Selinux aide-memoire

Table of Contents

SELinux apporte une couche de sécurité supplémentaire aux ressources du système. Il fournit le MAC (contrôle d'accès obligatoire) contrairement au DAC (Contrôle d'accès discrétionnaire). **SELinux** peut fonctionner dans l'un des 3 modes :

1. **Enforced**: les actions contraires à la politique sont bloquées et un événement correspondant est consigné dans le journal d'audit.
2. **Permissive**: Le mode permissif charge le logiciel **SELinux**, mais n'applique pas les règles, seule la journalisation est effectuée.
3. **Disabled**: Le **SELinux** est entièrement désactivé.

Commandes de base

Vérifier si SELinux est activé ou désactivé

Pour vérifier l'état de **SELinux**, utiliser cette commande pour vérifier l'état actuel :

```
getforce
```

La sortie sera l'une des 3 options décrites ci-dessus. Pour plus verbeux (dans le cas de permissif), utiliser :

```
sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

Pour vérifier par programmation le statut comme vrai/faux, une façon pourrait être :

```
# selinuxenabled
if [ $? -ne 0 ]
then
    echo "DISABLED"
else
    echo "ENABLED"
fi
```

Désactivation temporaire

Pour désactiver **SELinux** temporairement, dans le cadre d'un test par exemple :

```
setenforce 0
```

Cette commande **SELinux** met la valeur du fichier /selinux/enforce à 0. C'est le mode permissive, **SElinux** écoute des évènements et écrit de logs mais il ne bloquera plus rien.

On peut procéder aussi comme ci dessous :

```
echo "0" > /selinux/enforce
```

Désactivation temporaire car **SELinux** sera de nouveau activé au prochain démarrage ou si on le réacitive avec la commande suivante :

```
setenforce  1
```

ou encore

```
echo "1" > /selinux/enforce
```

Désactivation permanente :

Pour que ce changement soit permanentent il faut éditer le fichier `/etc/selinux/config` :

```
SELINUX=enforcing
```

enforcing signifie que la politique de sécurité **SELinux** est appliquer.

Pour ne plus l'appliqué il faut remplacer **enforcing** par **permissive**:

```

```

```
SELINUX=permissive
```

SELinux se contentera d'écrire dans les logs.

On peut également mettre `disabled` pour que **SELinux** soit complétement arrêté.

Restauration des contextes

La commande **restorecon** permet de restaurer le(s) contexte(s) original(aux) du ou des chemins spécifiés..

Pour restaurer uniquement le contexte d'un fichier:

```
restorecon -v /var/www/html/index.html
```

Pour restaurer de manière récursive les contextes de sécurité par défaut pour tout le répertoire:

```
restorecon -Rv /var/www/html
```

Pour simplement examiner les contextes de sécurité du répertoire `/var/www/html` pour voir si des fichiers nécessitaient que leurs contextes de sécurité soient restaurés, utiliser **restorecon** avec le commutateur **-n** pour empêcher tout changement de libellé:

```
restorecon -Rv -n /var/www/html
```

Dans certains cas, il peut également arriver que l'utilisateur ait déplacé des fichiers d'un type répertorié dans `/etc/selinux/target/contexts/customizable_types`. Ces types sont traités comme spéciaux par **restorecon**, en ce sens qu'ils ne seront généralement pas renommés par **restorecon**. En effet, ces types sont soit associés à des catégories (comme indiqué dans Sécurité multi-catégories), soit à des types de contenu utilisateur que l'utilisateur est autorisé à personnaliser. Pour renommer le contenu auquel est associé un type personnalisable, il faut exécuter **restorecon** avec l'indicateur **F**:

```
restorecon -RvF /var/www/html
```

Dépannage SElinux

Dépannage des messages SELinux AVC sur la ligne de commande

Lorsque SELinux refuse une action, un message Access Vector Cache (AVC) est enregistré dans les fichiers `/var/log/audit/audit.log` et `/var/log/messages` ou le démon **journald** l'enregistre.

Pour vérifier que SELinux a refusé une action, suivre ces étapes de dépannage de base :

Utiliser l'utilitaire **ausearch** pour rechercher les messages AVC récents et confirmer que SELinux refuse l'action :

```
ausearch -m AVC,USER_AVC -ts recent
```

L'option `-m` spécifie le type d'informations renvoyées par ausearch. L'option `-ts` spécifie l'horodatage. Par exemple, `-ts recent` renvoie les messages AVC des 10 dernières minutes ou `-ts Today` renvoie les messages de toute la journée.

Utiliser l'utilitaire **journalctl** pour afficher plus d'informations sur le message AVC : Brut

