Red Hat Enterprise Linux 9

Utilisation de SELinux

Empêcher les utilisateurs et les processus d'effectuer des interactions non autorisées avec des fichiers et des dispositifs en utilisant Security-Enhanced Linux (SELinux)

Last Updated: 2024-02-16

Red Hat Enterprise Linux 9 Utilisation de SELinux

Empêcher les utilisateurs et les processus d'effectuer des interactions non autorisées avec des fichiers et des dispositifs en utilisant Security-Enhanced Linux (SELinux)

Notice légale

Résumé

En configurant SELinux, vous pouvez renforcer la sécurité de votre système. SELinux est une implémentation du contrôle d'accès obligatoire (MAC) et fournit une couche de sécurité supplémentaire. La politique SELinux définit comment les utilisateurs et les processus peuvent interagir avec les fichiers du système. Vous pouvez contrôler quels utilisateurs peuvent effectuer quelles actions en les associant à des utilisateurs SELinux spécifiques.

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le message de notre directeur technique Chris Wright.

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de Bugzilla.
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. DÉMARRER AVEC SELINUX

Security Enhanced Linux (SELinux) fournit une couche supplémentaire de sécurité du système. SELinux répond fondamentalement à la question : *May <subject> do <action> to <object>?* par exemple : *May a web server access files in users' home directories?*

1.1. INTRODUCTION À SELINUX

La politique d'accès standard basée sur l'utilisateur, le groupe et d'autres autorisations, connue sous le nom de contrôle d'accès discrétionnaire (DAC), ne permet pas aux administrateurs système de créer des politiques de sécurité complètes et fines, telles que la restriction d'applications spécifiques à la seule visualisation des fichiers journaux, tout en permettant à d'autres applications d'ajouter de nouvelles données aux fichiers journaux.

Security Enhanced Linux (SELinux) met en œuvre le contrôle d'accès obligatoire (MAC). Chaque processus et ressource système possède une étiquette de sécurité spéciale appelée *SELinux context*. Un contexte SELinux, parfois appelé *SELinux label*, est un identifiant qui fait abstraction des détails au niveau du système et se concentre sur les propriétés de sécurité de l'entité. Cela permet non seulement de référencer les objets de manière cohérente dans la politique SELinux, mais aussi d'éliminer toute ambiguïté que l'on peut trouver dans d'autres méthodes d'identification. Par exemple, un fichier peut avoir plusieurs noms de chemin valides sur un système qui utilise des montages bind.

La politique SELinux utilise ces contextes dans une série de règles qui définissent comment les processus peuvent interagir entre eux et avec les différentes ressources du système. Par défaut, la politique n'autorise aucune interaction à moins qu'une règle n'en autorise explicitement l'accès.

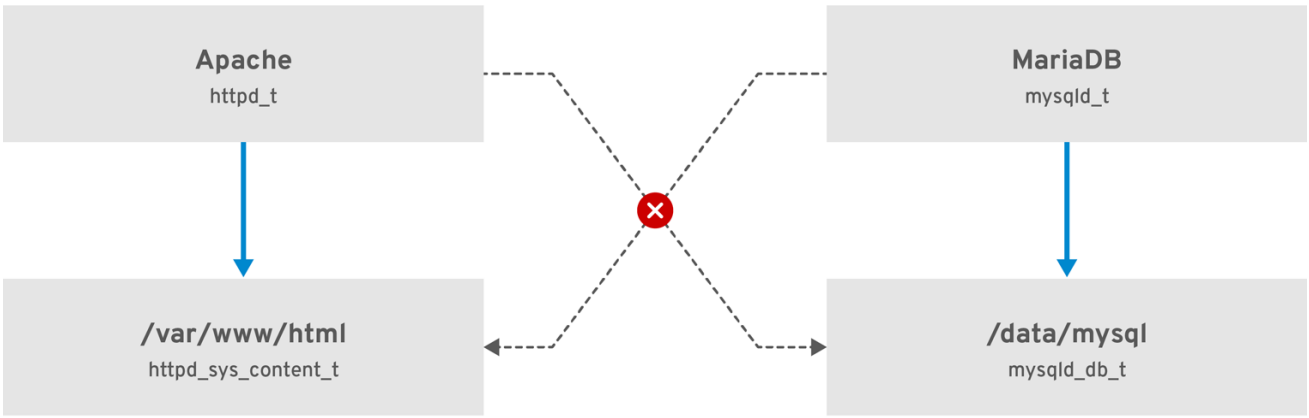
NOTE

N'oubliez pas que les règles de politique SELinux sont vérifiées après les règles DAC. Les règles SELinux ne sont pas utilisées si les règles DAC refusent l'accès en premier, ce qui signifie qu'aucun refus SELinux n'est enregistré si les règles DAC traditionnelles empêchent l'accès.

Les contextes SELinux comportent plusieurs champs : utilisateur, rôle, type et niveau de sécurité. Les informations relatives au type SELinux sont peut-être les plus importantes en ce qui concerne la politique SELinux, car la règle la plus courante qui définit les interactions autorisées entre les processus et les ressources du système utilise les types SELinux et non le contexte SELinux complet. Les types SELinux se terminent par **_t**. Par exemple, le nom du type pour le serveur web est **httpd_t**. Le contexte de type pour les fichiers et les répertoires qui se trouvent normalement dans **/var/www/html/** est **httpd_sys_content_t**. Le contexte de type pour les fichiers et les répertoires qui se trouvent normalement dans **/tmp** et **/var/tmp/** est **tmp_t**. Le contexte de type pour les ports du serveur web est **http_port_t**.

Une règle de politique autorise Apache (le processus du serveur web s'exécutant sous **httpd_t**) à accéder aux fichiers et aux répertoires dont le contexte se trouve normalement dans **/var/www/html/** et dans d'autres répertoires du serveur web (**httpd_sys_content_t**). Il n'y a pas de règle d'autorisation dans la politique pour les fichiers qui se trouvent normalement dans **/tmp** et **/var/tmp/**, et l'accès n'est donc pas autorisé. Avec SELinux, même si Apache est compromis et qu'un script malveillant y accède, il n'est toujours pas en mesure d'accéder au répertoire **/tmp**.

Figure 1.1. Un exemple de la manière dont SELinux peut aider à faire fonctionner Apache et MariaDB de manière sécurisée.



Comme le montre le schéma précédent, SELinux autorise le processus Apache s'exécutant sous le nom **httpd_t** à accéder au répertoire **/var/www/html/** et refuse au même processus d'accéder au répertoire **/data/mysql/** parce qu'il n'y a pas de règle d'autorisation pour les contextes de type **httpd_t** et **mysqld_db_t**. D'autre part, le processus MariaDB s'exécutant sous le nom de **mysqld_t** est en mesure d'accéder au répertoire **/data/mysql/** et SELinux refuse également, à juste titre, au processus de type **mysqld_t** d'accéder au répertoire **/var/www/html/** étiqueté comme **httpd_sys_content_t**.

Ressources supplémentaires

- **selinux(8)** et les pages de manuel répertoriées par la commande **apropos selinux**.
- Pages de manuel listées par la commande **man -k _selinux** lorsque le paquetage **selinux-policy-doc** est installé.
- Le SELinux Coloring Book vous aide à mieux comprendre les concepts de base de SELinux.
- SELinux Wiki FAQ

1.2. AVANTAGES DE SELINUX

SELinux offre les avantages suivants :

- Tous les processus et fichiers sont étiquetés. Les règles SELinux définissent la manière dont les processus interagissent avec les fichiers, ainsi que la manière dont les processus interagissent entre eux. L'accès n'est autorisé que s'il existe une règle SELinux qui l'autorise spécifiquement.
- Contrôle d'accès précis. Au-delà des autorisations UNIX traditionnelles qui sont contrôlées à la discrétion de l'utilisateur et basées sur les ID d'utilisateurs et de groupes Linux, les décisions d'accès SELinux sont basées sur toutes les informations disponibles, telles qu'un utilisateur SELinux, un rôle, un type et, en option, un niveau de sécurité.
- La politique SELinux est définie par l'administration et appliquée à l'ensemble du système.
- Amélioration de l'atténuation des attaques par escalade des privilèges. Les processus s'exécutent dans des domaines et sont donc séparés les uns des autres. Les règles SELinux définissent la manière dont les processus accèdent aux fichiers et aux autres processus. Si un processus est compromis, l'attaquant n'a accès qu'aux fonctions normales de ce processus et aux fichiers auxquels le processus a été configuré pour avoir accès. Par exemple, si le serveur

HTTP Apache est compromis, un pirate ne peut pas utiliser ce processus pour lire les fichiers dans les répertoires personnels des utilisateurs, à moins qu'une règle SELinux spécifique n'ait été ajoutée ou configurée pour autoriser cet accès.

- SELinux peut être utilisé pour assurer la confidentialité et l'intégrité des données, ainsi que pour protéger les processus contre les intrants non fiables.

Toutefois, SELinux ne l'est pas :

- logiciel antivirus,
- remplacer les mots de passe, les pare-feu et autres systèmes de sécurité,
- une solution de sécurité tout-en-un.

SELinux est conçu pour améliorer les solutions de sécurité existantes, et non pour les remplacer. Même lorsque SELinux est utilisé, il est important de continuer à suivre les bonnes pratiques de sécurité, telles que la mise à jour des logiciels, l'utilisation de mots de passe difficiles à deviner et de pare-feu.

1.3. EXEMPLES DE SELINUX

Les exemples suivants montrent comment SELinux renforce la sécurité :

- L'action par défaut est le refus. S'il n'existe pas de règle SELinux autorisant l'accès, par exemple pour un processus ouvrant un fichier, l'accès est refusé.
- SELinux peut confiner les utilisateurs de Linux. Il existe un certain nombre d'utilisateurs SELinux confinés dans la politique SELinux. Les utilisateurs Linux peuvent être associés à des utilisateurs SELinux confinés afin de tirer parti des règles et mécanismes de sécurité qui leur sont appliqués. Par exemple, en associant un utilisateur Linux à l'utilisateur SELinux **user_u**, on obtient un utilisateur Linux qui n'est pas en mesure d'exécuter, sauf configuration contraire, les applications set user ID (setuid), telles que **sudo** et **su**.
- Séparation accrue des processus et des données. Le concept de SELinux *domains* permet de définir quels processus peuvent accéder à certains fichiers et répertoires. Par exemple, avec SELinux, sauf configuration contraire, un attaquant ne peut pas compromettre un serveur Samba, puis utiliser ce serveur Samba comme vecteur d'attaque pour lire et écrire dans des fichiers utilisés par d'autres processus, tels que les bases de données MariaDB.
- SELinux permet d'atténuer les dommages causés par les erreurs de configuration. Les serveurs du système de noms de domaine (DNS) répliquent souvent les informations entre eux lors d'un transfert de zone. Les attaquants peuvent utiliser les transferts de zone pour mettre à jour les serveurs DNS avec de fausses informations. Lorsque le Berkeley Internet Name Domain (BIND) est utilisé comme serveur DNS dans RHEL, même si un administrateur oublie de limiter les serveurs pouvant effectuer un transfert de zone, la stratégie SELinux par défaut empêche les mises à jour des fichiers de zone [1] les mises à jour des fichiers de zone sont effectuées par les serveurs qui utilisent les transferts de zone, par le démon BIND **named** lui-même et par d'autres processus.
- Sans SELinux, un pirate peut abuser d'une vulnérabilité à la traversée de chemin sur un serveur web Apache et accéder aux fichiers et répertoires stockés sur le système de fichiers en utilisant des éléments spéciaux tels que **../**. Si un pirate tente une attaque sur un serveur fonctionnant avec SELinux en mode d'exécution, SELinux refuse l'accès aux fichiers auxquels le processus **httpd** ne doit pas accéder. SELinux ne peut pas bloquer complètement ce type d'attaque, mais il l'atténue efficacement.

- SELinux en mode exécutoire empêche avec succès l'exploitation des opérateurs de déréférencement de pointeur NULL du noyau sur les plates-formes non-SMAP (CVE-2019-9213). Les attaquants utilisent une vulnérabilité dans la fonction **mmap**, qui ne vérifie pas le mappage d'une page nulle, pour placer du code arbitraire sur cette page.

- Le booléen **deny_ptrace** SELinux et SELinux en mode exécution protègent les systèmes contre la vulnérabilité **PTRACE_TRACEME** (CVE-2019-13272). Une telle configuration permet d'éviter les scénarios dans lesquels un attaquant peut obtenir les privilèges **root**.

- Les booléens SELinux **nfs_export_all_rw** et **nfs_export_all_ro** constituent un outil facile à utiliser pour éviter les mauvaises configurations du système de fichiers réseau (NFS), telles que le partage accidentel de répertoires /**home**.

Ressources supplémentaires

- SELinux en tant que pilier de sécurité d'un système d'exploitation - Avantages et exemples concrets Article de la base de connaissances

1.4. ARCHITECTURE ET PAQUETS SELINUX

SELinux est un module de sécurité Linux (LSM) intégré au noyau Linux. Le sous-système SELinux du noyau est piloté par une politique de sécurité contrôlée par l'administrateur et chargée au démarrage. Toutes les opérations d'accès au système au niveau du noyau, pertinentes pour la sécurité, sont interceptées par SELinux et examinées dans le contexte de la politique de sécurité chargée. Si la politique chargée autorise l'opération, celle-ci se poursuit. Dans le cas contraire, l'opération est bloquée et le processus reçoit une erreur.

Les décisions SELinux, telles que l'autorisation ou le refus d'accès, sont mises en cache. Ce cache est connu sous le nom de cache de vecteur d'accès (AVC). Lorsque ces décisions sont mises en cache, les règles de politique SELinux doivent être vérifiées moins souvent, ce qui améliore les performances. N'oubliez pas que les règles de politique SELinux n'ont aucun effet si les règles DAC refusent l'accès en premier. Les messages d'audit bruts sont enregistrés sur le site **/var/log/audit/audit.log** et commencent par la chaîne **type=AVC**.

Dans RHEL 9, les services système sont contrôlés par le démon **systemd**; **systemd** démarre et arrête tous les services, et les utilisateurs et processus communiquent avec **systemd** à l'aide de l'utilitaire **systemctl**. Le démon **systemd** peut consulter la politique SELinux et vérifier l'étiquette du processus appelant et l'étiquette du fichier unitaire que l'appelant tente de gérer, puis demander à SELinux si l'appelant est autorisé ou non à accéder au système. Cette approche renforce le contrôle d'accès aux capacités critiques du système, qui comprennent le démarrage et l'arrêt des services du système.

Le démon **systemd** fonctionne également comme un gestionnaire d'accès SELinux. Il récupère l'étiquette du processus exécutant **systemctl** ou du processus qui a envoyé un message **D-Bus** à **systemd**. Le démon recherche ensuite l'étiquette du fichier d'unité que le processus voulait configurer. Enfin, **systemd** peut extraire des informations du noyau si la politique SELinux autorise l'accès spécifique entre l'étiquette du processus et l'étiquette du fichier d'unité. Cela signifie qu'une application compromise qui doit interagir avec **systemd** pour un service spécifique peut maintenant être confinée par SELinux. Les rédacteurs de politiques peuvent également utiliser ces contrôles fins pour confiner les administrateurs.

Si un processus envoie un message **D-Bus** à un autre processus et si la politique SELinux n'autorise pas la communication **D-Bus** entre ces deux processus, le système affiche un message de refus **USER_AVC** et la communication D-Bus est interrompue. Notez que la communication D-Bus entre deux processus fonctionne de manière bidirectionnelle.

IMPORTANT

Pour éviter un étiquetage SELinux incorrect et les problèmes qui en découlent, assurez-vous que vous démarrez les services à l'aide d'une commande **systemctl start**.

RHEL 9 fournit les paquets suivants pour travailler avec SELinux :

- politiques : **selinux-policy-targeted**, **selinux-policy-mls**
- outils : **policycoreutils**, **policycoreutils-gui**, **libselinux-utils**, **policycoreutils-python-utils**, **setools-console**, **checkpolicy**

1.5. ÉTATS ET MODES SELINUX

SELinux peut fonctionner dans l'un des trois modes suivants : renforcement, permissif ou désactivé.

- Le mode "Enforcing" est le mode de fonctionnement par défaut et recommandé. En mode "Enforcing", SELinux fonctionne normalement, appliquant la politique de sécurité chargée sur l'ensemble du système.
- En mode permissif, le système agit comme si SELinux appliquait la politique de sécurité chargée, notamment en étiquetant les objets et en émettant des entrées de refus d'accès dans les journaux, mais il ne refuse en fait aucune opération. Bien qu'il ne soit pas recommandé pour les systèmes de production, le mode permissif peut être utile pour le développement et le débogage de la politique SELinux.
- Le mode désactivé est fortement déconseillé ; non seulement le système évite d'appliquer la politique SELinux, mais il évite également d'étiqueter les objets persistants tels que les fichiers, ce qui rend difficile l'activation de SELinux à l'avenir.

Utilisez l'utilitaire **setenforce** pour passer du mode "enforcing" au mode "permissive". Les modifications apportées à l'aide de **setenforce** ne sont pas prises en compte lors des redémarrages. Pour passer en mode "enforcing", entrez la commande **setenforce 1** en tant qu'utilisateur racine de Linux. Pour passer en mode permissif, entrez la commande **setenforce 0**. Utilisez l'utilitaire **getenforce** pour afficher le mode SELinux actuel :

```
# getenforce
Enforcing
```

```
# setenforce 0
# getenforce
Permissive
```

```
# setenforce 1
# getenforce
Enforcing
```

Dans Red Hat Enterprise Linux, vous pouvez configurer des domaines individuels en mode permissif alors que le système fonctionne en mode forcé. Par exemple, pour rendre le domaine *httpd_t* permissif :

```
# semanage permissive -a httpd_t
```

Notez que les domaines permissifs sont un outil puissant qui peut compromettre la sécurité de votre système. Red Hat recommande d'utiliser les domaines permissifs avec prudence, par exemple, lors du débogage d'un scénario spécifique.

---

[1] Fichiers texte contenant des informations DNS, telles que les correspondances entre les noms d'hôtes et les adresses IP.

CHAPITRE 2. MODIFIER LES ÉTATS ET LES MODES SELINUX

Lorsqu'il est activé, SELinux peut fonctionner dans l'un des deux modes suivants : "enforcing" ou "permissive". Les sections suivantes expliquent comment passer de manière permanente à ces modes.

2.1. MODIFICATIONS PERMANENTES DES ÉTATS ET MODES SELINUX

Comme indiqué dans les états et modes SELinux, SELinux peut être activé ou désactivé. Lorsqu'il est activé, SELinux dispose de deux modes : "enforcing" et "permissive".

Utilisez les commandes **getenforce** ou **sestatus** pour vérifier dans quel mode SELinux est exécuté. La commande **getenforce** renvoie **Enforcing**, **Permissive**, ou **Disabled**.

La commande **sestatus** renvoie l'état de SELinux et la politique SELinux utilisée :

```
$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      31
```

AVERTISSEMENT

Lorsque les systèmes utilisent SELinux en mode permissif, les utilisateurs et les processus peuvent étiqueter de manière incorrecte divers objets du système de fichiers. Les objets du système de fichiers créés lorsque SELinux est désactivé ne sont pas étiquetés du tout. Ce comportement pose des problèmes lors du passage au mode "enforcing", car SELinux s'appuie sur l'étiquetage correct des objets du système de fichiers.

