers nécessaire, puis recompiler la politique avec `./<name>.sh`

Le fichier .te contient les régles de "policy type enforcement"

Le fichier .fc contient les "droits" selinux appliqué aux fichiers

Fichier .te

En haut du fichier .te il y a `permissive <..._t>`, ça place le service en mode permissif, selinux ne va pas s'appliquer, il faut le commenter.

le fichier .te est composé en différentes parties:

Des définitions de types:

`type gonic_t;` : ici on définit le type gonic_t (merci captain obvious)

```
type gonic_file_t;
files_type(gonic_file_t);
```

on définit le type gonic_file_t, et on dit à SELINUX que ce sera un type pour les fichiers (et dossier)

Des imports de type déja existant:

```
require {
    type httpd_user_ra_content_t;
    type var_t;
}
```

Des fonctions

```
manage_files_pattern(gonic_t, gonic_file_t, gonic_file_t);
```

Cette fonction peut être découverte via la commande bash suivante: `macro-expander "manage_files_pattern(gonic_t, gonic_file_t, gonic_file_t)"`

Qui nous donne:

```
allow gonic_t gonic_file_t:dir { open read getattr lock search ioctl add_name remove_name write };
allow gonic_t gonic_file_t:file { create open getattr setattr read write append rename link unlink ioctl lock watch watch_reads };
```

en gros, les executables `gonic_t` pourront lire et ecrire les dossier et fichiers avec `gonic_file_t`

Quelques fonction utiles:

- `files_read_etc_files(gonic_t)` pour lire les fichiers dans /etc
- `corenet_tcp_bind_generic_node(gonic_t)`
- `corenet_tcp_bind_generic_port(gonic_t)`
- `manage_files_pattern(gonic_t, gonic_file_t, gonic_file_t)`

Des règles:

Les règles commencent par `allow` puis le type SELINUX de l'executable `gonic_t`, puis le type SELINUX sur lequel on veut une permission `gonic_file_t:dir` ou `self:unix_stream_socket` puis un droit ou plusieurs `{read map getattr open}`

Par exemple:

- `allow gonic_t httpd_user_ra_content_t:file {read map getattr open};` on autorise `gonic_t` a 'read, map, getattr et open' sur les fichiers `httpd_user_ra_content_t`

on peux remplacer le :file par :dir

- `allow gonic_t httpd_user_ra_content_t:dir { getattr search open read};`

Fichier .fc (TFC huhuhu)

ce sont les droits SELINUX qu'on va appliquer aux fichiers:

une regex `/srv/gonic(/.*)?` puis un type (fichier/link/dir/...) `-d` puis un droit selinux `gen_context(system_u:object_r:gonic_file_t,s0)`

Pour les types on a

- `--` seulement les fichiers

- `-d` seulement les dossiers

Ne rien mettre signifie "yolo fait le sur tout les types"

L'exemple du fichier TC de gonic:

```
/srv/gonic/go/bin/gonic□□--□gen_context(system_u:object_r:gonic_exec_t,s0)


/srv/gonic(/.*)?□□--□gen_context(system_u:object_r:gonic_file_t,s0)
/srv/gonic(/.*)?□□-d□gen_context(system_u:object_r:gonic_file_t,s0)
/srv/gonic□□□-d□gen_context(system_u:object_r:gonic_file_t,s0)
/data/media/musique_tom(/.*)    --□gen_context(system_u:object_r:httpd_user_ra_content_t,s0)
/data/media/musique_tom(/.*)    -d□gen_context(system_u:object_r:httpd_user_ra_content_t,s0)
```

OK mais ca marche pas

1 - Aller sur cockpit.altair.ppsfleet.navy dans l'onglet SELINUX (les dernierres erreurs sont en bas)

2 - Demander a SELINUX une conf automatique: `ausearch -c "gonic" --raw | audit2allow -R`

3 - Bien appliquer les droits SELINUX sur les fichiers `restorecon -R <path>`

4 - Vérifier les droits `ls -lZ <path>`

5 - Des fois c'est pas SELINUX, c'est juste un chmod qui est mal passé.

Révision #17
Créé 28 mai 2023 17:15:10 par blacksponge
Mis à jour 16 octobre 2023 09:14:01 par gurimarukin