```
journalctl -t setroubleshoot --since=[time]
```

<WRAPinfo>Remplacer [time] par l'heure du message AVC trouvé lors de la première étape.</WRAP>

Utiliser l'utilitaire **sealert** pour inspecter davantage le message AVC :

```
sealert -l [message_ID]
```

Remplacer [message_ID] par le numéro du message AVC.

Dépannage

Effectuer des actions selon les suggestions fournies par **sealert**. Par exemple, utiliser l'utilitaire **recoverycon** pour corriger des fichiers mal étiquetés ou activer des booléens particuliers.

Dans l'exemple suivant, **SELinux** a empêché le processus **httpd** d'accéder au fichier `/var/log/httpd/error_log` car il était incorrectement étiqueté avec le type **SELinux** `var_log_t` :

```
# sealert -l e9d8fa2e-3608-4ffa-9e72-31a1b85e460b
    SELinux is preventing httpd from open access on the file
/var/log/httpd/error_log.
```

Pour corriger l'étiquette par défaut de `/var/log/httpd/error.log` sur `httpd_log_t`, puis ensuite, exécuter **recoverycon**:

```
/sbin/restorecon -v /var/log/httpd/error_log
```

Dans l'exemple suivant, **SELinux** a empêché le processus **plugin-containe** de se connecter au réseau à l'aide du protocole TCP et d'utiliser le service **Bluejeans** car les booléens `mozilla_plugin_can_network_connect` et `mozilla_plugin_use_bluejeans` n'étaient pas activés :

```
 sealert -l fc46b9d4-e5a1-4738-95a7-26616d0858b0
 SELinux empêche le plugin-containe d'accéder à name_connect sur le port
tcp_socket 5000.
```

Pour autoriser le domaine du plugin Mozilla à se connecter au réseau via TCP, il faut e informer SELinux en activant le booléen 'mozilla*plugin*can*network*connect'.

```
 setsebool -P mozilla_plugin_can_network_connect 1
```

Pour autoriser le plugin Mozilla à utiliser Bluejeans, il faut en informer SELinux en activant le booléen 'mozilla*plugin*use_bluejeans'.

```
 setsebool -P mozilla_plugin_use_bluejet
```

Dans l'exemple suivant, **SELinux** a refusé au processus **passwd** d'accéder au fichier `/home/user/output.txt` car il n'y a aucune règle dans la politique **SELinux** qui permet à **passwd** d'écrire dans des fichiers étiquetés avec le type `user_home_t SELinux` :

```
sealert -l 1dd524dd-1784-44ef-b6d1-fff9238ed927

 SELinux empêche passwd d'accéder en écriture au fichier
/home/user/output.txt.
```

Pour que **passwd** soit autorisé à accéder en écriture au fichier output.txt par défaut, autoriser cet accès en exécutant :

```
grep passwd /var/log/audit/audit.log | audit2allow -M monpol
semodule -i monpol.pp
```

Cette autorisation est temporaire, pour l'autoriser de façon permanente,il faut générer un module de politique locale pour autoriser cet accès.

Politiques SElinux

Politique SElinux pour les contrôles ICMP

De nombreux problèmes signalés lors de l'analyse des sous-réseaux et de la mise à jour des statuts des hôtes sont liés à l'activation de SElinux. Jusqu'à présent, la solution consistait à désactiver complètement SElinux, mais il s'agissait plus d'une solution de contournement qu'autre chose. Pour activer SElinux sur le serveur, une politique SElinux doit être utilisée avec phpipam.

Fondamentalement, cela permet l'ouverture de sockets IP bruts pour les utilisateurs non root, nécessaires à l'exécution de la commande ping.

- Créer le fichier `http_ping.tt`:

```
cat > http_ping.tt <<'EOF'
module http_ping 1.0;

require {
type httpd_t;
class capability net_raw;
class rawip_socket { getopt create setopt write read };
}

#============= httpd_t ==============
allow httpd_t self:capability net_raw;
allow httpd_t self:rawip_socket { getopt create setopt write read };
EOF
```

- Exécuter les commandes suivantes (en tant qu'utilisateur root) :

```
checkmodule -M -m -o http_ping.mod http_ping.tt
semodule_package -o http_ping.pp -m http_ping.mod
semodule -i http_ping.pp
```