Pour éviter que des fichiers mal étiquetés ou non étiquetés ne causent des problèmes, SELinux réétiquette automatiquement les systèmes de fichiers lorsqu'il passe de l'état désactivé au mode permissif ou exécutoire. Utilisez la commande **fixfiles -F onboot** en tant que super-utilisateur pour créer le fichier **/.autorelabel** contenant l'option **-F** afin de garantir que les fichiers seront réétiquetés au prochain redémarrage.

Avant de redémarrer le système pour le ré-étiquetage, assurez-vous que le système démarrera en mode permissif, par exemple en utilisant l'option de noyau **enforcing=0**. Cela empêche le système de ne pas démarrer s'il contient des fichiers non étiquetés requis par **systemd** avant de lancer le service **selinux-autorelabel**. Pour plus d'informations, voir RHBZ#2021835.

2.2. PASSAGE AU MODE PERMISSIF

Utilisez la procédure suivante pour modifier de façon permanente le mode SELinux en mode permissif. Lorsque SELinux fonctionne en mode permissif, la politique SELinux n'est pas appliquée. Le système reste opérationnel et SELinux ne refuse aucune opération, mais enregistre uniquement les messages AVC, qui peuvent ensuite être utilisés pour le dépannage, le débogage et l'amélioration de la politique SELinux. Dans ce cas, chaque AVC n'est consigné qu'une seule fois.

Conditions préalables

- Les paquets **selinux-policy-targeted**, **libselinux-utils** et **policycoreutils** sont installés sur votre système.
- Les paramètres du noyau **selinux=0** ou **enforcing=0** ne sont pas utilisés.

Procédure

1. Ouvrez le fichier **/etc/selinux/config** dans un éditeur de texte de votre choix, par exemple :

```
# vi /etc/selinux/config
```

2. Configurez l'option **SELINUX=permissive**:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. Redémarrer le système :

```
# reboot
```

Vérification

1. Après le redémarrage du système, confirmez que la commande **getenforce** renvoie **Permissive**:

```
$ getenforce
Permissive
```

2.3. PASSAGE AU MODE D'EXÉCUTION

Utilisez la procédure suivante pour faire passer SELinux en mode d'application. Lorsque SELinux fonctionne en mode d'exécution, il applique la stratégie SELinux et refuse l'accès en fonction des règles de la stratégie SELinux. Dans RHEL, le mode d'application est activé par défaut lors de l'installation initiale du système avec SELinux.

Conditions préalables

- Les paquets **selinux-policy-targeted**, **libselinux-utils** et **policycoreutils** sont installés sur votre système.
- Les paramètres du noyau **selinux=0** ou **enforcing=0** ne sont pas utilisés.

Procédure

1. Ouvrez le fichier **/etc/selinux/config** dans un éditeur de texte de votre choix, par exemple :

```
# vi /etc/selinux/config
```

2. Configurez l'option **SELINUX=enforcing**:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. Enregistrez la modification et redémarrez le système :

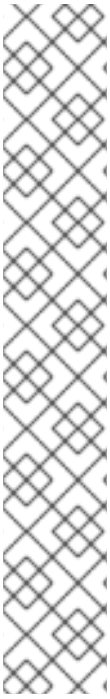
```
# reboot
```

Au démarrage suivant, SELinux réétiquette tous les fichiers et répertoires du système et ajoute le contexte SELinux pour les fichiers et répertoires qui ont été créés lorsque SELinux était désactivé.

Vérification

1. Après le redémarrage du système, confirmez que la commande **getenforce** renvoie **Enforcing**:

```
$ getenforce
Enforcing
```



NOTE

Après le passage en mode d'application, SELinux peut refuser certaines actions en raison de règles de politique SELinux incorrectes ou manquantes. Pour afficher les actions refusées par SELinux, entrez la commande suivante en tant que super-utilisateur :

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts today
```

Alternativement, si le paquetage **setroubleshoot-server** est installé, entrez :

```
# grep "SELinux is preventing" /var/log/messages
```

Si SELinux est actif et que le démon d'audit (**auditd**) ne fonctionne pas sur votre système, recherchez certains messages SELinux dans la sortie de la commande **dmesg**:

```
# dmesg | grep -i -e type=1300 -e type=1400
```

Pour plus d'informations, voir Résolution des problèmes liés à SELinux.

2.4. ACTIVATION DE SELINUX SUR DES SYSTÈMES QUI L'AVAIENT PRÉCÉDEMMENT DÉSACTIVÉ

Pour éviter des problèmes tels que des systèmes incapables de démarrer ou des défaillances de processus, suivez cette procédure lorsque vous activez SELinux sur des systèmes qui l'avaient précédemment désactivé.

AVERTISSEMENT

Lorsque les systèmes utilisent SELinux en mode permissif, les utilisateurs et les processus peuvent étiqueter de manière incorrecte divers objets du système de fichiers. Les objets du système de fichiers créés lorsque SELinux est désactivé ne sont pas étiquetés du tout. Ce comportement pose des problèmes lors du passage au mode "enforcing", car SELinux s'appuie sur l'étiquetage correct des objets du système de fichiers.

Pour éviter que des fichiers incorrectement étiquetés ou non étiquetés ne causent des problèmes, SELinux réétiquette automatiquement les systèmes de fichiers lorsqu'il passe de l'état désactivé au mode permissif ou à l'état renforcé.

Avant de redémarrer le système pour le ré-étiquetage, assurez-vous que le système démarrera en mode permissif, par exemple en utilisant l'option de noyau **enforcing=0**. Cela empêche le système de ne pas démarrer s'il contient des fichiers non étiquetés requis par **systemd** avant de lancer le service **selinux-autorelabel**. Pour plus d'informations, voir RHBZ#2021835.

Procédure

1. Activer SELinux en mode permissif. Pour plus d'informations, voir Passer en mode permissif.

2. Redémarrez votre système :

```
# reboot
```

3. Pour plus d'informations, voir Identification des refus SELinux.

4. Veillez à ce que les fichiers soient réétiquetés lors du prochain redémarrage :

```
# fixfiles -F onboot
```

Cela crée le fichier **/.autorelabel** contenant l'option **-F**.

AVERTISSEMENT

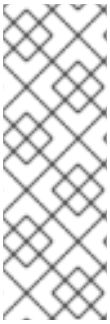
Passez toujours en mode permissif avant d'entrer dans la commande **fixfiles -F onboot**. Cela permet d'éviter que le système ne démarre pas s'il contient des fichiers non étiquetés. Pour plus d'informations, voir RHBZ#2021835.

5. S'il n'y a pas de refus, passez en mode d'application. Pour plus d'informations, voir Modifier les modes SELinux au démarrage.

Vérification

1. Après le redémarrage du système, confirmez que la commande **getenforce** renvoie **Enforcing**:

```
$ getenforce
Enforcing
```



NOTE

Pour exécuter des applications personnalisées avec SELinux en mode d'exécution, choisissez l'un des scénarios suivants :

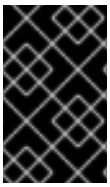
- Exécutez votre application dans le domaine **unconfined_service_t**.
- Rédigez une nouvelle politique pour votre application. Pour plus d'informations, voir la section Rédiger une politique SELinux personnalisée.

Ressources supplémentaires

- La section SELinux states and modes (états et modes SELinux) couvre les changements temporaires de modes.

2.5. DÉSACTIVATION DE SELINUX

Lorsque SELinux est désactivé, la politique SELinux n'est pas chargée du tout ; elle n'est pas appliquée et les messages AVC ne sont pas enregistrés. Par conséquent, tous les avantages liés à l'utilisation de SELinux sont perdus.



IMPORTANT

Red Hat recommande fortement d'utiliser le mode permissif au lieu de désactiver SELinux de manière permanente. Voir Passer en mode permissif pour plus d'informations sur le mode permissif.

Conditions préalables

- Le paquet **grubby** est installé :

```
$ rpm -q grubby
grubby-version
```

Procédure

Pour désactiver SELinux de manière permanente :

1. Configurez votre chargeur de démarrage pour ajouter **selinux=0** à la ligne de commande du noyau :

```
$ sudo grubby --update-kernel ALL --args selinux=0
```

2. Redémarrez votre système :

```
$ reboot
```

Vérification

- Après le redémarrage, confirmez que la commande **getenforce** renvoie **Disabled**:

```
$ getenforce
Disabled
```

2.6. MODIFIER LES MODES SELINUX AU DÉMARRAGE

Au démarrage, vous pouvez définir plusieurs paramètres du noyau pour modifier le fonctionnement de SELinux :

enforcing=0

La définition de ce paramètre entraîne le démarrage du système en mode permissif, ce qui est utile lors de la résolution des problèmes. L'utilisation du mode permissif peut être la seule option pour détecter un problème si votre système de fichiers est trop corrompu. De plus, en mode permissif, le système continue à créer les étiquettes correctement. Les messages AVC créés dans ce mode peuvent être différents de ceux créés en mode "enforcing".
En mode permissif, seul le premier refus d'une série de refus identiques est signalé. Cependant, en mode "enforcing", vous pouvez obtenir un refus lié à la lecture d'un répertoire, et une application s'arrête. En mode permissif, vous obtenez le même message AVC, mais l'application continue à lire les fichiers dans le répertoire et vous obtenez un AVC pour chaque refus supplémentaire.

selinux=0

Ce paramètre fait en sorte que le noyau ne charge aucune partie de l'infrastructure SELinux. Les scripts d'initialisation remarquent que le système a démarré avec le paramètre **selinux=0** et touchent le fichier **/.autorelabel**. Ainsi, le système sera automatiquement réétiqueté lors du prochain démarrage avec SELinux activé.

IMPORTANT

Red Hat ne recommande pas l'utilisation du paramètre **selinux=0**. Pour déboguer votre système, préférez l'utilisation du mode permissif.

autorelabel=1

Ce paramètre oblige le système à procéder à un nouvel étiquetage, de la même manière que pour les commandes suivantes :

```
# touch /.autorelabel
# reboot
```

Si un système de fichiers contient un grand nombre d'objets mal étiquetés, démarrez le système en mode permissif pour que le processus d'étiquetage automatique réussisse.

Ressources supplémentaires

- Pour d'autres paramètres de démarrage du noyau liés à SELinux, tels que **checkreqprot**, voir le fichier **/usr/share/doc/kernel-doc-*<KERNEL_VER>*/Documentation/admin-guide/kernel-parameters.txt** installé avec le paquetage **kernel-doc**. Remplacez la chaîne *<KERNEL_VER>* par le numéro de version du noyau installé, par exemple :

```
# dnf install kernel-doc
$ less /usr/share/doc/kernel-doc-4.18.0/Documentation/admin-guide/kernel-parameters.txt
```

CHAPITRE 3. GESTION DES UTILISATEURS CONFINÉS ET NON CONFINÉS

Les sections suivantes expliquent le mappage des utilisateurs Linux aux utilisateurs SELinux, décrivent les domaines d'utilisateurs confinés de base et démontrent le mappage d'un nouvel utilisateur à un utilisateur SELinux.

3.1. UTILISATEURS CONFINÉS ET NON CONFINÉS

Chaque utilisateur Linux est associé à un utilisateur SELinux à l'aide de la politique SELinux. Cela permet aux utilisateurs Linux d'hériter des restrictions imposées aux utilisateurs SELinux.

Pour voir le mappage des utilisateurs SELinux sur votre système, utilisez la commande **semanage login -l** en tant que root :

```
# semanage login -l
Login Name           SELinux User         MLS/MCS Range        Service

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
```

Dans Red Hat Enterprise Linux, les utilisateurs de Linux sont associés par défaut à l'identifiant SELinux ***default*** par défaut, qui est associé à l'utilisateur SELinux **unconfined_u**. La ligne suivante définit le mappage par défaut :

```
__default__          unconfined_u         s0-s0:c0.c1023       *
```

Les utilisateurs confinés sont soumis aux règles SELinux explicitement définies dans la politique SELinux en vigueur. Les utilisateurs non confinés ne sont soumis qu'à des restrictions minimales par SELinux.

Les utilisateurs de Linux, qu'ils soient confinés ou non, sont soumis à des contrôles de la mémoire exécutable et inscriptible, et sont également soumis à des restrictions par MCS ou MLS.

Pour dresser la liste des utilisateurs SELinux disponibles, entrez la commande suivante :

```
$ seinfo -u
Users: 8
   guest_u
   root
   staff_u
   sysadm_u
   system_u
   unconfined_u
   user_u
   xguest_u
```

Notez que la commande **seinfo** est fournie par le paquetage **setools-console**, qui n'est pas installé par défaut.

Si un utilisateur Linux non confiné exécute une application que la politique SELinux définit comme pouvant passer du domaine **unconfined_t** à son propre domaine confiné, l'utilisateur Linux non confiné est toujours soumis aux restrictions de ce domaine confiné. L'avantage en termes de sécurité est que, même si l'utilisateur de Linux fonctionne en mode non confiné, l'application reste confinée. Par conséquent, l'exploitation d'une faille dans l'application peut être limitée par la politique.

De même, nous pouvons appliquer ces contrôles aux utilisateurs confinés. Chaque utilisateur confiné est limité par un domaine d'utilisateur confiné. La politique SELinux peut également définir une transition entre un domaine d'utilisateurs confinés et son propre domaine confiné cible. Dans ce cas, les utilisateurs confinés sont soumis aux restrictions de ce domaine confiné cible. L'essentiel est que des privilèges spéciaux soient associés aux utilisateurs confinés en fonction de leur rôle.

3.2. RÔLES D'ADMINISTRATEUR CONFINÉS DANS SELINUX

Dans SELinux, les rôles d'administrateur restreint accordent des ensembles spécifiques de privilèges et d'autorisations pour l'exécution de tâches spécifiques aux utilisateurs de Linux qui leur sont assignés. En attribuant des rôles d'administrateur limités distincts, vous pouvez répartir les privilèges sur différents domaines de l'administration du système entre des utilisateurs individuels. Cela s'avère utile dans les scénarios où il y a plusieurs administrateurs, chacun ayant un domaine distinct.

SELinux a les rôles d'administrateur confinés suivants :

auditadm_r

Le rôle d'administrateur d'audit permet de gérer le sous-système d'audit.

dbadm_r

Le rôle d'administrateur de base de données permet de gérer les bases de données MariaDB et PostgreSQL.

logadm_r

Le rôle d'administrateur de journaux permet de gérer les journaux.

webadm_r

L'administrateur web permet de gérer le serveur HTTP Apache.

secadm_r

Le rôle d'administrateur de sécurité permet de gérer la base de données SELinux.

sysadm_r

Le rôle d'administrateur du système permet de faire tout ce que font les rôles précédemment énumérés et dispose de privilèges supplémentaires. Dans les configurations autres que celles par défaut, l'administration de la sécurité peut être séparée de l'administration du système en désactivant le module **sysadm_secadm** dans la stratégie SELinux. Pour des instructions détaillées, voir Section 6.8, « Séparer l'administration du système de l'administration de la sécurité dans MLS » .

Pour affecter un utilisateur Linux à un rôle d'administrateur confiné, voir Section 3.7, « Confiner un administrateur en le faisant correspondre à sysadm_u ».

Ressources supplémentaires

- Pour plus d'informations sur chaque rôle et les types associés, voir les pages de manuel correspondantes :
  - **auditadm_selinux(8)**
  - **dbadm_selinux (8)**
  - **logadm_selinux(8)**
  - **webadm_selinux(8)**
  - **secadm_selinux(8)**

- **sysadm_selinux(8)**

3.3. CAPACITÉS DES UTILISATEURS SELINUX

La politique SELinux associe chaque utilisateur Linux à un utilisateur SELinux. Cela permet aux utilisateurs Linux d'hériter des restrictions des utilisateurs SELinux.

Vous pouvez personnaliser les autorisations pour les utilisateurs confinés dans votre politique SELinux en fonction de besoins spécifiques en ajustant les booléens dans la politique. Vous pouvez déterminer l'état actuel de ces booléens à l'aide de la commande **semanage boolean -l**. Pour dresser la liste de tous les utilisateurs SELinux, de leurs rôles SELinux et des niveaux et plages MLS/MCS, utilisez la commande **semanage user -l** comme **root**.

Tableau 3.1. Rôles des utilisateurs SELinux

User	Rôle par défaut	Autres rôles
unconfined_u	**unconfined_r**	**system_r**
guest_u	**guest_r**	
xguest_u	**xguest_r**	
user_u	**user_r**	
staff_u	**staff_r**	**sysadm_r**
		unconfined_r
		system_r
sysadm_u	**sysadm_r**	
root	**staff_r**	**sysadm_r**
		unconfined_r
		system_r
system_u	**system_r**	

Notez que **system_u** est une identité d'utilisateur spéciale pour les processus et les objets du système, et que **system_r** est le rôle associé. Les administrateurs ne doivent jamais associer cet utilisateur **system_u** et le rôle **system_r** à un utilisateur Linux. Par ailleurs, **unconfined_u** et **root** sont des utilisateurs non restreints. Pour ces raisons, les rôles associés à ces utilisateurs SELinux ne sont pas inclus dans le tableau suivant Types et accès aux rôles SELinux.

Chaque rôle SELinux correspond à un type SELinux et fournit des droits d'accès spécifiques.

Tableau 3.2. Types et accès aux rôles SELinux

Rôle	Type	Connexion à l'aide du système X Window	**su** et **sudo**	Exécuter dans le répertoire personnel et à l'adresse /**tmp** (par défaut)	Mise en réseau
unconfined_r	**unconfined_t**	yes	yes	yes	yes
guest_r	**guest_t**	non	non	yes	non
xguest_r	**xguest_t**	yes	non	yes	navigateurs web uniquement (Firefox, GNOME Web)
user_r	**user_t**	yes	non	yes	yes
staff_r	**staff_t**	yes	seulement **sudo**	yes	yes
auditadm_r	**auditadm_t**		yes	yes	yes
secadm_r	**secadm_t**		yes	yes	yes
sysadm_r	**sysadm_t**	uniquement lorsque le booléen **xdm_sysadm_login** est **on**	yes	yes	yes

- Les utilisateurs de Linux dans les domaines **user_t**, **guest_t** et **xguest_t** ne peuvent exécuter les applications set user ID (setuid) que si la politique SELinux le permet (par exemple, **passwd**). Ces utilisateurs ne peuvent pas exécuter les applications setuid **su** et **sudo** et ne peuvent donc pas utiliser ces applications pour devenir root.

- Les utilisateurs de Linux dans les domaines **sysadm_t**, **staff_t**, **user_t**, et **xguest_t** peuvent se connecter en utilisant le système X Window et un terminal.

- Par défaut, les utilisateurs de Linux dans les domaines **staff_t**, **user_t**, **guest_t** et **xguest_t** peuvent exécuter des applications dans leurs répertoires personnels et /**tmp**.
  Pour les empêcher d'exécuter des applications, qui héritent des autorisations des utilisateurs, dans les répertoires auxquels ils ont accès en écriture, définissez les booléens **guest_exec_content** et **xguest_exec_content** sur **off**. Cela permet d'éviter que des applications défectueuses ou malveillantes ne modifient les fichiers des utilisateurs.

- Le seul accès au réseau dont disposent les utilisateurs de Linux dans le domaine **xguest_t** est la connexion de Firefox aux pages web.

- L'utilisateur **sysadm_u** ne peut pas se connecter directement en utilisant SSH. Pour activer les connexions SSH pour **sysadm_u**, définissez le booléen **ssh_sysadm_login** sur **on**:

```
# setsebool -P ssh_sysadm_login on
```

Outre les utilisateurs SELinux déjà mentionnés, il existe des rôles spéciaux qui peuvent être attribués à ces utilisateurs à l'aide de la commande **semanage user**. Ces rôles déterminent ce que SELinux permet à l'utilisateur de faire :

- **webadm_r** ne peut administrer que les types SELinux liés au serveur HTTP Apache.
- **dbadm_r** ne peut administrer que les types SELinux liés à la base de données MariaDB et au système de gestion de base de données PostgreSQL.
- **logadm_r** ne peut administrer que les types SELinux liés aux processus **syslog** et **auditlog**.
- **secadm_r** ne peut qu'administrer SELinux.
- **auditadm_r** ne peut administrer que les processus liés au sous-système d'audit.

Pour dresser la liste de tous les rôles disponibles, entrez la commande **seinfo -r**:

```
$ seinfo -r
Roles: 14
   auditadm_r
   dbadm_r
   guest_r
   logadm_r
   nx_server_r
   object_r
   secadm_r
   staff_r
   sysadm_r
   system_r
   unconfined_r
   user_r
   webadm_r
   xguest_r
```

Notez que la commande **seinfo** est fournie par le paquetage **setools-console**, qui n'est pas installé par défaut.

Ressources supplémentaires

- **seinfo(1)**, **semanage-login(8)**, et **xguest_selinux(8)** pages de manuel

3.4. AJOUT D'UN NOUVEL UTILISATEUR AUTOMATIQUEMENT ASSOCIÉ À L'UTILISATEUR SELINUX UNCONFINED_U

La procédure suivante montre comment ajouter un nouvel utilisateur Linux au système. L'utilisateur est automatiquement associé à l'utilisateur SELinux **unconfined_u**.

Conditions préalables

- L'utilisateur **root** s'exécute sans contrainte, comme c'est le cas par défaut dans Red Hat Enterprise Linux.

Procédure

1. Entrez la commande suivante pour créer un nouvel utilisateur Linux nommé *example.user*:

```
# useradd example.user
```

2. Pour attribuer un mot de passe à l'utilisateur Linux *example.user*:

```
# passwd example.user
Changing password for user example.user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

3. Déconnexion de la session en cours.

4. Connectez-vous en tant qu'utilisateur Linux *example.user*. Lorsque vous vous connectez, le module PAM **pam_selinux** fait automatiquement correspondre l'utilisateur Linux à un utilisateur SELinux (dans ce cas, **unconfined_u**) et met en place le contexte SELinux qui en résulte. L'interpréteur de commandes de l'utilisateur Linux est alors lancé avec ce contexte.

Vérification

1. Lorsque vous êtes connecté en tant qu'utilisateur *example.user*, vérifiez le contexte d'un utilisateur Linux :

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Ressources supplémentaires

- **pam_selinux(8)** page de manuel.

3.5. AJOUT D'UN NOUVEL UTILISATEUR EN TANT QU'UTILISATEUR DÉFINI PAR SELINUX

Procédez comme suit pour ajouter un nouvel utilisateur SELinux au système. Cet exemple de procédure associe l'utilisateur au droit d'utilisateur SELinux **staff_u** avec la commande de création du compte d'utilisateur.

Conditions préalables

- L'utilisateur **root** s'exécute sans contrainte, comme c'est le cas par défaut dans Red Hat Enterprise Linux.

Procédure

1. Entrez la commande suivante pour créer un nouvel utilisateur Linux nommé *example.user* et l'associer à l'utilisateur SELinux **staff_u**:

```
# useradd -Z staff_u example.user
```

2. Pour attribuer un mot de passe à l'utilisateur Linux *example.user*:

```
# passwd example.user
Changing password for user example.user.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

3. Déconnexion de la session en cours.

4. Connectez-vous en tant qu'utilisateur de Linux *example.user*. L'interpréteur de commandes de l'utilisateur est lancé dans le contexte **staff_u**.

Vérification

1. Lorsque vous êtes connecté en tant qu'utilisateur *example.user*, vérifiez le contexte d'un utilisateur Linux :

```
$ id -Z
uid=1000(example.user) gid=1000(example.user) groups=1000(example.user)
context=staff_u:staff_r:staff_t:s0-s0:c0.c1023
```

Ressources supplémentaires

- **pam_selinux(8)** page de manuel.

3.6. CONFINER LES UTILISATEURS RÉGULIERS

Vous pouvez confiner tous les utilisateurs ordinaires de votre système en les associant à l'utilisateur SELinux **user_u**.

Par défaut, tous les utilisateurs Linux de Red Hat Enterprise Linux, y compris les utilisateurs disposant de privilèges administratifs, sont mappés à l'utilisateur SELinux non confiné **unconfined_u**. Vous pouvez améliorer la sécurité du système en assignant des utilisateurs à des utilisateurs SELinux confined. Ceci est utile pour se conformer au Guide de mise en œuvre technique de la sécurité V-71971 .

Procédure

1. Affiche la liste des enregistrements de connexion SELinux. La liste affiche les correspondances entre les utilisateurs Linux et les utilisateurs SELinux :

```
# semanage login -l

Login Name    SELinux User  MLS/MCS Range   Service

__default__   unconfined_u  s0-s0:c0.c1023      *
root          unconfined_u  s0-s0:c0.c1023      *
```

2. Mapper l'utilisateur __default__, qui représente tous les utilisateurs sans correspondance explicite, à l'utilisateur SELinux **user_u**:

```
# semanage login -m -s user_u -r s0 __default__
```

Vérification

1. Vérifiez que l'utilisateur __default__ est associé à l'utilisateur SELinux **user_u**:

```
# semanage login -l

Login Name    SELinux User   MLS/MCS Range    Service

__default__   user_u         s0               *
root          unconfined_u   s0-s0:c0.c1023   *
```

2. Vérifiez que les processus d'un nouvel utilisateur s'exécutent dans le contexte SELinux **user_u:user_r:user_t:s0**.

   a. Créer un nouvel utilisateur :

   ```
   # adduser example.user
   ```

   b. Définir un mot de passe pour *example.user*:

   ```
   # passwd example.user
   ```

   c. Déconnectez-vous en tant que **root** et connectez-vous en tant que nouvel utilisateur.

   d. Affiche le contexte de sécurité pour l'ID de l'utilisateur :

   ```
   [example.user@localhost ~]$ id -Z
   user_u:user_r:user_t:s0
   ```

   e. Affiche le contexte de sécurité des processus en cours de l'utilisateur :

   ```
   [example.user@localhost ~]$ ps axZ
   LABEL                            PID TTY      STAT   TIME COMMAND
   -                                  1 ?        Ss     0:05 /usr/lib/systemd/systemd --switched-root --
   system --deserialize 18
   -                               3729 ?        S      0:00 (sd-pam)
   user_u:user_r:user_t:s0         3907 ?        Ss     0:00 /usr/lib/systemd/systemd --user
   -                               3911 ?        S      0:00 (sd-pam)
   user_u:user_r:user_t:s0         3918 ?        S      0:00 sshd: example.user@pts/0
   user_u:user_r:user_t:s0         3922 pts/0    Ss     0:00 -bash
   user_u:user_r:user_dbusd_t:s0   3969 ?        Ssl    0:00 /usr/bin/dbus-daemon --session --
   address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
   user_u:user_r:user_t:s0         3971 pts/0    R+     0:00 ps axZ
   ```

3.7. CONFINER UN ADMINISTRATEUR EN LE FAISANT CORRESPONDRE À SYSADM_U

Vous pouvez confiner un utilisateur avec des privilèges administratifs en le faisant correspondre directement à l'utilisateur SELinux **sysadm_u**. Lorsque l'utilisateur se connecte, la session s'exécute dans le contexte SELinux **sysadm_u:sysadm_r:sysadm_t**.

Par défaut, tous les utilisateurs Linux de Red Hat Enterprise Linux, y compris les utilisateurs disposant de privilèges administratifs, sont mappés à l'utilisateur SELinux non confiné **unconfined_u**. Vous pouvez améliorer la sécurité du système en assignant des utilisateurs à des utilisateurs SELinux confined. Ceci est utile pour se conformer au Guide de mise en œuvre technique de la sécurité V-71971 .

Conditions préalables

- L'utilisateur **root** s'exécute sans contrainte. Il s'agit de la configuration par défaut de Red Hat Enterprise Linux.

Procédure

1. Facultatif : Pour permettre aux utilisateurs de **sysadm_u** de se connecter au système à l'aide de SSH :

```
# setsebool -P ssh_sysadm_login on
```

2. Créez un nouvel utilisateur, ajoutez-le au groupe d'utilisateurs **wheel** et associez-le à l'utilisateur SELinux **sysadm_u**:

```
# adduser -G wheel -Z sysadm_u example.user
```

3. Facultatif : Associez un utilisateur existant à l'utilisateur SELinux **sysadm_u** et ajoutez l'utilisateur au groupe d'utilisateurs **wheel**:

```
# usermod -G wheel -Z sysadm_u example.user
```

Vérification

1. Vérifiez que ***example.user*** est associé à l'utilisateur SELinux **sysadm_u**:

```
# semanage login -l | grep example.user
example.user     sysadm_u    s0-s0:c0.c1023   *
```

2. Se connecter en tant que ***example.user***par exemple, en utilisant SSH, et afficher le contexte de sécurité de l'utilisateur :

```
[example.user@localhost ~]$ id -Z
sysadm_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

3. Passez à l'utilisateur **root**:

```
$ sudo -i
[sudo] password for example.user:
```

4. Vérifiez que le contexte de sécurité reste inchangé :

```
# id -Z
sysadm_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

5. Essayez une tâche administrative, par exemple en redémarrant le service **sshd**:

```
# systemctl restart sshd
```

S'il n'y a pas de sortie, la commande s'est terminée avec succès.

Si la commande ne se termine pas avec succès, elle affiche le message suivant :

```
Failed to restart sshd.service: Access denied
See system logs and 'systemctl status sshd.service' for details.
```

3.8. CONFINER UN ADMINISTRATEUR À L'AIDE DE SUDO ET DU RÔLE SYSADM_R

Vous pouvez associer un utilisateur spécifique disposant de privilèges administratifs à l'utilisateur SELinux **staff_u** et configurer **sudo** de manière à ce que l'utilisateur puisse obtenir le rôle d'administrateur SELinux **sysadm_r**. Ce rôle permet à l'utilisateur d'effectuer des tâches administratives sans être confronté à des refus SELinux. Lorsque l'utilisateur se connecte, la session s'exécute dans le contexte SELinux **staff_u:staff_r:staff_t**, mais lorsque l'utilisateur entre une commande à l'aide de **sudo**, la session passe dans le contexte **staff_u:sysadm_r:sysadm_t**.

Par défaut, tous les utilisateurs Linux de Red Hat Enterprise Linux, y compris les utilisateurs disposant de privilèges administratifs, sont mappés à l'utilisateur SELinux non confiné **unconfined_u**. Vous pouvez améliorer la sécurité du système en assignant des utilisateurs à des utilisateurs SELinux confined. Ceci est utile pour se conformer au Guide de mise en œuvre technique de la sécurité V-71971 .

Conditions préalables

- L'utilisateur **root** s'exécute sans contrainte. Il s'agit de la configuration par défaut de Red Hat Enterprise Linux.

Procédure

1. Créez un nouvel utilisateur, ajoutez-le au groupe d'utilisateurs **wheel** et associez-le à l'utilisateur SELinux **staff_u**:

```
# adduser -G wheel -Z staff_u example.user
```

2. Facultatif : Associez un utilisateur existant à l'utilisateur SELinux **staff_u** et ajoutez l'utilisateur au groupe d'utilisateurs **wheel**:

```
# usermod -G wheel -Z staff_u example.user
```

3. Pour permettre à *example.user* d'obtenir le rôle d'administrateur SELinux, créez un nouveau fichier dans le répertoire **/etc/sudoers.d/**, par exemple :

```
# visudo -f /etc/sudoers.d/example.user
```

4. Ajoutez la ligne suivante au nouveau fichier :

```
example.user ALL=(ALL) TYPE=sysadm_t ROLE=sysadm_r ALL
```

Vérification

1. Vérifiez que ***example.user*** est associé à l'utilisateur SELinux **staff_u**:

```
# semanage login -l | grep example.user
example.user     staff_u    s0-s0:c0.c1023   *
```

2. Connectez-vous en tant que *example.user*, par exemple en utilisant SSH, et passez à l'utilisateur **root**:

```
[example.user@localhost ~]$ sudo -i
[sudo] password for example.user:
```

3. Afficher le contexte de sécurité de **root**:

```
# id -Z
staff_u:sysadm_r:sysadm_t:s0-s0:c0.c1023
```

4. Essayez une tâche administrative, par exemple en redémarrant le service **sshd**:

```
# systemctl restart sshd
```

S'il n'y a pas de sortie, la commande s'est terminée avec succès.

Si la commande ne se termine pas avec succès, elle affiche le message suivant :

```
Failed to restart sshd.service: Access denied
See system logs and 'systemctl status sshd.service' for details.
```

3.9. RESSOURCES SUPPLÉMENTAIRES

- **unconfined_selinux(8)**, **user_selinux(8)**, **staff_selinux(8)**, et **sysadm_selinux(8)** pages de manuel
- Comment configurer un système avec des utilisateurs limités par SELinux ?

CHAPITRE 4. CONFIGURATION DE SELINUX POUR LES APPLICATIONS ET LES SERVICES AVEC DES CONFIGURATIONS NON STANDARD

Lorsque SELinux est en mode d'application, la politique par défaut est la politique ciblée. Les sections suivantes fournissent des informations sur la mise en place et la configuration de la stratégie SELinux pour divers services après avoir modifié les valeurs par défaut de la configuration, telles que les ports, l'emplacement des bases de données ou les autorisations du système de fichiers pour les processus.

Vous apprenez à modifier les types SELinux pour les ports non standard, à identifier et à corriger les étiquettes incorrectes pour les changements de répertoires par défaut et à ajuster la politique à l'aide des booléens SELinux.

4.1. PERSONNALISATION DE LA POLITIQUE SELINUX POUR LE SERVEUR HTTP APACHE DANS UNE CONFIGURATION NON STANDARD

Vous pouvez configurer le serveur HTTP Apache pour qu'il écoute sur un port différent et pour qu'il fournisse du contenu dans un répertoire autre que celui par défaut. Pour éviter les refus SELinux qui en découlent, suivez les étapes de cette procédure pour ajuster la politique SELinux de votre système.

Conditions préalables

- Le paquet **httpd** est installé et le serveur HTTP Apache est configuré pour écouter sur le port TCP 3131 et pour utiliser le répertoire **/var/test_www/** au lieu du répertoire par défaut **/var/www/**.
- Les paquets **policycoreutils-python-utils** et **setroubleshoot-server** sont installés sur votre système.

Procédure

1. Démarrez le service **httpd** et vérifiez l'état :

```
# systemctl start httpd
# systemctl status httpd
...
httpd[14523]: (13)Permission denied: AH00072: make_sock: could not bind to address
[::]:3131
...
systemd[1]: Failed to start The Apache HTTP Server.
...
```

2. La politique SELinux suppose que **httpd** fonctionne sur le port 80 :

```
# semanage port -l | grep http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
```

3. Modifier le type SELinux du port 3131 pour qu'il corresponde au port 80 :

```
# semanage port -a -t http_port_t -p tcp 3131
```

4. Recommencer **httpd**:

```
# systemctl start httpd
```

5. Cependant, le contenu reste inaccessible :

```
# wget localhost:3131/index.html
...
HTTP request sent, awaiting response... 403 Forbidden
...
```

Trouvez la raison à l'aide de l'outil **sealert**:

```
# sealert -l "*"
...
SELinux is preventing httpd from getattr access on the file /var/test_www/html/index.html.
...
```

6. Comparez les types SELinux pour le chemin standard et le nouveau chemin à l'aide de l'outil **matchpathcon**:

```
# matchpathcon /var/www/html /var/test_www/html
/var/www/html       system_u:object_r:httpd_sys_content_t:s0
/var/test_www/html  system_u:object_r:var_t:s0
```

7. Modifiez le type SELinux du nouveau répertoire de contenu **/var/test_www/html/** pour qu'il corresponde au type du répertoire par défaut **/var/www/html**:

```
# semanage fcontext -a -e /var/www /var/test_www
```

8. Réétiqueter le répertoire **/var** de manière récursive :

```
# restorecon -Rv /var/
...
Relabeled /var/test_www/html from unconfined_u:object_r:var_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /var/test_www/html/index.html from unconfined_u:object_r:var_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
```

Vérification

1. Vérifiez que le service **httpd** est en cours d'exécution :

```
# systemctl status httpd
...
Active: active (running)
...
```

```
systemd[1]: Started The Apache HTTP Server.
httpd[14888]: Server configured, listening on: port 3131
...
```

2. Vérifiez que le contenu fourni par le serveur HTTP Apache est accessible :

```
# wget localhost:3131/index.html
...
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/html]
Saving to: ‘index.html’
...
```

Ressources supplémentaires

- Les pages de manuel **semanage(8)**, **matchpathcon(8)**, et **sealert(8)**.

4.2. AJUSTEMENT DE LA POLITIQUE DE PARTAGE DES VOLUMES NFS ET CIFS À L'AIDE DES BOOLÉENS SELINUX

Vous pouvez modifier certaines parties de la politique SELinux au moment de l'exécution à l'aide de booléens, même si vous n'avez aucune connaissance de l'écriture de la politique SELinux. Cela permet d'apporter des modifications, telles que l'autorisation pour les services d'accéder aux volumes NFS, sans recharger ou recompiler la politique SELinux. La procédure suivante permet de dresser la liste des booléens SELinux et de les configurer afin d'apporter les modifications requises à la politique.

Les montages NFS du côté client sont étiquetés avec un contexte par défaut défini par une stratégie pour les volumes NFS. Dans RHEL, ce contexte par défaut utilise le type **nfs_t**. De même, les partages Samba montés du côté client sont étiquetés avec un contexte par défaut défini par la stratégie. Ce contexte par défaut utilise le type **cifs_t**. Vous pouvez activer ou désactiver des booléens pour contrôler les services autorisés à accéder aux types **nfs_t** et **cifs_t**.

Pour permettre au service Apache HTTP Server (**httpd**) d'accéder aux volumes NFS et CIFS et de les partager, procédez comme suit :

Conditions préalables

- En option, installez le paquet **selinux-policy-devel** pour obtenir des descriptions plus claires et plus détaillées des booléens SELinux dans la sortie de la commande **semanage boolean -l**.

Procédure

1. Identifier les booléens SELinux pertinents pour NFS, CIFS et Apache :

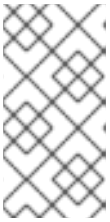
```
# semanage boolean -l | grep 'nfs\|cifs' | grep httpd
httpd_use_cifs                 (off  ,  off)  Allow httpd to access cifs file systems
httpd_use_nfs                  (off  ,  off)  Allow httpd to access nfs file systems
```

2. Liste l'état actuel des booléens :

```
$ getsebool -a | grep 'nfs\|cifs' | grep httpd
httpd_use_cifs --> off
httpd_use_nfs --> off
```

3. Activer les booléens identifiés :

```
# setsebool httpd_use_nfs on
# setsebool httpd_use_cifs on
```



NOTE

Utilisez la commande **setsebool** avec l'option **-P** pour que les modifications soient prises en compte lors des redémarrages. La commande **setsebool -P** nécessite une reconstruction de l'ensemble de la stratégie, ce qui peut prendre un certain temps en fonction de votre configuration.

Vérification

1. Vérifier que les booléens sont bien **on**:

```
$ getsebool -a | grep 'nfs\|cifs' | grep httpd
httpd_use_cifs --> on
httpd_use_nfs --> on
```

Ressources supplémentaires

- **semanage-boolean(8)**, **sepolicy-booleans(8)**, **getsebool(8)**, **setsebool(8)**, **booleans(5)**, et **booleans(8)** pages de manuel

4.3. RESSOURCES SUPPLÉMENTAIRES

- Résolution des problèmes liés à SELinux

CHAPITRE 5. RÉSOLUTION DES PROBLÈMES LIÉS À SELINUX

Si vous envisagez d'activer SELinux sur des systèmes où il a été désactivé précédemment ou si vous exécutez un service dans une configuration non standard, vous devrez peut-être résoudre des situations potentiellement bloquées par SELinux. Notez que dans la plupart des cas, les refus de SELinux sont le signe d'une mauvaise configuration.

5.1. IDENTIFIER LES DÉNIS SELINUX

Ne suivez que les étapes nécessaires de cette procédure ; dans la plupart des cas, vous ne devez effectuer que l'étape 1.

Procédure

1. Lorsque votre scénario est bloqué par SELinux, le fichier **/var/log/audit/audit.log** est le premier endroit à consulter pour obtenir plus d'informations sur un refus. Pour interroger les journaux d'audit, utilisez l'outil **ausearch**. Étant donné que les décisions de SELinux, telles que l'autorisation ou le refus d'accès, sont mises en cache et que ce cache est connu sous le nom d'Access Vector Cache (AVC), utilisez les valeurs **AVC** et **USER_AVC** pour le paramètre de type de message, par exemple :

   ```
   # ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
   ```

   S'il n'y a pas de correspondance, vérifiez si le démon d'audit est en cours d'exécution. Si ce n'est pas le cas, répétez le scénario refusé après avoir démarré **auditd** et vérifiez à nouveau le journal d'audit.

2. Si **auditd** est en cours d'exécution, mais qu'il n'y a pas de correspondance dans la sortie de **ausearch**, vérifiez les messages fournis par le journal **systemd**:

   ```
   # journalctl -t setroubleshoot
   ```

3. Si SELinux est actif et que le démon d'audit ne fonctionne pas sur votre système, recherchez certains messages SELinux dans la sortie de la commande **dmesg**:

   ```
   # dmesg | grep -i -e type=1300 -e type=1400
   ```

4. Même après les trois contrôles précédents, il est encore possible que vous n'ayez rien trouvé. Dans ce cas, les refus de l'AVC peuvent être réduits au silence grâce aux règles du site **dontaudit**.
   Pour désactiver temporairement les règles **dontaudit**, ce qui permet d'enregistrer tous les refus :

   ```
   # semodule -DB
   ```

   Après avoir réexécuté votre scénario de refus et trouvé des messages de refus à l'aide des étapes précédentes, la commande suivante active à nouveau les règles **dontaudit** dans la stratégie :

   ```
   # semodule -B
   ```

5. Si vous appliquez les quatre étapes précédentes et que le problème n'est toujours pas identifié, demandez-vous si SELinux bloque réellement votre scénario :

- Passer en mode permissif :

```
# setenforce 0
$ getenforce
Permissive
```

- Répétez votre scénario.

Si le problème persiste, c'est que quelque chose d'autre que SELinux bloque votre scénario.

5.2. ANALYSE DES MESSAGES DE REFUS SELINUX

Après avoir identifié que SELinux bloque votre scénario, vous devrez peut-être analyser la cause première avant de choisir une solution.

Conditions préalables

- Les paquets **policycoreutils-python-utils** et **setroubleshoot-server** sont installés sur votre système.

Procédure

1. Lister plus de détails sur un refus enregistré à l'aide de la commande **sealert**, par exemple :

```
$ sealert -l "*"
SELinux is preventing /usr/bin/passwd from write access on the file
/root/test.

*****  Plugin leaks (86.2 confidence) suggests ******************************

If you want to ignore passwd trying to write access the test file,
because you believe it should not need this access.
Then you should report this as a bug.
You can generate a local policy module to dontaudit this access.
Do
# ausearch -x /usr/bin/passwd --raw | audit2allow -D -M my-passwd
# semodule -X 300 -i my-passwd.pp

*****  Plugin catchall (14.7 confidence) suggests **************************

...

Raw Audit Messages
type=AVC msg=audit(1553609555.619:127): avc:  denied  { write } for
pid=4097 comm="passwd" path="/root/test" dev="dm-0" ino=17142697
scontext=unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file permissive=0

...

Hash: passwd,passwd_t,admin_home_t,file,write
```

2. Si les résultats obtenus à l'étape précédente ne contiennent pas de suggestions claires :

- Activez l'audit du chemin complet pour voir les chemins d'accès complets aux objets accédés et pour rendre visibles des champs supplémentaires de l'événement Linux Audit :

```
# auditctl -w /etc/shadow -p w -k shadow-write
```

- Vider le cache de **setroubleshoot**:

```
# rm -f /var/lib/setroubleshoot/setroubleshoot.xml
```

- Reproduire le problème.
- Répéter l'étape 1.
  Une fois la procédure terminée, désactivez l'audit du chemin complet :

```
# auditctl -W /etc/shadow -p w -k shadow-write
```

3. Si **sealert** ne renvoie que des suggestions **catchall** ou suggère d'ajouter une nouvelle règle à l'aide de l'outil **audit2allow**, comparez votre problème aux exemples répertoriés et expliqués dans SELinux denials (refus d'autorisation SELinux) dans le journal d'audit .

Ressources supplémentaires

- **sealert(8)** page de manuel

5.3. CORRECTION DES DÉNIS SELINUX ANALYSÉS

Dans la plupart des cas, les suggestions fournies par l'outil **sealert** vous indiquent comment résoudre les problèmes liés à la politique SELinux. Voir Analyse des messages de refus SELinux pour savoir comment utiliser **sealert** pour analyser les refus SELinux.

Soyez prudent lorsque l'outil suggère d'utiliser l'outil **audit2allow** pour les changements de configuration. Vous ne devez pas utiliser **audit2allow** pour générer un module de politique locale comme première option lorsque vous constatez un refus SELinux. Le dépannage doit commencer par une vérification de l'existence d'un problème d'étiquetage. Le deuxième cas le plus fréquent est que vous avez modifié la configuration d'un processus et que vous avez oublié d'en informer SELinux.

Problèmes d'étiquetage

Une cause fréquente de problèmes d'étiquetage est l'utilisation d'un répertoire non standard pour un service. Par exemple, au lieu d'utiliser **/var/www/html/** pour un site Web, un administrateur pourrait vouloir utiliser **/srv/myweb/**. Sur Red Hat Enterprise Linux, le répertoire **/srv** est étiqueté avec le type **var_t**. Les fichiers et les répertoires créés dans **/srv** héritent de ce type. De même, les objets nouvellement créés dans les répertoires de premier niveau, tels que **/myserver**, peuvent être étiquetés avec le type **default_t**. SELinux empêche le serveur HTTP Apache (**httpd**) d'accéder à ces deux types. Pour autoriser l'accès, SELinux doit savoir que les fichiers de **/srv/myweb/** doivent être accessibles par **httpd**:

```
# semanage fcontext -a -t httpd_sys_content_t "/srv/myweb(/.*)?"
```

La commande **semanage** ajoute le contexte du répertoire **/srv/myweb/** et de tous les fichiers et répertoires qu'il contient à la configuration SELinux du contexte des fichiers. L'utilitaire **semanage** ne modifie pas le contexte. En tant que super-utilisateur, utilisez l'utilitaire **restorecon** pour appliquer les modifications :

```
# restorecon -R -v /srv/myweb
```

Contexte incorrect

L'utilitaire **matchpathcon** vérifie le contexte d'un chemin de fichier et le compare à l'étiquette par défaut pour ce chemin. L'exemple suivant illustre l'utilisation de **matchpathcon** sur un répertoire contenant des fichiers mal étiquetés :

```
$ matchpathcon -V /var/www/html/*
/var/www/html/index.html has context unconfined_u:object_r:user_home_t:s0, should be
system_u:object_r:httpd_sys_content_t:s0
/var/www/html/page1.html has context unconfined_u:object_r:user_home_t:s0, should be
system_u:object_r:httpd_sys_content_t:s0
```

Dans cet exemple, les fichiers **index.html** et **page1.html** sont étiquetés avec le type **user_home_t**. Ce type est utilisé pour les fichiers se trouvant dans les répertoires personnels des utilisateurs. L'utilisation de la commande **mv** pour déplacer des fichiers à partir de votre répertoire personnel peut entraîner l'étiquetage des fichiers avec le type **user_home_t**. Ce type de fichier ne devrait pas exister en dehors des répertoires personnels. Utilisez l'utilitaire **restorecon** pour rétablir le type correct de ces fichiers :

```
# restorecon -v /var/www/html/index.html
restorecon reset /var/www/html/index.html context unconfined_u:object_r:user_home_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Pour restaurer le contexte de tous les fichiers d'un répertoire, utilisez l'option **-R**:

```
# restorecon -R -v /var/www/html/
restorecon reset /var/www/html/page1.html context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /var/www/html/index.html context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Applications confinées configurées de manière non standard

Les services peuvent être exécutés de différentes manières. Pour tenir compte de cela, vous devez spécifier comment vous exécutez vos services. Pour ce faire, vous pouvez utiliser les booléens SELinux qui permettent de modifier certaines parties de la politique SELinux au moment de l'exécution. Cela permet d'effectuer des changements, tels que l'accès des services aux volumes NFS, sans recharger ou recompiler la politique SELinux. Par ailleurs, l'exécution de services sur des numéros de port autres que ceux par défaut nécessite la mise à jour de la configuration de la politique à l'aide de la commande **semanage**.

Par exemple, pour permettre au serveur HTTP Apache de communiquer avec MariaDB, activez le booléen **httpd_can_network_connect_db**:

```
# setsebool -P httpd_can_network_connect_db on
```

Notez que l'option **-P** rend le paramètre persistant à travers les redémarrages du système.

Si l'accès est refusé pour un service particulier, utilisez les utilitaires **getsebool** et **grep** pour voir s'il existe des booléens permettant d'autoriser l'accès. Par exemple, utilisez la commande **getsebool -a | grep ftp** pour rechercher les booléens relatifs à FTP :

```
$ getsebool -a | grep ftp
```

```
ftpd_anon_write --> off
ftpd_full_access --> off
ftpd_use_cifs --> off
ftpd_use_nfs --> off

ftpd_connect_db --> off
httpd_enable_ftp_server --> off
tftp_anon_write --> off
```

Pour obtenir une liste de booléens et savoir s'ils sont activés ou désactivés, utilisez la commande **getsebool -a**. Pour obtenir une liste de booléens avec leur signification et savoir s'ils sont activés ou désactivés, installez le paquet **selinux-policy-devel** et utilisez la commande **semanage boolean -l** en tant que root.

Numéros de port

En fonction de la configuration de la politique, les services ne peuvent être autorisés à fonctionner que sur certains numéros de port. Si vous tentez de modifier le port sur lequel un service s'exécute sans modifier la stratégie, le service risque de ne pas démarrer. Par exemple, exécutez la commande **semanage port -l | grep http** en tant que super-utilisateur pour dresser la liste des ports liés à **http**:

```
# semanage port -l | grep http
http_cache_port_t              tcp      3128, 8080, 8118
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 443, 488, 8008, 8009, 8443
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
```

Le type de port **http_port_t** définit les ports sur lesquels le serveur HTTP Apache peut écouter, qui dans ce cas sont les ports TCP 80, 443, 488, 8008, 8009 et 8443. Si un administrateur configure **httpd.conf** de sorte que **httpd** écoute sur le port 9876 (**Listen 9876**), mais que la stratégie n'est pas mise à jour pour refléter ce changement, la commande suivante échoue :

```
# systemctl start httpd.service
Job for httpd.service failed. See 'systemctl status httpd.service' and 'journalctl -xn' for details.

# systemctl status httpd.service
httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled)
   Active: failed (Result: exit-code) since Thu 2013-08-15 09:57:05 CEST; 59s ago
  Process: 16874 ExecStop=/usr/sbin/httpd $OPTIONS -k graceful-stop (code=exited,
status=0/SUCCESS)
  Process: 16870 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited,
status=1/FAILURE)
```

Un message de refus SELinux similaire au suivant est enregistré à l'adresse **/var/log/audit/audit.log**:

```
type=AVC msg=audit(1225948455.061:294): avc:  denied  { name_bind } for  pid=4997
comm="httpd" src=9876 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=system_u:object_r:port_t:s0 tclass=tcp_socket
```

Pour permettre à **httpd** d'écouter sur un port qui n'est pas répertorié pour le type de port **http_port_t**, utilisez la commande **semanage port** pour attribuer une étiquette différente au port :

```
# semanage port -a -t http_port_t -p tcp 9876
```

L'option **-a** ajoute un nouvel enregistrement, l'option **-t** définit un type et l'option **-p** définit un protocole. Le dernier argument est le numéro de port à ajouter.

Cas particuliers, applications évolutives ou défectueuses et systèmes compromis

Les applications peuvent contenir des bogues, ce qui conduit SELinux à refuser l'accès. De plus, les règles SELinux évoluent - SELinux peut ne pas avoir vu une application fonctionner d'une certaine manière, ce qui peut entraîner un refus d'accès, même si l'application fonctionne comme prévu. Par exemple, si une nouvelle version de PostgreSQL est publiée, elle peut effectuer des actions que la politique actuelle ne prend pas en compte, ce qui entraîne un refus d'accès, alors que l'accès devrait être autorisé.

Dans ce cas, lorsque l'accès est refusé, utilisez l'utilitaire **audit2allow** pour créer un module de politique personnalisé afin d'autoriser l'accès. Vous pouvez signaler les règles manquantes dans la politique SELinux dans le fichier **Red Hat Bugzilla**. Pour Red Hat Enterprise Linux 9, créez des bogues contre le produit **Red Hat Enterprise Linux 9** et sélectionnez le composant **selinux-policy**. Incluez la sortie des commandes **audit2allow -w -a** et **audit2allow -a** dans ces rapports de bogue.

Si une application demande des privilèges de sécurité importants, cela peut être un signal que l'application est compromise. Utilisez des outils de détection d'intrusion pour inspecter ces comportements suspects.

La page **Solution Engine** sur le portail client de Red Hat peut également fournir des conseils sous la forme d'un article contenant une solution possible pour le même problème ou un problème très similaire que vous rencontrez. Sélectionnez le produit et la version appropriés et utilisez les mots-clés liés à SELinux, tels que *selinux* ou *avc*, ainsi que le nom de votre service ou application bloqué(e), par exemple : **selinux samba**.

5.4. DÉNÉGATIONS SELINUX DANS LE JOURNAL D'AUDIT

Le système d'audit Linux stocke par défaut les entrées du journal dans le fichier **/var/log/audit/audit.log**.

Pour ne répertorier que les enregistrements liés à SELinux, utilisez la commande **ausearch** avec le paramètre "message type" fixé à **AVC** et **AVC_USER** au minimum, par exemple :

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR
```

Une entrée de refus SELinux dans le fichier journal d'audit peut se présenter comme suit :

```
type=AVC msg=audit(1395177286.929:1638): avc:  denied  { read } for  pid=6591 comm="httpd"
name="webpages" dev="0:37" ino=2112 scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:nfs_t:s0 tclass=dir
```

Les parties les plus importantes de cette entrée sont les suivantes :

- **avc: denied** - l'action effectuée par SELinux et enregistrée dans l'Access Vector Cache (AVC)
- **{ read }** - l'action refusée
- **pid=6591** - l'identificateur de processus du sujet qui a tenté d'effectuer l'action refusée
- **comm="httpd"** - le nom de la commande qui a été utilisée pour invoquer le processus analysé
- **httpd_t** - le type SELinux du processus

- **nfs_t** - le type SELinux de l'objet affecté par l'action du processus
- **tclass=dir** - la classe d'objet cible

L'entrée de journal précédente peut être traduite par :

*SELinux denied the **httpd** process with PID 6591 and the **httpd_t** type to read from a directory with the **nfs_t** type.*

Le message de refus SELinux suivant se produit lorsque le serveur HTTP Apache tente d'accéder à un répertoire étiqueté avec un type de la suite Samba :

```
type=AVC msg=audit(1226874073.147:96): avc:  denied  { getattr } for  pid=2465 comm="httpd"
path="/var/www/html/file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file
```

- **{ getattr }** - l'entrée **getattr** indique que le processus source essayait de lire les informations d'état du fichier cible. Cela se produit avant la lecture des fichiers. SELinux refuse cette action parce que le processus accède au fichier et qu'il n'a pas d'étiquette appropriée. Les permissions les plus courantes sont **getattr**, **read**, et **write**.
- **path="/var/www/html/file1"** - le chemin d'accès à l'objet (cible) auquel le processus a tenté d'accéder.
- **scontext="unconfined_u:system_r:httpd_t:s0"** - le contexte SELinux du processus (source) qui a tenté l'action refusée. Dans ce cas, il s'agit du contexte SELinux du serveur HTTP Apache, qui fonctionne avec le type **httpd_t**.
- **tcontext="unconfined_u:object_r:samba_share_t:s0"** - le contexte SELinux de l'objet (cible) auquel le processus a tenté d'accéder. Dans ce cas, il s'agit du contexte SELinux de **file1**.

Ce refus SELinux peut être traduit par :

*SELinux denied the **httpd** process with PID 2465 to access the **/var/www/html/file1** file with the **samba_share_t** type, which is not accessible to processes running in the **httpd_t** domain unless configured otherwise.*

Ressources supplémentaires

- **auditd(8)** et **ausearch(8)** pages de manuel

5.5. RESSOURCES SUPPLÉMENTAIRES

- Dépannage SELinux de base en CLI
- Qu'est-ce que SELinux essaie de me dire ? Les 4 causes principales des erreurs SELinux

CHAPITRE 6. UTILISATION DE LA SÉCURITÉ MULTINIVEAUX (MLS)

La politique de sécurité multiniveaux (MLS) utilise le site *levels*, tel qu'il a été conçu à l'origine par la communauté de la défense américaine. MLS répond à un ensemble très étroit d'exigences de sécurité basées sur la gestion de l'information dans des environnements rigoureusement contrôlés tels que l'armée.

L'utilisation de MLS est complexe et ne correspond pas bien aux scénarios d'utilisation généraux.

6.1. SÉCURITÉ MULTINIVEAUX (MLS)

La technologie MLS (Multi-Level Security) permet de classer les données de manière hiérarchique en utilisant des niveaux de sécurité de l'information, par exemple :

- [le plus bas] Non classé
- [bas] Confidentiel
- [haut] Secret
- [le plus haut] Top secret

Par défaut, la politique SELinux de MLS utilise 16 niveaux de sensibilité :

- **s0** est le moins sensible.
- **s15** est le plus sensible.

Le MLS utilise une terminologie spécifique pour aborder les niveaux de sensibilité :

- Les utilisateurs et les processus sont appelés **subjects**, dont le niveau de sensibilité est appelé **clearance**.
- Les fichiers, dispositifs et autres composants passifs du système sont appelés **objects**, dont le niveau de sensibilité est appelé **classification**.

Pour mettre en œuvre la MLS, SELinux utilise le modèle **Bell-La Padula Model** (BLP). Ce modèle spécifie comment l'information peut circuler dans le système en fonction des étiquettes attachées à chaque sujet et à chaque objet.

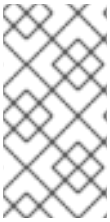
Le principe de base du BLP est "**No read up, no write down.**". Cela signifie que les utilisateurs ne peuvent lire que les fichiers dont le niveau de sensibilité est inférieur ou égal au leur, et que les données ne peuvent circuler que des niveaux inférieurs vers les niveaux supérieurs, et jamais l'inverse.

La politique SELinux MLS, qui est la mise en œuvre de MLS sur RHEL, applique un principe modifié appelé **Bell-La Padula with write equality**. Cela signifie que les utilisateurs peuvent lire des fichiers à leur propre niveau de sensibilité et à un niveau inférieur, mais qu'ils ne peuvent écrire qu'à leur propre niveau. Cela empêche, par exemple, les utilisateurs à faible habilitation d'écrire du contenu dans des fichiers top secret.

Par exemple, par défaut, un utilisateur ayant le niveau d'habilitation **s2**:

- Peut lire les fichiers avec les niveaux de sensibilité **s0**, **s1**, et **s2**.
- Impossible de lire les fichiers dont le niveau de sensibilité est égal ou supérieur à **s3**.

- Peut modifier des fichiers avec un niveau de sensibilité d'exactement **s2**.
- Impossible de modifier des fichiers dont le niveau de sensibilité est différent de **s2**.



NOTE

Les administrateurs de sécurité peuvent adapter ce comportement en modifiant la politique SELinux du système. Par exemple, ils peuvent autoriser les utilisateurs à modifier des fichiers à des niveaux inférieurs, ce qui augmente le niveau de sensibilité du fichier en fonction du niveau d'autorisation de l'utilisateur.

Dans la pratique, les utilisateurs se voient généralement attribuer une série de niveaux d'autorisation, par exemple **s1-s2**. Un utilisateur peut lire des fichiers dont le niveau de sensibilité est inférieur au niveau maximal de l'utilisateur, et écrire dans tous les fichiers situés dans cette fourchette.

Par exemple, par défaut, un utilisateur ayant une plage d'autorisation **s1-s2**:

- Peut lire des fichiers avec les niveaux de sensibilité **s0** et **s1**.
- Impossible de lire les fichiers dont le niveau de sensibilité est égal ou supérieur à **s2**.
- Peut modifier les fichiers avec le niveau de sensibilité **s1**.
- Impossible de modifier des fichiers dont le niveau de sensibilité est différent de **s1**.
- Peut modifier son propre niveau d'habilitation à **s2**.

Le contexte de sécurité d'un utilisateur non privilégié dans un environnement MLS est, par exemple, le suivant :

```
user_u:user_r:user_t:s1
```

Où ?

user_u
: est l'utilisateur SELinux.

user_r
: est le rôle SELinux.

user_t
: est le type SELinux.

s1
: est la gamme des niveaux de sensibilité MLS.

Le système combine toujours les règles d'accès MLS avec les autorisations classiques d'accès aux fichiers. Par exemple, si un utilisateur ayant un niveau de sécurité "Secret" utilise le contrôle d'accès discrétionnaire (DAC) pour bloquer l'accès à un fichier par d'autres utilisateurs, même les utilisateurs "Top Secret" ne peuvent pas accéder à ce fichier. Un niveau de sécurité élevé ne permet pas automatiquement à un utilisateur de parcourir l'ensemble du système de fichiers.

Les utilisateurs disposant d'une habilitation de haut niveau n'acquièrent pas automatiquement des droits administratifs sur les systèmes à plusieurs niveaux. Bien qu'ils puissent avoir accès à toutes les informations sensibles du système, ce n'est pas la même chose que d'avoir des droits administratifs.

En outre, les droits administratifs ne donnent pas accès aux informations sensibles. Par exemple, même si une personne se connecte en tant que **root**, elle ne peut toujours pas lire les informations top secrètes.

Vous pouvez ajuster davantage l'accès au sein d'un système MLS en utilisant des catégories. Avec la sécurité multi-catégories (MCS), vous pouvez définir des catégories telles que des projets ou des départements, et les utilisateurs ne seront autorisés à accéder qu'aux fichiers des catégories auxquelles ils sont affectés. Pour plus d'informations, voir Utilisation de la sécurité multi-catégories (MCS) pour la confidentialité des données.

6.2. RÔLES SELINUX DANS MLS

La politique SELinux associe chaque utilisateur Linux à un utilisateur SELinux. Cela permet aux utilisateurs Linux d'hériter des restrictions des utilisateurs SELinux.

IMPORTANT

La politique MLS ne contient pas le module **unconfined**, y compris les utilisateurs, les types et les rôles non limités. Par conséquent, les utilisateurs qui ne seraient pas confinés, y compris **root**, ne peuvent pas accéder à tous les objets et effectuer toutes les actions qu'ils pourraient effectuer dans la politique ciblée.

Vous pouvez personnaliser les autorisations pour les utilisateurs confinés dans votre politique SELinux en fonction de besoins spécifiques en ajustant les booléens dans la politique. Vous pouvez déterminer l'état actuel de ces booléens à l'aide de la commande **semanage boolean -l**. Pour dresser la liste de tous les utilisateurs SELinux, de leurs rôles SELinux et des niveaux et plages MLS/MCS, utilisez la commande **semanage user -l** comme **root**.

Tableau 6.1. Rôles des utilisateurs SELinux dans MLS

User	Rôle par défaut	Autres rôles
guest_u	**guest_r**	
xguest_u	**xguest_r**	
user_u	**user_r**	
staff_u	**staff_r**	**auditadm_r**
		secadm_r
		sysadm_r
		staff_r
sysadm_u	**sysadm_r**	
root	**staff_r**	**auditadm_r**
		secadm_r

User	Rôle par défaut	Autres rôles
		sysadm_r
		system_r
system_u	**system_r**	

Notez que **system_u** est une identité d'utilisateur spéciale pour les processus et les objets du système, et que **system_r** est le rôle associé. Les administrateurs ne doivent jamais associer cet utilisateur **system_u** et le rôle **system_r** à un utilisateur Linux. Par ailleurs, **unconfined_u** et **root** sont des utilisateurs non restreints. Pour ces raisons, les rôles associés à ces utilisateurs SELinux ne sont pas inclus dans le tableau suivant Types et accès aux rôles SELinux.

Chaque rôle SELinux correspond à un type SELinux et fournit des droits d'accès spécifiques.

Tableau 6.2. Types et accès aux rôles SELinux dans MLS

Rôle	Type	Connexion à l'aide du système X Window	**su** et **sudo**	Exécuter dans le répertoire personnel et à l'adresse /**tmp** (par défaut)	Mise en réseau
guest_r	**guest_t**	non	non	yes	non
xguest_r	**xguest_t**	yes	non	yes	navigateurs web uniquement (Firefox, GNOME Web)
user_r	**user_t**	yes	non	yes	yes
staff_r	**staff_t**	yes	seulement **sudo**	yes	yes
auditadm_r	**auditadm_t**		yes	yes	yes
secadm_r	**secadm_t**		yes	yes	yes
sysadm_r	**sysadm_t**	uniquement lorsque le booléen **xdm_sysadm_login** est **on**	yes	yes	yes

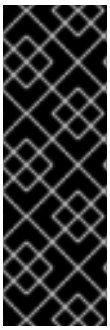
- Par défaut, le rôle **sysadm_r** a les droits du rôle **secadm_r**, ce qui signifie qu'un utilisateur ayant le rôle **sysadm_r** peut gérer la politique de sécurité. Si cela ne correspond pas à votre cas d'utilisation, vous pouvez séparer les deux rôles en désactivant le module **sysadm_secadm** dans la politique. Pour plus d'informations, voir

Séparation de l'administration du système et de l'administration de la sécurité dans MLS

- Les rôles sans login **dbadm_r**, **logadm_r**, et **webadm_r** peuvent être utilisés pour un sous-ensemble de tâches administratives. Par défaut, ces rôles ne sont associés à aucun utilisateur SELinux.

6.3. PASSAGE DE LA POLITIQUE SELINUX À MLS

Procédez comme suit pour passer d'une politique SELinux ciblée à une politique de sécurité multiniveaux (MLS).



IMPORTANT

Red Hat ne recommande pas l'utilisation de la stratégie MLS sur un système qui exécute le système X Window. De plus, lorsque vous ré-étiqueter le système de fichiers avec des étiquettes MLS, le système peut empêcher l'accès à des domaines confinés, ce qui empêche votre système de démarrer correctement. Veillez donc à passer SELinux en mode permissif avant de ré-étiqueter les fichiers. Sur la plupart des systèmes, vous verrez beaucoup de refus SELinux après le passage à MLS, et beaucoup d'entre eux ne sont pas triviaux à corriger.

Procédure

1. Installez le paquetage **selinux-policy-mls**:

```
# dnf install selinux-policy-mls
```

2. Ouvrez le fichier **/etc/selinux/config** dans un éditeur de texte de votre choix, par exemple :

```
# vi /etc/selinux/config
```

3. Changez le mode SELinux de "enforcing" à "permissive" et passez de la politique ciblée à la politique MLS :

```
SELINUX=permissive
SELINUXTYPE=mls
```

   Enregistrez les modifications et quittez l'éditeur.

4. Avant d'activer la stratégie MLS, vous devez attribuer une étiquette MLS à chaque fichier du système de fichiers :

```
# fixfiles -F onboot
System will relabel on next boot
```

5. Redémarrer le système :

```
# reboot
```

6. Vérifier les refus SELinux :

```
# ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent -i
```

La commande précédente ne couvrant pas tous les cas de figure, reportez-vous à la section Dépannage des problèmes liés à SEL inux pour obtenir des conseils sur l'identification, l'analyse et la résolution des refus SELinux.

7. Après vous être assuré qu'il n'y a pas de problèmes liés à SELinux sur votre système, remettez SELinux en mode "enforcing" en modifiant l'option correspondante dans **/etc/selinux/config**:

```
SELINUX=enforcing
```

8. Redémarrer le système :

```
# reboot
```

IMPORTANT

Si votre système ne démarre pas ou si vous ne pouvez pas vous connecter après avoir basculé vers MLS, ajoutez le paramètre **enforcing=0** à la ligne de commande de votre noyau. Pour plus d'informations, voir Modifier les modes SELinux au démarrage .

Notez également que dans MLS, les connexions SSH en tant qu'utilisateur **root** associé au rôle SELinux **sysadm_r** diffèrent des connexions en tant que **root** dans **staff_r**. Avant de démarrer votre système dans MLS pour la première fois, envisagez d'autoriser les connexions SSH en tant que **sysadm_r** en définissant le booléen SELinux **ssh_sysadm_login** sur **1**. Pour activer **ssh_sysadm_login** ultérieurement, alors que vous êtes déjà dans MLS, vous devez vous connecter en tant que **root** dans **staff_r**, passer à **root** dans **sysadm_r** à l'aide de la commande **newrole -r sysadm_r**, puis définir le booléen sur **1**.

Vérification

1. Vérifiez que SELinux fonctionne en mode d'exécution :

```
# getenforce
Enforcing
```

2. Vérifiez que le statut de SELinux renvoie la valeur **mls**:

```
# sestatus | grep mls
Loaded policy name:             mls
```

Ressources supplémentaires

- Les pages de manuel **fixfiles(8)**, **setsebool(8)**, et **ssh_selinux(8)**.

6.4. ÉTABLISSEMENT DE L'AUTORISATION DE L'UTILISATEUR DANS LE SYSTÈME MLS

Après avoir basculé la politique SELinux sur MLS, vous devez attribuer des niveaux d'habilitation de sécurité aux utilisateurs en les associant à des utilisateurs SELinux confinés. Par défaut, un utilisateur ayant une habilitation de sécurité donnée :

- Impossible de lire des objets ayant un niveau de sensibilité plus élevé.
- Impossible d'écrire sur des objets ayant un niveau de sensibilité différent.

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Le mode SELinux est défini sur **enforcing**.
- Le paquet **policycoreutils-python-utils** est installé.
- Un utilisateur assigné à un utilisateur confiné SELinux :
  - Pour un utilisateur non privilégié, assigné à **user_u** (*example_user* dans la procédure suivante).
  - Pour un utilisateur privilégié, affecté à **staff_u** (*staff* dans la procédure suivante) .

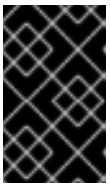
NOTE

Assurez-vous que les utilisateurs ont été créés lorsque la politique MLS était active. Les utilisateurs créés dans d'autres politiques SELinux ne peuvent pas être utilisés dans MLS.

Procédure

1. Facultatif : Pour éviter d'ajouter des erreurs à votre politique SELinux, passez au mode SELinux **permissive**, qui facilite le dépannage :

```
# setenforce 0
```



IMPORTANT

En mode permissif, SELinux n'applique pas la politique active mais enregistre uniquement les messages AVC (Access Vector Cache), qui peuvent ensuite être utilisés à des fins de dépannage et de débogage.

2. Définir une plage d'autorisation pour l'utilisateur SELinux **staff_u**. Par exemple, cette commande définit la plage d'autorisation de **s1** à **s15**, **s1** étant le niveau d'autorisation par défaut :

```
# semanage user -m -L s1 -r s1-s15 _staff_u
```

3. Génère des entrées de configuration du contexte de fichier SELinux pour les répertoires personnels des utilisateurs :

```
# genhomedircon
```

4. Rétablir les contextes de sécurité des fichiers par défaut :

```
# restorecon -R -F -v /home/
Relabeled /home/staff from staff_u:object_r:user_home_dir_t:s0 to
staff_u:object_r:user_home_dir_t:s1
Relabeled /home/staff/.bash_logout from staff_u:object_r:user_home_t:s0 to
staff_u:object_r:user_home_t:s1
Relabeled /home/staff/.bash_profile from staff_u:object_r:user_home_t:s0 to
staff_u:object_r:user_home_t:s1
Relabeled /home/staff/.bashrc from staff_u:object_r:user_home_t:s0 to
staff_u:object_r:user_home_t:s1
```

5. Attribuer un niveau d'autorisation à l'utilisateur :

```
# semanage login -m -r s1 example_user
```

Où ***s1*** est le niveau d'habilitation attribué à l'utilisateur.

6. Adapter le répertoire personnel de l'utilisateur au niveau d'autorisation de l'utilisateur :

```
# chcon -R -l s1 /home/example_user
```

7. Facultatif : si vous êtes passé au mode SELinux **permissive**, et après avoir vérifié que tout fonctionne comme prévu, repassez au mode SELinux **enforcing**:

```
# setenforce 1
```

Verification steps

1. Vérifiez que l'utilisateur est associé au bon utilisateur SELinux et que le niveau d'autorisation attribué est correct :

```
# semanage login -l
Login Name      SELinux User         MLS/MCS Range        Service
__default__     user_u               s0-s0                *
example_user    user_u               s1                   *
...
```

2. Connectez-vous en tant qu'utilisateur dans MLS.

3. Vérifiez que le niveau de sécurité de l'utilisateur fonctionne correctement :

IMPORTANT

Les fichiers utilisés pour la vérification ne doivent pas contenir d'informations sensibles, au cas où la configuration serait incorrecte et où l'utilisateur pourrait accéder aux fichiers sans autorisation.

   a. Vérifiez que l'utilisateur ne peut pas lire un fichier dont le niveau de sensibilité est plus élevé.

   b. Vérifiez que l'utilisateur peut écrire dans un fichier avec la même sensibilité.

   c. Vérifiez que l'utilisateur peut lire un fichier dont le niveau de sensibilité est inférieur.

Ressources supplémentaires

- Section 6.3, « Passage de la politique SELinux à MLS » .
- Section 3.5, « Ajout d'un nouvel utilisateur en tant qu'utilisateur défini par SELinux » .
- Chapitre 2, *Modifier les états et les modes SELinux*.
- Chapitre 5, *Résolution des problèmes liés à SELinux* .
- L'article de la base de connaissances " Basic SELinux Troubleshooting in CLI ".

6.5. MODIFICATION DU NIVEAU D'HABILITATION D'UN UTILISATEUR DANS LA PLAGE DE SÉCURITÉ DÉFINIE DANS MLS

En tant qu'utilisateur du système de sécurité multiniveaux (MLS), vous pouvez modifier votre niveau d'habilitation actuel à l'intérieur de la fourchette qui vous a été attribuée par l'administrateur. Vous ne pouvez jamais dépasser la limite supérieure de votre fourchette ni réduire votre niveau en dessous de la limite inférieure de votre fourchette. Cela vous permet, par exemple, de modifier des fichiers moins sensibles sans augmenter leur niveau de sensibilité jusqu'à votre niveau d'habilitation le plus élevé.

Par exemple, en tant qu'utilisateur affecté à la plage **s1-s3**:

- Vous pouvez passer aux niveaux **s1**, **s2**, et **s3**.
- Vous pouvez passer aux plages **s1-s2**, et **s2-s3**.
- Vous ne pouvez pas passer aux plages **s0-s3** ou **s1-s4**.

NOTE

Le passage à un niveau différent ouvre un nouveau shell avec un niveau d'autorisation différent. Cela signifie que vous ne pouvez pas revenir à votre niveau d'habilitation initial de la même manière que si vous le diminuiez. Cependant, vous pouvez toujours revenir au shell précédent en entrant **exit**.

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Le mode SELinux est défini sur **enforcing**.
- Vous pouvez vous connecter en tant qu'utilisateur assigné à une série de niveaux d'habilitation MLS.

Procédure

1. Se connecter en tant qu'utilisateur à partir d'un terminal sécurisé.

NOTE

Les terminaux sécurisés sont définis dans le fichier **/etc/selinux/mls/contexts/securetty_types**. Par défaut, la console est un terminal sécurisé, mais pas SSH.

2. Vérifier le contexte de sécurité de l'utilisateur actuel :

-

```
$ id -Z
user_u:user_r:user_t:s0-s2
```

Dans cet exemple, l'utilisateur est affecté à l'utilisateur SELinux **user_u**, au rôle **user_r**, au type **user_t** et à la plage de sécurité MLS **s0-s2**.

3. Vérifier le contexte de sécurité de l'utilisateur actuel :

```
$ id -Z
user_u:user_r:user_t:s1-s2
```

4. Passer à une autre plage d'habilitation de sécurité dans la plage d'habilitation de l'utilisateur :

```
$ newrole -l s1
```

Vous pouvez passer à n'importe quelle plage dont le maximum est inférieur ou égal à la plage qui vous a été attribuée. La saisie d'une plage à niveau unique modifie la limite inférieure de la plage attribuée. Par exemple, la saisie de **newrole -l s1** en tant qu'utilisateur ayant une plage **s0-s2** équivaut à la saisie de **newrole -l s1-s2**.

Vérification

1. Affiche le contexte de sécurité de l'utilisateur actuel :

```
$ id -Z
user_u:user_r:user_t:s1-s2
```

2. Retourner à l'interpréteur de commandes précédent avec la portée originale en mettant fin à l'interpréteur de commandes actuel :

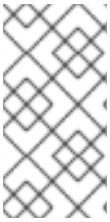
```
$ exit
```

Ressources supplémentaires

- Section 6.4, « Établissement de l'autorisation de l'utilisateur dans le système MLS »
- **newrole(1)** page de manuel
- **securetty_types(5)** page de manuel

6.6. AUGMENTATION DES NIVEAUX DE SENSIBILITÉ DES FICHIERS DANS LE SYSTÈME MLS

Par défaut, les utilisateurs du système de sécurité multiniveau (MLS) ne peuvent pas augmenter le niveau de sensibilité des fichiers. Toutefois, l'administrateur de sécurité (**secadm_r**) peut modifier ce comportement par défaut pour permettre aux utilisateurs d'augmenter la sensibilité des fichiers en ajoutant le module local **mlsfilewrite** à la politique SELinux du système. Ensuite, les utilisateurs assignés au type SELinux défini dans le module de politique peuvent augmenter les niveaux de classification des fichiers en les modifiant. Chaque fois qu'un utilisateur modifie un fichier, le niveau de sensibilité du fichier augmente jusqu'à la valeur inférieure de la plage de sécurité actuelle de l'utilisateur.



NOTE

L'administrateur de sécurité, lorsqu'il est connecté en tant qu'utilisateur ayant le rôle **secadm_r**, peut modifier les niveaux de sécurité des fichiers à l'aide de la commande **chcon -l *s0 /path/to/file*** pour modifier les niveaux de sécurité des fichiers. Pour plus d'informations, voir Section 6.7, « Modification de la sensibilité des fichiers dans MLS »

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Le mode SELinux est défini sur **enforcing**.
- Le paquet **policycoreutils-python-utils** est installé.
- Le module local **mlsfilewrite** est installé dans la politique SELinux MLS.
- Vous êtes connecté en tant qu'utilisateur dans MLS qui est :
  - Assigné à une plage de sécurité définie. Cet exemple montre un utilisateur avec une plage de sécurité **s0-s2**.
  - Assigné au même type SELinux que celui défini dans le module **mlsfilewrite**. Cet exemple nécessite le module **(typeattributeset mlsfilewrite (user_t))**.

Procédure

1. Facultatif : Affiche le contexte de sécurité de l'utilisateur actuel :

```
$ id -Z
user_u:user_r:user_t:s0-s2
```

2. Modifiez le niveau inférieur de l'habilitation MLS de l'utilisateur pour qu'il corresponde au niveau que vous souhaitez attribuer au dossier :

```
$ newrole -l s1-s2
```

3. Facultatif : Affiche le contexte de sécurité de l'utilisateur actuel :

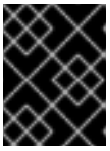
```
$ id -Z
user_u:user_r:user_t:s1-s2
```

4. Facultatif : Afficher le contexte de sécurité du fichier :

```
$ ls -Z /path/to/file
user_u:object_r:user_home_t:s0 /path/to/file
```

5. Modifier le niveau de sensibilité du fichier pour qu'il corresponde au niveau le plus bas de l'habilitation de l'utilisateur en modifiant le fichier :

```
$ touch /path/to/file
```



IMPORTANT

Le niveau de classification reprend sa valeur par défaut si la commande **restorecon** est utilisée sur le système.

6. Facultatif : Quitter le shell pour revenir à la plage de sécurité précédente de l'utilisateur :

```
$ exit
```

Vérification

- Affiche le contexte de sécurité du fichier :

```
$ ls -Z /path/to/file
user_u:object_r:user_home_t:s1 /path/to/file
```

Ressources supplémentaires

- Section 6.10, « Permettre aux utilisateurs MLS de modifier les fichiers aux niveaux inférieurs » .

6.7. MODIFICATION DE LA SENSIBILITÉ DES FICHIERS DANS MLS

Dans la politique SELinux de MLS, les utilisateurs ne peuvent modifier que les fichiers de leur propre niveau de sensibilité. Cela permet d'éviter que des informations très sensibles ne soient exposées à des utilisateurs ayant un niveau d'habilitation inférieur et que des utilisateurs ayant un niveau d'habilitation inférieur ne créent des documents très sensibles. Les administrateurs peuvent toutefois augmenter manuellement la classification d'un fichier, par exemple pour que le fichier soit traité à un niveau supérieur.

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Le mode SELinux est défini sur "enforcing".
- Vous avez des droits d'administration de la sécurité, ce qui signifie que vous êtes assigné à l'une ou l'autre des deux fonctions :
  - Le rôle de **secadm_r**.
  - Si le module **sysadm_secadm** est activé, au rôle **sysadm_r**. Le module **sysadm_secadm** est activé par défaut.
- Le paquet **policycoreutils-python-utils** est installé.
- Un utilisateur assigné à n'importe quel niveau d'autorisation. Pour plus d'informations, voir Établissement des niveaux d'autorisation des utilisateurs dans MLS.
  Dans cet exemple, ***User1*** a un niveau d'habilitation **s1**.
- Fichier auquel un niveau de classification a été attribué et auquel vous avez accès.
  Dans cet exemple, ***/path/to/file*** a le niveau de classification **s1**.

Procédure

1. Vérifier le niveau de classification du fichier :

```
# ls -lZ /path/to/file
-rw-r-----. 1 User1 User1 user_u:object_r:user_home_t:s1 0 12. Feb 10:43 /path/to/file
```

2. Modifier le niveau de classification par défaut du fichier :

```
# semanage fcontext -a -r s2 /path/to/file
```

3. Force le changement d'étiquette du contexte SELinux du fichier :

```
# restorecon -F -v /path/to/file
Relabeled /path/to/file from user_u:object_r:user_home_t:s1 to
user_u:object_r:user_home_t:s2
```

Vérification

1. Vérifier le niveau de classification du fichier :

```
# ls -lZ /path/to/file
-rw-r-----. 1 User1 User1 user_u:object_r:user_home_t:s2 0 12. Feb 10:53 /path/to/file
```

2. Facultatif : Vérifiez que l'utilisateur ayant un niveau d'habilitation inférieur ne peut pas lire le fichier :

```
$ cat /path/to/file
cat: file: Permission denied
```

Ressources supplémentaires

- Section 6.4, « Établissement de l'autorisation de l'utilisateur dans le système MLS » .

6.8. SÉPARER L'ADMINISTRATION DU SYSTÈME DE L'ADMINISTRATION DE LA SÉCURITÉ DANS MLS

Par défaut, le rôle **sysadm_r** a les droits du rôle **secadm_r**, ce qui signifie qu'un utilisateur ayant le rôle **sysadm_r** peut gérer la politique de sécurité. Si vous avez besoin de plus de contrôle sur les autorisations de sécurité, vous pouvez séparer l'administration du système de l'administration de la sécurité en attribuant à un utilisateur Linux le rôle **secadm_r** et en désactivant le module **sysadm_secadm** dans la stratégie SELinux.

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Le mode SELinux est défini sur **enforcing**.
- Le paquet **policycoreutils-python-utils** est installé.
- Un utilisateur Linux auquel sera attribué le rôle **secadm_r**:
  - L'utilisateur est assigné à l'utilisateur SELinux **staff_u**

- Un mot de passe a été défini pour cet utilisateur.

AVERTISSEMENT

Assurez-vous que vous pouvez vous connecter en tant qu'utilisateur auquel sera attribué le rôle **secadm**. Si ce n'est pas le cas, vous pouvez empêcher toute modification future de la politique SELinux du système.

Procédure

1. Créez un nouveau fichier **sudoers** dans le répertoire **/etc/sudoers.d** pour l'utilisateur :

```
# visudo -f /etc/sudoers.d/<sec_adm_user>
```

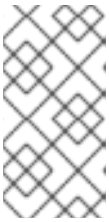
Pour que les fichiers **sudoers** restent organisés, remplacez ***<sec_adm_user>*** par l'utilisateur Linux auquel sera attribué le rôle **secadm**.

2. Ajoutez le contenu suivant dans le fichier **/etc/sudoers.d/*<sec_adm_user>*** le contenu suivant :

```
<sec_adm_user> ALL=(ALL) TYPE=secadm_t ROLE=secadm_r ALL
```

Cette ligne autorise ***<secadmuser>*** sur tous les hôtes à exécuter toutes les commandes, et attribue à l'utilisateur le type et le rôle SELinux **secadm** par défaut.

3. Connectez-vous en tant qu'utilisateur de *<sec_adm_user>*:



NOTE

Pour s'assurer que le contexte SELinux (qui comprend l'utilisateur SELinux, le rôle et le type) est modifié, connectez-vous en utilisant **ssh**, la console ou **xdm**. Les autres moyens, tels que **su** et **sudo**, ne peuvent pas modifier l'ensemble du contexte SELinux.

4. Vérifier le contexte de sécurité de l'utilisateur :

```
$ id
uid=1000(<sec_adm_user>) gid=1000(<sec_adm_user>) groups=1000(<sec_adm_user>)
context=staff_u:staff_r:staff_t:s0-s15:c0.c1023
```

5. Exécute l'interpréteur de commandes interactif pour l'utilisateur root :

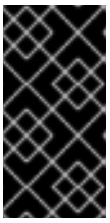
```
$ sudo -i
[sudo] password for <sec_adm_user>:
```

6. Vérifier le contexte de sécurité de l'utilisateur actuel :

```
# id
uid=0(root) gid=0(root) groups=0(root) context=staff_u:secadm_r:secadm_t:s0-s15:c0.c1023
```

7. Désactiver le module **sysadm_secadm** de la politique :

```
# semodule -d sysadm_secadm
```



IMPORTANT

Utilisez la commande **semodule -d** au lieu de supprimer le module de politique du système à l'aide de la commande **semodule -r**. La commande **semodule -r** supprime le module du stockage de votre système, ce qui signifie qu'il ne peut pas être chargé à nouveau sans réinstaller le paquetage **selinux-policy-mls**.

Vérification

1. En tant qu'utilisateur assigné au rôle **secadm** et dans le shell interactif de l'utilisateur root, vérifiez que vous pouvez accéder aux données de la politique de sécurité :

```
# seinfo -xt secadm_t

Types: 1
   type secadm_t, can_relabelto_shadow_passwords, (...) userdomain;
```

2. Se déconnecter de l'interpréteur de commandes racine :

```
# logout
```

3. Se déconnecter de l'utilisateur ***<sec_adm_user>*** l'utilisateur :

```
$ logout
Connection to localhost closed.
```

4. Affiche le contexte de sécurité actuel :

```
# id
uid=0(root) gid=0(root) groups=0(root) context=root:sysadm_r:sysadm_t:s0-s15:c0.c1023
```

5. Tentative d'activation du module **sysadm_secadm**. La commande devrait échouer :

```
# semodule -e sysadm_secadm
SELinux:  Could not load policy file /etc/selinux/mls/policy/policy.31:  Permission denied
/sbin/load_policy:  Can't load policy:  Permission denied
libsemanage.semanage_reload_policy: load_policy returned error code 2. (No such file or
directory).
SELinux:  Could not load policy file /etc/selinux/mls/policy/policy.31:  Permission denied
/sbin/load_policy:  Can't load policy:  Permission denied
libsemanage.semanage_reload_policy: load_policy returned error code 2. (No such file or
directory).
semodule:  Failed!
```

6. Tentative d'affichage des détails concernant le type SELinux **sysadm_t**. La commande doit échouer :

```
# seinfo -xt sysadm_t
[Errno 13] Permission denied: '/sys/fs/selinux/policy'
```

-

6.9. DÉFINITION D'UN TERMINAL SÉCURISÉ EN MLS

La politique SELinux vérifie le type de terminal à partir duquel un utilisateur est connecté et n'autorise l'exécution de certaines applications SELinux, par exemple **newrole**, qu'à partir de terminaux sécurisés. Si l'on tente de le faire à partir d'un terminal non sécurisé, on obtient une erreur : **Error: you are not allowed to change levels on a non secure terminal;**.

Le fichier **/etc/selinux/mls/contexts/securetty_types** définit les terminaux sécurisés pour la politique de sécurité multiniveaux (MLS).

Contenu par défaut du fichier :

```
console_device_t
sysadm_tty_device_t
user_tty_device_t
staff_tty_device_t
auditadm_tty_device_t
secureadm_tty_device_t
```

AVERTISSEMENT

L'ajout de types de terminaux à la liste des terminaux sécurisés peut exposer votre système à des risques de sécurité.

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Vous êtes connecté à partir d'un terminal déjà sécurisé, ou SELinux est en mode permissif.
- Vous avez des droits d'administration de la sécurité, ce qui signifie que vous êtes assigné à l'une ou l'autre des deux fonctions :
  - Le rôle de **secadm_r**.
  - Si le module **sysadm_secadm** est activé, au rôle **sysadm_r**. Le module **sysadm_secadm** est activé par défaut.
- Le paquet **policycoreutils-python-utils** est installé.

Procédure

1. Déterminer le type de borne de courant :

```
# ls -Z `tty`
root:object_r:user_devpts_t:s0 /dev/pts/0
```

Dans cet exemple, **user_devpts_t** est le type de terminal actuel.

2. Ajoutez le type SELinux approprié sur une nouvelle ligne du fichier **/etc/selinux/mls/contexts/securetty_types**.

3. Optionnel : Passer SELinux en mode d'application :

```
# setenforce 1
```

Vérification

- Connectez-vous à partir du terminal non sécurisé que vous avez ajouté au fichier **/etc/selinux/mls/contexts/securetty_types**.

Ressources supplémentaires

- **securetty_types(5)** page de manuel

6.10. PERMETTRE AUX UTILISATEURS MLS DE MODIFIER LES FICHIERS AUX NIVEAUX INFÉRIEURS

Par défaut, les utilisateurs MLS ne peuvent pas écrire dans des fichiers dont le niveau de sensibilité est inférieur à la valeur la plus basse de la plage d'autorisation. Si votre scénario exige que vous autorisiez les utilisateurs à modifier des fichiers à des niveaux inférieurs, vous pouvez le faire en créant un module SELinux local. Toutefois, l'écriture dans un fichier augmentera son niveau de sensibilité jusqu'à la valeur inférieure de la plage actuelle de l'utilisateur.

Conditions préalables

- La politique SELinux est définie sur **mls**.
- Le mode SELinux est défini sur **enforcing**.
- Le paquet **policycoreutils-python-utils** est installé.
- Les paquets **setools-console** et **audit** pour vérification.

Procédure

1. En option : Passer en mode permissif pour faciliter le dépannage.

```
# setenforce 0
```

2. Ouvrez un nouveau fichier **.cil** avec un éditeur de texte, par exemple **~/local_mlsfilewrite.cil**, et insérez la règle personnalisée suivante :

```
(typeattributeset mlsfilewrite (_staff_t_))
```

   Vous pouvez remplacer ***staff_t*** par un type SELinux différent. En spécifiant le type SELinux ici, vous pouvez contrôler quels rôles SELinux peuvent éditer des fichiers de niveau inférieur.

   Pour mieux organiser vos modules locaux, utilisez le préfixe **local_** dans les noms des modules de politique SELinux locaux.

3. Installer le module de politique :

```
# semodule -i ~/local_mlsfilewrite.cil
```

NOTE

Pour supprimer le module de politique locale, utilisez **semodule -r ~/*local_mlsfilewrite***. Notez que vous devez vous référer au nom du module sans le suffixe **.cil**.

4. Facultatif : si vous avez précédemment rétabli le mode permissif, revenez au mode exécutoire :

```
# setenforce 1
```

Vérification

1. Trouver le module local dans la liste des modules SELinux installés :

```
# semodule -lfull | grep "local_mls"
400 local_mlsfilewrite  cil
```

Comme les modules locaux ont la priorité **400**, vous pouvez également les lister en utilisant la commande **semodule -lfull | grep -v ^100**.

2. Se connecter en tant qu'utilisateur assigné au type défini dans la règle personnalisée, par exemple, ***staff_t***.

3. Tenter d'écrire dans un fichier dont le niveau de sensibilité est inférieur. Le niveau de classification du fichier est alors porté au niveau d'habilitation de l'utilisateur.

IMPORTANT

Les fichiers utilisés pour la vérification ne doivent pas contenir d'informations sensibles, au cas où la configuration serait incorrecte et où l'utilisateur pourrait accéder aux fichiers sans autorisation.

CHAPITRE 7. UTILISATION DE LA SÉCURITÉ MULTI-CATÉGORIES (MCS) POUR LA CONFIDENTIALITÉ DES DONNÉES

Vous pouvez utiliser MCS pour améliorer la confidentialité des données de votre système en classant les données par catégories, puis en accordant à certains processus et utilisateurs l'accès à des catégories spécifiques

7.1. SÉCURITÉ MULTI-CATÉGORIES (MCS)

La sécurité multi-catégories (MCS) est un mécanisme de contrôle d'accès qui utilise des catégories attribuées aux processus et aux fichiers. Les fichiers ne sont alors accessibles que par les processus affectés aux mêmes catégories. L'objectif de la sécurité multi-catégories est de maintenir la confidentialité des données sur votre système.

Les catégories du SCM sont définies par les valeurs **c0** à **c1023**, mais vous pouvez également définir un libellé pour chaque catégorie ou combinaison de catégories, par exemple "Personnel", "ProjectX" ou "ProjectX.Personnel". Le service de traduction MCS (**mcstrans**) remplace alors les valeurs des catégories par les étiquettes appropriées dans les entrées et sorties du système, de sorte que les utilisateurs puissent utiliser ces étiquettes au lieu des valeurs des catégories.

Lorsque les utilisateurs sont affectés à des catégories, ils peuvent étiqueter n'importe lequel de leurs fichiers avec n'importe laquelle des catégories auxquelles ils ont été affectés.

Le MCS fonctionne selon un principe simple : pour accéder à un fichier, un utilisateur doit être assigné à toutes les catégories qui ont été assignées au fichier. Le contrôle MCS est appliqué après les règles Linux normales de contrôle d'accès discrétionnaire (DAC) et d'application de type SELinux (TE), de sorte qu'il ne peut que restreindre davantage la configuration de sécurité existante.

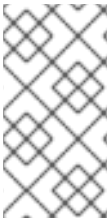
MCS au sein de Multi-Level Security

Vous pouvez utiliser MCS seul en tant que système non hiérarchique ou en combinaison avec Multi-Level Security (MLS) en tant que couche non hiérarchique au sein d'un système hiérarchique.

Un exemple de SCM au sein de MLS pourrait être une organisation de recherche secrète, où les dossiers sont classés comme suit :

Tableau 7.1. Exemple de combinaisons de niveaux et de catégories de sécurité

Security level	Category			
	Non spécifié	Projet X	Projet Y	Projet Z
Non classé	**s0**	**s0:c0**	**s0:c1**	**s0:c2**
Confidentiel	**s1**	**s1:c0**	**s1:c1**	**s1:c2**
Secret	**s2**	**s2:c0**	**s2:c1**	**s2:c2**
Top secret	**s3**	**s3:c0**	**s3:c1**	**s3:c2**



NOTE

Un utilisateur disposant d'une plage **s0:c0.1023** pourra accéder à tous les fichiers affectés à toutes les catégories du niveau **s0**, à moins que l'accès ne soit interdit par d'autres mécanismes de sécurité, tels que le CED ou les règles de politique d'application des types.

Le contexte de sécurité d'un fichier ou d'un processus est une combinaison des éléments suivants :

- Utilisateur SELinux
- Rôle SELinux
- Type SELinux
- Niveau de sensibilité MLS
- Catégorie MCS

Par exemple, un utilisateur non privilégié ayant accès au niveau de sensibilité 1 et à la catégorie 2 dans un environnement MLS/MCS pourrait avoir le contexte SELinux suivant :

```
user_u:user_r:user_t:s1:c2
```

Ressources supplémentaires

- Utilisation de la sécurité multiniveaux (MLS) .

7.2. CONFIGURATION DE LA SÉCURITÉ MULTI-CATÉGORIES POUR LA CONFIDENTIALITÉ DES DONNÉES

Par défaut, la MCS est active dans les politiques SELinux **targeted** et **mls** mais n'est pas configurée pour les utilisateurs. Dans la politique **targeted**, MCS n'est configuré que pour :

- OpenShift
- virt
- bac à sable
- étiquetage du réseau
- conteneurs (**container-selinux**)

Vous pouvez configurer MCS pour catégoriser les utilisateurs en créant un module SELinux local avec une règle qui limite le type SELinux **user_t** par des règles MCS en plus de l'application du type.

AVERTISSEMENT

La modification des catégories de certains fichiers peut rendre certains services non opérationnels. Si vous n'êtes pas un expert, contactez votre représentant commercial Red Hat et demandez des services de conseil.

Conditions préalables

- Le mode SELinux est défini sur **enforcing**.
- La politique SELinux est définie sur **targeted** ou **mls**.
- Les paquets **policycoreutils-python-utils** et **setools-console** sont installés.

Procédure

1. Créez un nouveau fichier nommé, par exemple, **local_mcs_user.cil**:

```
# vim local_mcs_user.cil
```

2. Insérer la règle suivante :

```
(typeattributeset mcs_constrained_type (user_t))
```

3. Installer le module de politique :

```
# semodule -i local_mcs_user.cil
```

Vérification

- Pour chaque domaine d'utilisateur, afficher des détails supplémentaires pour tous les composants :

```
# seinfo -xt user_t

Types: 1
type user_t, application_domain_type, nsswitch_domain, corenet_unlabeled_type, domain,
kernel_system_state_reader, mcs_constrained_type, netlabel_peer_type, privfd,
process_user_target, scsi_generic_read, scsi_generic_write, syslog_client_type,
pcmcia_typeattr_1, user_usertype, login_userdomain, userdomain, unpriv_userdomain,
userdom_home_reader_type, userdom_filetrans_type, xdmhomewriter, x_userdomain,
x_domain, dridomain, xdrawable_type, xcolormap_type;
```

Ressources supplémentaires

- Création d'un module local de politique SELinux

- Pour plus d'informations sur le MCS dans le contexte des conteneurs, consultez les articles de blog Comment SELinux sépare les conteneurs à l'aide de Multi-Level Security et Pourquoi vous devriez utiliser Multi-Category Security pour vos conteneurs Linux.

7.3. DÉFINITION DES ÉTIQUETTES DE CATÉGORIES DANS MCS

Vous pouvez gérer et maintenir les étiquettes pour les catégories MCS, ou les combinaisons de catégories MCS avec les niveaux MLS, sur votre système en éditant le fichier **setrans.conf**. Dans ce fichier, SELinux maintient une correspondance entre les niveaux de sensibilité et de catégorie internes et leurs étiquettes lisibles par l'homme.

NOTE

Les étiquettes de catégorie ne font que faciliter l'utilisation des catégories par les utilisateurs. Le système MCS fonctionne de la même manière, que vous définissiez des étiquettes ou non.

Conditions préalables

- Le mode SELinux est défini sur **enforcing**.
- La politique SELinux est définie sur **targeted** ou **mls**.
- Les paquets **policycoreutils-python-utils** et **mcstrans** sont installés.

Procédure

1. Modifiez les catégories existantes ou créez-en de nouvelles en éditant le fichier **/etc/selinux/*<selinuxpolicy>*/setrans.conf** dans un éditeur de texte. Remplacez *<selinuxpolicy>* par **targeted** ou **mls** en fonction de la politique SELinux que vous utilisez. Par exemple :

```
# vi /etc/selinux/targeted/setrans.conf
```

2. Dans le fichier **setrans.conf** de votre police, définissez les combinaisons de catégories requises par votre scénario à l'aide de la syntaxe suivante **s_<security level>_:c_<category number>_=*<category.name>***par exemple :

```
s0:c0=Marketing
s0:c1=Finance
s0:c2=Payroll
s0:c3=Personnel
```

   - Vous pouvez utiliser les numéros de catégorie de **c0** à **c1023**.
   - Dans la politique **targeted**, utilisez le niveau de sécurité **s0**.
   - Dans la politique **mls**, vous pouvez étiqueter chaque combinaison de niveaux de sensibilité et de catégories.

3. Facultatif : dans le fichier **setrans.conf**, vous pouvez également étiqueter les niveaux de sensibilité MLS.
4. Save and exit the file.
5. Pour que les modifications soient effectives, redémarrez le service de traduction MCS :

```
# systemctl restart mcstrans
```

Vérification

- Affiche les catégories actuelles :

```
# chcat -L
```

L'exemple ci-dessus produit le résultat suivant :

```
s0:c0                   Marketing
s0:c1                   Finance
s0:c2                   Payroll
s0:c3                   Personnel
s0
s0-s0:c0.c1023              SystemLow-SystemHigh
s0:c0.c1023                 SystemHigh
```

Ressources supplémentaires

- La page de manuel **setrans.conf(5)**.

7.4. ATTRIBUTION DE CATÉGORIES AUX UTILISATEURS DANS MCS

Vous pouvez définir les autorisations des utilisateurs en attribuant des catégories aux utilisateurs de Linux. Un utilisateur auquel des catégories ont été attribuées peut accéder et modifier des fichiers qui relèvent d'un sous-ensemble de catégories de l'utilisateur. Les utilisateurs peuvent également affecter des fichiers qu'ils possèdent aux catégories qui leur ont été attribuées.

Un utilisateur Linux ne peut pas être affecté à une catégorie qui se situe en dehors de la plage de sécurité définie pour l'utilisateur SELinux concerné.

NOTE

L'accès aux catégories est attribué lors de la connexion. Par conséquent, les utilisateurs n'ont pas accès aux catégories nouvellement attribuées tant qu'ils ne se sont pas reconnectés. De même, si vous révoquez l'accès d'un utilisateur à une catégorie, cette révocation n'est effective qu'après une nouvelle connexion de l'utilisateur.

Conditions préalables

- Le mode SELinux est défini sur **enforcing**.
- La politique SELinux est définie sur **targeted** ou **mls**.
- Le paquet **policycoreutils-python-utils** est installé.
- Les utilisateurs de Linux sont affectés à des utilisateurs confinés SELinux :
  - Les utilisateurs non privilégiés sont affectés à **user_u**.
  - Les utilisateurs privilégiés sont affectés à **staff_u**.

Procédure

1. Définir la plage de sécurité pour l'utilisateur SELinux.

```
# semanage user -m -rs0:c0,c1-s0:c0.c9 <user_u>
```

Utilisez les numéros de catégorie **c0** à **c1023** ou les étiquettes de catégorie définies dans le fichier **setrans.conf**. Pour plus d'informations, voir Définir des étiquettes de catégorie dans MCS.

2. Attribuer des catégories MCS à un utilisateur Linux. Vous ne pouvez spécifier qu'une plage à l'intérieur de la plage définie pour l'utilisateur SELinux concerné :

```
# semanage login -m -rs0:c1 <Linux.user1>
```

NOTE

Vous pouvez ajouter ou supprimer des catégories d'utilisateurs Linux à l'aide de la commande **chcat**. L'exemple suivant ajoute ***<category1>*** et supprime ***<category2>*** de ***<Linux.user1>*** et ***<Linux.user2>***:

```
# chcat -l -- <category1>,-<category2> <Linux.user1>,<Linux.user2>
```

Notez que vous devez spécifier **--** sur la ligne de commande avant d'utiliser la syntaxe **-*<category>*** avant d'utiliser la syntaxe. Sinon, la commande **chcat** interprète mal la suppression de la catégorie comme une option de commande.

Vérification

- Dressez la liste des catégories attribuées aux utilisateurs de Linux :

```
# chcat -L -l <Linux.user1>,<Linux.user2>
<Linux.user1>: <category1>,<category2>
<Linux.user2>: <category1>,<category2>
```

Ressources supplémentaires

- La page de manuel **chcat(8)**.

7.5. ATTRIBUTION DE CATÉGORIES AUX FICHIERS DANS MCS

Vous devez disposer de privilèges administratifs pour attribuer des catégories aux utilisateurs. Les utilisateurs peuvent ensuite attribuer des catégories aux fichiers. Pour modifier les catégories d'un fichier, les utilisateurs doivent avoir des droits d'accès à ce fichier. Les utilisateurs ne peuvent affecter un fichier qu'à une catégorie qui leur est attribuée.

NOTE

Le système combine les règles d'accès par catégorie avec les autorisations classiques d'accès aux fichiers. Par exemple, si un utilisateur de la catégorie **bigfoot** utilise le contrôle d'accès discrétionnaire (DAC) pour bloquer l'accès à un fichier par d'autres utilisateurs, les autres utilisateurs de **bigfoot** ne peuvent pas accéder à ce fichier. Un utilisateur assigné à toutes les catégories disponibles peut néanmoins ne pas être en mesure d'accéder à l'ensemble du système de fichiers.

Conditions préalables

- Le mode SELinux est défini sur **enforcing**.
- La politique SELinux est définie sur **targeted** ou **mls**.
- Le paquet **policycoreutils-python-utils** est installé.
- Accès et permissions à un utilisateur Linux qui est :
  - Attribué à un utilisateur SELinux.
  - Affecté à la catégorie à laquelle vous souhaitez affecter le fichier. Pour plus d'informations, voir Affectation de catégories aux utilisateurs dans MCS .
- Accès et autorisations au fichier que vous souhaitez ajouter à la catégorie.
- À des fins de vérification : Accès et autorisations à un utilisateur Linux non affecté à cette catégorie

Procédure

- Ajouter des catégories à un fichier :

```
$ chcat -- <category1>, <category2> <path/to/file1>
```

Utilisez les numéros de catégorie **c0** à **c1023** ou les étiquettes de catégorie définies dans le fichier **setrans.conf**. Pour plus d'informations, voir Définir des étiquettes de catégorie dans MCS.

Vous pouvez supprimer des catégories d'un fichier en utilisant la même syntaxe :

```
$ chcat -- -<category1>,-<category2> <path/to/file1>
```

NOTE

Lors de la suppression d'une catégorie, vous devez spécifier **--** sur la ligne de commande avant d'utiliser la syntaxe ***-<category>*** avant d'utiliser la syntaxe. Sinon, la commande **chcat** pourrait interpréter à tort la suppression de la catégorie comme une option de commande.

Vérification

1. Affichez le contexte de sécurité du fichier pour vérifier qu'il possède les catégories correctes :

```
$ ls -lZ <path/to/file>
-rw-r--r--  <LinuxUser1> <Group1> root:object_r:user_home_t:_<sensitivity>_:_<category>_
<path/to/file>
```

Le contexte de sécurité spécifique du fichier peut être différent.

2. Facultatif : Tenter d'accéder au fichier en étant connecté en tant qu'utilisateur Linux n'appartenant pas à la même catégorie que le fichier :

```
$ cat <path/to/file>
cat: <path/to/file>: Permission Denied
```

Ressources supplémentaires

- La page de manuel **semanage(8)**.
- La page de manuel **chcat(8)**.

CHAPITRE 8. RÉDACTION D'UNE POLITIQUE SELINUX PERSONNALISÉE

Cette section vous explique comment rédiger et utiliser une stratégie personnalisée qui vous permet d'exécuter vos applications dans les limites de SELinux.

8.1. POLITIQUES SELINUX PERSONNALISÉES ET OUTILS CONNEXES

Une politique de sécurité SELinux est un ensemble de règles SELinux. Une politique est un élément central de SELinux et est chargée dans le noyau par les outils SELinux de l'espace utilisateur. Le noyau impose l'utilisation d'une politique SELinux pour évaluer les demandes d'accès au système. Par défaut, SELinux refuse toutes les demandes à l'exception de celles qui correspondent aux règles spécifiées dans la politique chargée.

Chaque règle SELinux décrit une interaction entre un processus et une ressource du système :

```
ALLOW apache_process apache_log:FILE READ;
```

Vous pouvez lire cet exemple de règle comme suit : *The **Apache** process can **read** its **logging file***. Dans cette règle, **apache_process** et **apache_log** sont **labels**. Une politique de sécurité SELinux attribue des étiquettes aux processus et définit les relations avec les ressources du système. De cette manière, une politique fait correspondre les entités du système d'exploitation à la couche SELinux.

Les étiquettes SELinux sont stockées en tant qu'attributs étendus des systèmes de fichiers, tels que **ext2**. Vous pouvez les lister à l'aide de l'utilitaire **getfattr** ou de la commande **ls -Z**, par exemple :

```
$ ls -Z /etc/passwd
system_u:object_r:passwd_file_t:s0 /etc/passwd
```

Où **system_u** est un utilisateur SELinux, **object_r** est un exemple de rôle SELinux et **passwd_file_t** est un domaine SELinux.

La politique SELinux par défaut fournie par les paquetages **selinux-policy** contient des règles pour les applications et les démons qui font partie de Red Hat Enterprise Linux 9 et qui sont fournis par des paquetages dans ses dépôts. Les applications qui ne sont pas décrites dans une règle de cette politique de distribution ne sont pas limitées par SELinux. Pour changer cela, vous devez modifier la politique à l'aide d'un module de politique, qui contient des définitions et des règles supplémentaires.

Dans Red Hat Enterprise Linux 9, vous pouvez interroger la politique SELinux installée et générer de nouveaux modules de politique à l'aide de l'outil **sepolicy**. Les scripts que **sepolicy** génère avec les modules de stratégie contiennent toujours une commande utilisant l'utilitaire **restorecon**. Cet utilitaire est un outil de base permettant de résoudre les problèmes d'étiquetage dans une partie sélectionnée d'un système de fichiers.

Ressources supplémentaires

- **sepolicy(8)** et **getfattr(1)** pages de manuel

8.2. CRÉATION ET APPLICATION D'UNE POLITIQUE SELINUX POUR UNE APPLICATION PERSONNALISÉE

Cet exemple de procédure fournit les étapes pour confiner un simple démon par SELinux. Remplacez le démon par votre application personnalisée et modifiez la règle de l'exemple en fonction des exigences de cette application et de votre politique de sécurité.

Conditions préalables

- Le paquetage **policycoreutils-devel** et ses dépendances sont installés sur votre système.

Procédure

1. Pour cet exemple de procédure, préparez un simple démon qui ouvre le fichier **/var/log/messages** en écriture :

   a. Créez un nouveau fichier et ouvrez-le dans l'éditeur de texte de votre choix :

   ```
   $ vi mydaemon.c
   ```

   b. Insérer le code suivant :

   ```
   #include <unistd.h>
   #include <stdio.h>

   FILE *f;

   int main(void)
   {
   while(1) {
   f = fopen("/var/log/messages","w");
           sleep(5);
           fclose(f);
       }
   }
   ```

   c. Compiler le fichier :

   ```
   $ gcc -o mydaemon mydaemon.c
   ```

   d. Créez un fichier d'unité **systemd** pour votre démon :

   ```
   $ vi mydaemon.service
   [Unit]
   Description=Simple testing daemon

   [Service]
   Type=simple
   ExecStart=/usr/local/bin/mydaemon

   [Install]
   WantedBy=multi-user.target
   ```

   e. Installer et démarrer le démon :

   ```
   # cp mydaemon /usr/local/bin/
   # cp mydaemon.service /usr/lib/systemd/system
   ```

```
# systemctl start mydaemon
# systemctl status mydaemon
● mydaemon.service - Simple testing daemon
   Loaded: loaded (/usr/lib/systemd/system/mydaemon.service; disabled; vendor preset:
disabled)
   Active: active (running) since Sat 2020-05-23 16:56:01 CEST; 19s ago
 Main PID: 4117 (mydaemon)
    Tasks: 1
   Memory: 148.0K
   CGroup: /system.slice/mydaemon.service
           └─4117 /usr/local/bin/mydaemon

May 23 16:56:01 localhost.localdomain systemd[1]: Started Simple testing daemon.
```

f. Vérifiez que le nouveau démon n'est pas limité par SELinux :

```
$ ps -efZ | grep mydaemon
system_u:system_r:unconfined_service_t:s0 root 4117    1  0 16:56 ?        00:00:00
/usr/local/bin/mydaemon
```

2. Générer une politique personnalisée pour le démon :

```
$ sepolicy generate --init /usr/local/bin/mydaemon
Created the following files:
/home/example.user/mysepol/mydaemon.te # Type Enforcement file
/home/example.user/mysepol/mydaemon.if # Interface file
/home/example.user/mysepol/mydaemon.fc # File Contexts file
/home/example.user/mysepol/mydaemon_selinux.spec # Spec file
/home/example.user/mysepol/mydaemon.sh # Setup Script
```

3. Reconstruire la politique du système avec le nouveau module de politique en utilisant le script d'installation créé par la commande précédente :

```
# ./mydaemon.sh
Building and Loading Policy
+ make -f /usr/share/selinux/devel/Makefile mydaemon.pp
Compiling targeted mydaemon module
Creating targeted mydaemon.pp policy package
rm tmp/mydaemon.mod.fc tmp/mydaemon.mod
+ /usr/sbin/semodule -i mydaemon.pp
...
```

Notez que le script d'installation renomme la partie correspondante du système de fichiers à l'aide de la commande **restorecon**:

```
restorecon -v /usr/local/bin/mydaemon /usr/lib/systemd/system
```

4. Redémarrez le démon et vérifiez qu'il fonctionne désormais dans le respect de SELinux :

```
# systemctl restart mydaemon
$ ps -efZ | grep mydaemon
system_u:system_r:mydaemon_t:s0 root       8150       1  0 17:18 ?        00:00:00
/usr/local/bin/mydaemon
```

5. Étant donné que le démon est désormais confiné par SELinux, SELinux l'empêche également d'accéder à **/var/log/messages**. Affichez le message de refus correspondant :

```
# ausearch -m AVC -ts recent
...
type=AVC msg=audit(1590247112.719:5935): avc:  denied  { open } for  pid=8150
comm="mydaemon" path="/var/log/messages" dev="dm-0" ino=2430831
scontext=system_u:system_r:mydaemon_t:s0 tcontext=unconfined_u:object_r:var_log_t:s0
tclass=file permissive=1
...
```

6. Vous pouvez également obtenir des informations supplémentaires en utilisant l'outil **sealert**:

```
$ sealert -l "*"
SELinux is preventing mydaemon from open access on the file /var/log/messages.

 Plugin catchall (100. confidence) suggests *

If you believe that mydaemon should be allowed open access on the messages file by
default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'mydaemon' --raw | audit2allow -M my-mydaemon
# semodule -X 300 -i my-mydaemon.pp

Additional Information:
Source Context                system_u:system_r:mydaemon_t:s0
Target Context                unconfined_u:object_r:var_log_t:s0
Target Objects                /var/log/messages [ file ]
Source                        mydaemon

...
```

7. **Utilisez l'outil audit2allow pour suggérer des changements :**

```
$ ausearch -m AVC -ts recent | audit2allow -R

require {
 type mydaemon_t;
}

#============= mydaemon_t ==============
logging_write_generic_logs(mydaemon_t)
```

8. **Les règles suggérées par audit2allow pouvant être incorrectes dans certains cas, n'utilisez qu'une partie de ses résultats pour trouver l'interface de politique correspondante :**

```
$ grep -r "logging_write_generic_logs" /usr/share/selinux/devel/include/ | grep .if
/usr/share/selinux/devel/include/system/logging.if:interface(`logging_write_generic_logs',`
```

9. **Vérifier la définition de l'interface :**

```
$ cat /usr/share/selinux/devel/include/system/logging.if
...
interface(`logging_write_generic_logs',`
        gen_require(`
                type var_log_t;
        ')

        files_search_var($1)
        allow $1 var_log_t:dir list_dir_perms;
        write_files_pattern($1, var_log_t, var_log_t)
')
...
```

10. **Dans ce cas, vous pouvez utiliser l'interface proposée. Ajoutez la règle correspondante à votre fichier d'application des types :**

```
$ echo "logging_write_generic_logs(mydaemon_t)" >> mydaemon.te
```

Vous pouvez également ajouter cette règle au lieu d'utiliser l'interface :

```
$ echo "allow mydaemon_t var_log_t:file { open write getattr };" >> mydaemon.te
```

11. **Réinstaller la politique :**

```
# ./mydaemon.sh
Building and Loading Policy
+ make -f /usr/share/selinux/devel/Makefile mydaemon.pp
Compiling targeted mydaemon module
Creating targeted mydaemon.pp policy package
rm tmp/mydaemon.mod.fc tmp/mydaemon.mod
+ /usr/sbin/semodule -i mydaemon.pp
...
```

Vérification

1. **Vérifiez que votre application fonctionne dans le cadre de SELinux, par exemple :**

```
$ ps -efZ | grep mydaemon
system_u:system_r:mydaemon_t:s0 root        8150       1  0 17:18 ?        00:00:00
/usr/local/bin/mydaemon
```

2. **Vérifiez que votre application personnalisée ne provoque pas de refus SELinux :**

```
# ausearch -m AVC -ts recent
<no matches>
```

Ressources supplémentaires

- **sepolgen(8)**, **ausearch(8)**, **audit2allow(1)**, **audit2why(1)**, **sealert(8)**, et **restorecon(8)** pages de manuel

8.3. CRÉATION D'UN MODULE LOCAL DE POLITIQUE SELINUX

L'ajout de modules de politique SELinux spécifiques à une politique SELinux active peut résoudre certains problèmes avec la politique SELinux. Vous pouvez utiliser cette procédure pour corriger un problème connu spécifique décrit dans les notes de mise à jour de RedHat, ou pour mettre en œuvre une solution Red Hatspécifique.

AVERTISSEMENT

Utilisez uniquement les règles fournies par Red Hat. Red Hat ne prend pas en charge la création de modules de politiques SELinux avec des règles personnalisées, car cela sort de la portée de la couverture de l'assistance à la production. Si vous n'êtes pas un expert, contactez votre représentant commercial Red Hat et demandez des services de conseil.

Conditions préalables

- Les paquets **setools-console** et **audit** pour vérification.

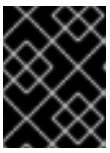
Procédure

1. Ouvrez un nouveau fichier **.cil** avec un éditeur de texte, par exemple :

```
# vim <local_module>.cil
```

Pour mieux organiser vos modules locaux, utilisez le préfixe **local_** dans les noms des modules de politique SELinux locaux.

2. Insérez les règles personnalisées à partir d'un problème connu ou d'une solution Red Hat.



IMPORTANT

N'écrivez pas vos propres règles. Utilisez uniquement les règles fournies dans un problème connu spécifique ou une solution Red Hat.

Par exemple, pour mettre en œuvre la solution SELinux denies cups-lpd read access to cups.sock in RHEL, insérez la règle suivante :

NOTE

L'exemple de solution a été corrigé de manière permanente pour RHEL dans RHBA-2021:4420. Par conséquent, les parties de cette procédure spécifiques à cette solution n'ont aucun effet sur les systèmes RHEL 8 et 9 mis à jour, et ne sont incluses qu'à titre d'exemples de syntaxe.

```
(allow cupsd_lpd_t cupsd_var_run_t (sock_file (read)))
```

Notez que vous pouvez utiliser l'une ou l'autre des deux syntaxes de règles SELinux, Common Intermediate Language (CIL) et m4. Par exemple, **(allow cupsd_lpd_t cupsd_var_run_t (sock_file (read)))** en CIL est équivalent à ce qui suit en m4 :

```
module local_cupslpd-read-cupssock 1.0;

require {
    type cupsd_var_run_t;
    type cupsd_lpd_t;
    class sock_file read;
}

#============= cupsd_lpd_t ==============
allow cupsd_lpd_t cupsd_var_run_t:sock_file read;
```

3. **Enregistrez et fermez le fichier.**

4. **Installer le module de politique :**

```
# semodule -i <local_module>.cil
```

NOTE

Pour supprimer un module de politique locale créé à l'aide de semodule -i, il convient de se référer au nom du module sans le suffixe .cil. Pour supprimer un module de politique locale, utilisez semodule -r *<local_module>*.

5. **Redémarrer tous les services liés aux règles :**

```
# systemctl restart <service-name>
```

Vérification

1. **Liste des modules locaux installés dans votre politique SELinux :**

```
# semodule -lfull | grep "local_"
400 local_module  cil
```

NOTE

Les modules locaux ayant la priorité 400, vous pouvez également les filtrer de la liste en utilisant cette valeur, par exemple en utilisant la commande semodule -lfull | grep -v ^100.

2. **Recherchez les règles d'autorisation pertinentes dans la politique SELinux :**

```
# sesearch -A --source=<SOURCENAME> --target=<TARGETNAME> --
class=<CLASSNAME> --perm=<P1>,<P2>
```

Où *<SOURCENAME>* est le type SELinux source,*<TARGETNAME>* est le type SELinux cible, *<CLASSNAME>* est le nom de la classe de sécurité ou de la classe d'objets, et *<P1>* et *<P2>* sont les autorisations spécifiques de la règle.

Par exemple, pour la solution RHEL, SELinux refuse à cups-lpd l'accès en lecture à cups.sock :

```
# sesearch -A --source=cupsd_lpd_t --target=cupsd_var_run_t --class=sock_file --
perm=read
allow cupsd_lpd_t cupsd_var_run_t:sock_file { append getattr open read write };
```

La dernière ligne doit maintenant inclure l'opération **read**.

3. **Vérifiez que le service concerné fonctionne dans le cadre de SELinux :**

   a. **Identifier le processus lié au service concerné :**

   ```
   $ systemctl status <service-name>
   ```

   b. **Vérifiez le contexte SELinux du processus répertorié dans la sortie de la commande précédente :**

   ```
   $ ps -efZ | grep <process-name>
   ```

4. **Vérifier que le service ne provoque pas de déni SELinux :**

```
# ausearch -m AVC -ts recent
<no matches>
```

Ressources supplémentaires

- Chapitre 5, *Résolution des problèmes liés à SELinux*

8.4. RESSOURCES SUPPLÉMENTAIRES

- Atelier sur la politique SELinux

CHAPITRE 9. CRÉATION DE POLITIQUES SELINUX POUR LES CONTENEURS

Red Hat Enterprise Linux 9 fournit un outil permettant de générer des politiques SELinux pour les conteneurs à l'aide du paquetage **udica**. Avec **udica**, vous pouvez créer une politique de sécurité personnalisée pour mieux contrôler la façon dont un conteneur accède aux ressources du système hôte, telles que le stockage, les périphériques et le réseau. Cela vous permet de renforcer vos déploiements de conteneurs contre les violations de sécurité et simplifie également l'obtention et le maintien de la conformité réglementaire.

9.1. INTRODUCTION AU GÉNÉRATEUR DE POLITIQUES SELINUX UDICA

Pour simplifier la création de nouvelles politiques SELinux pour les conteneurs personnalisés, RHEL 9 fournit l'utilitaire **udica**. Vous pouvez utiliser cet outil pour créer une stratégie basée sur une inspection du fichier JavaScript Object Notation (JSON) du conteneur, qui contient les capacités Linux, les points de montage et les définitions de ports. L'outil combine ensuite les règles générées à l'aide des résultats de l'inspection avec les règles héritées d'un bloc SELinux Common Intermediate Language (CIL) spécifié.

Le processus de génération d'une politique SELinux pour un conteneur à l'aide de **udica** comporte trois parties principales :

1. Analyse du fichier de spécification du conteneur au format JSON
2. Trouver des règles d'autorisation appropriées sur la base des résultats de la première partie
3. Génération de la politique SELinux finale

Au cours de la phase d'analyse, **udica** recherche les capacités Linux, les ports réseau et les points de montage.

Sur la base des résultats, **udica** détecte les capacités Linux requises par le conteneur et crée une règle SELinux autorisant toutes ces capacités. Si le conteneur se lie à un port spécifique, **udica** utilise les bibliothèques SELinux de l'espace utilisateur pour obtenir l'étiquette SELinux correcte d'un port utilisé par le conteneur inspecté.

Ensuite, **udica** détecte les répertoires montés dans l'espace de noms du système de fichiers du conteneur à partir de l'hôte.

La fonction d'héritage de blocs de la CIL permet à **udica** de créer des modèles de SELinux*allow rules* axés sur une action spécifique, par exemple :

- *allow accessing home directories*
- *allow accessing log files*
- *allow accessing communication with Xserver.*

Ces modèles sont appelés blocs et la politique SELinux finale est créée en fusionnant les blocs.

Ressources supplémentaires

- Générer des politiques SELinux pour les conteneurs avec udicaRed Hat Blog article

9.2. CRÉATION ET UTILISATION D'UNE POLITIQUE SELINUX POUR UN CONTENEUR PERSONNALISÉ

Pour générer une politique de sécurité SELinux pour un conteneur personnalisé, suivez les étapes de cette procédure.

Conditions préalables

- L'outil **podman** de gestion des conteneurs est installé. Si ce n'est pas le cas, utilisez la commande **dnf install podman**.
- Un conteneur Linux personnalisé - *ubi8* dans cet exemple.

Procédure

1. Installez le paquetage **udica**:

```
# dnf install -y udica
```

Vous pouvez également installer le module **container-tools**, qui fournit un ensemble de logiciels de conteneurs, dont **udica**:

```
# dnf module install -y container-tools
```

2. Démarrez le conteneur *ubi8* qui monte le répertoire/**home** avec des autorisations de lecture seule et le répertoire /**var**/**spool** avec des autorisations de lecture et d'écriture. Le conteneur expose le port *21*.

```
# podman run --env container=podman -v /home:/home:ro -v /var/spool:/var/spool:rw -p
21:21 -it ubi8 bash
```

Notez que le conteneur fonctionne maintenant avec le type SELinux **container_t**. Ce type est un domaine générique pour tous les conteneurs dans la politique SELinux et il peut être trop strict ou trop lâche pour votre scénario.

3. Ouvrez un nouveau terminal et entrez la commande **podman ps** pour obtenir l'ID du conteneur :

```
# podman ps
CONTAINER ID  IMAGE                                 COMMAND  CREATED         STATUS
PORTS  NAMES
37a3635afb8f  registry.access.redhat.com/ubi8:latest  bash     15 minutes ago  Up 15
minutes ago          heuristic_lewin
```

4. Créez un fichier JSON de conteneur et utilisez **udica** pour créer un module de politique basé sur les informations contenues dans le fichier JSON :

```
# podman inspect 37a3635afb8f > container.json
# udica -j container.json my_container
Policy my_container with container id 37a3635afb8f created!
[...]
```

Alternativement :

-

```
# podman inspect 37a3635afb8f | udica my_container
Policy my_container with container id 37a3635afb8f created!

Please load these modules using:
# semodule -i my_container.cil
/usr/share/udica/templates/{base_container.cil,net_container.cil,home_container.cil}

Restart the container with: "--security-opt label=type:my_container.process" parameter
```

5. **Comme le suggère la sortie de udica à l'étape précédente, chargez le module de politique :**

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

6. **Arrêtez le conteneur et redémarrez-le avec l'option --security-opt label=type:my_container.process:**

```
# podman stop 37a3635afb8f
# podman run --security-opt label=type:my_container.process -v /home:/home:ro -v
/var/spool:/var/spool:rw -p 21:21 -it ubi8 bash
```

Vérification

1. **Vérifiez que le conteneur fonctionne avec le type my_container.process:**

```
# ps -efZ | grep my_container.process
unconfined_u:system_r:container_runtime_t:s0-s0:c0.c1023 root 2275 434  1 13:49 pts/1
00:00:00 podman run --security-opt label=type:my_container.process -v /home:/home:ro -v
/var/spool:/var/spool:rw -p 21:21 -it ubi8 bash
system_u:system_r:my_container.process:s0:c270,c963 root 2317 2305  0 13:49 pts/0
00:00:00 bash
```

2. **Vérifiez que SELinux autorise désormais l'accès aux points de montage /home et /var/spool:**

```
[root@37a3635afb8f /]# cd /home
[root@37a3635afb8f home]# ls
username
[root@37a3635afb8f ~]# cd /var/spool/
[root@37a3635afb8f spool]# touch test
[root@37a3635afb8f spool]#
```

3. **Vérifiez que SELinux n'autorise que la connexion au port 21 :**

```
[root@37a3635afb8f /]# dnf install nmap-ncat
[root@37a3635afb8f /]# nc -lvp 21
...
Ncat: Listening on :::21
Ncat: Listening on 0.0.0.0:21
^C
[root@37a3635afb8f /]# nc -lvp 80
...
Ncat: bind to :::80: Permission denied. QUITTING.
```

Ressources supplémentaires

- **udica(8)** et **podman(1)** pages de manuel
- Construire, exécuter et gérer des conteneurs

9.3. RESSOURCES SUPPLÉMENTAIRES

- udica - Générer des politiques SELinux pour les conteneurs

CHAPITRE 10. DÉPLOYER LA MÊME CONFIGURATION SELINUX SUR PLUSIEURS SYSTÈMES

Cette section présente deux méthodes recommandées pour déployer votre configuration SELinux vérifiée sur plusieurs systèmes :

- Utilisation des rôles système RHEL et d'Ansible
- Utilisation des commandes d'exportation et d'importation de **semanage** dans vos scripts

10.1. INTRODUCTION AU RÔLE DU SYSTÈME SELINUX

RHEL System Roles est une collection de rôles et de modules Ansible qui fournissent une interface de configuration cohérente pour gérer à distance plusieurs systèmes RHEL. Le rôle de système **selinux** permet les actions suivantes :

- Nettoyage des modifications de politiques locales liées aux booléens SELinux, aux contextes de fichiers, aux ports et aux connexions.
- Définition des booléens de la politique SELinux, des contextes de fichiers, des ports et des connexions.
- Restauration des contextes de fichiers sur les fichiers ou répertoires spécifiés.
- Gestion des modules SELinux.

Le tableau suivant donne un aperçu des variables d'entrée disponibles dans le rôle de système **selinux**.

Tableau 10.1. **selinux** Variables de rôle du système

Variable de rôle	Description	Alternative à l'interface de programmation
politique_sélinux	Choix d'une politique de protection des processus ciblés ou d'une protection de sécurité multi-niveaux.	**SELINUXTYPE** en **/etc/selinux/config**
état_sélinux	Change les modes SELinux.	**setenforce** et **SELINUX** dans **/etc/selinux/config**.
booléens_sélinux	Active et désactive les booléens SELinux.	**setsebool**
selinux_fcontexts	Ajoute ou supprime une correspondance de contexte de fichier SELinux.	**semanage fcontext**
selinux_restore_dirs	Rétablit les étiquettes SELinux dans l'arborescence du système de fichiers.	**restorecon -R**

Variable de rôle	Description	Alternative à l'interface de programmation
ports_sélinux	Définit les étiquettes SELinux sur les ports.	**semanage port**
selinux_logins	Définit les utilisateurs en fonction du mappage des utilisateurs SELinux.	**semanage login**
selinux_modules	Installe, active, désactive ou supprime les modules SELinux.	**semodule**

L'exemple d'exécution **/usr/share/doc/rhel-system-roles/selinux/example-selinux-playbook.yml** installé par le paquetage **rhel-system-roles** montre comment définir la stratégie ciblée en mode d'exécution. Il applique également plusieurs modifications de la stratégie locale et restaure les contextes de fichiers dans le répertoire **/tmp/test_dir/**.

Pour une référence détaillée sur les variables de rôle **selinux**, installez le paquetage **rhel-system-roles** et consultez les fichiers **README.md** ou **README.html** dans le répertoire **/usr/share/doc/rhel-system-roles/selinux/**.

Ressources supplémentaires

- Introduction aux rôles du système RHEL

10.2. UTILISATION DU RÔLE DE SYSTÈME SELINUX POUR APPLIQUER LES PARAMÈTRES SELINUX À PLUSIEURS SYSTÈMES

Suivez les étapes pour préparer et appliquer un playbook Ansible avec vos paramètres SELinux vérifiés.

Conditions préalables

- Accès et autorisations à un ou plusieurs *managed nodes*, qui sont des systèmes que vous souhaitez configurer avec le rôle de système **selinux**.
- Accès et permissions à un *control node*, qui est un système à partir duquel Red Hat Ansible Core configure d'autres systèmes.
  Sur le nœud de contrôle :
  - Les paquets **ansible-core** et **rhel-system-roles** sont installés.
  - Un fichier d'inventaire qui répertorie les nœuds gérés.

IMPORTANT

RHEL 8.0-8.5 donne accès à un dépôt Ansible distinct qui contient Ansible Engine 2.9 pour l'automatisation basée sur Ansible. Ansible Engine contient des utilitaires de ligne de commande tels que **ansible**, **ansible-playbook**, des connecteurs tels que **docker** et **podman**, ainsi que de nombreux plugins et modules. Pour plus d'informations sur la manière d'obtenir et d'installer Ansible Engine, consultez l'article de la base de connaissances Comment télécharger et installer Red Hat Ansible Engine.

RHEL 8.6 et 9.0 ont introduit Ansible Core (fourni en tant que paquetage **ansible-core**), qui contient les utilitaires de ligne de commande Ansible, les commandes et un petit ensemble de plugins Ansible intégrés. RHEL fournit ce paquetage par l'intermédiaire du dépôt AppStream, et sa prise en charge est limitée. Pour plus d'informations, consultez l'article de la base de connaissances intitulé Scope of support for the Ansible Core package included in the RHEL 9 and RHEL 8.6 and later AppStream repositories (Portée de la prise en charge du package Ansible Core inclus dans les dépôts AppStream RHEL 9 et RHEL 8.6 et versions ultérieures).

- Un fichier d'inventaire qui répertorie les nœuds gérés.

Procédure

1. Préparez votre playbook. Vous pouvez partir de zéro ou modifier le playbook d'exemple installé avec le paquetage **rhel-system-roles**:

```
# cp /usr/share/doc/rhel-system-roles/selinux/example-selinux-playbook.yml my-selinux-playbook.yml
# vi my-selinux-playbook.yml
```

2. Modifiez le contenu du playbook pour l'adapter à votre scénario. Par exemple, la partie suivante garantit que le système installe et active le module SELinux **selinux-local-1.pp**:

```
selinux_modules:
- { path: "selinux-local-1.pp", priority: "400" }
```

3. Enregistrez les modifications et quittez l'éditeur de texte.

4. Exécutez votre manuel de jeu sur les systèmes *host1*, *host2* et *host3*:

```
# ansible-playbook -i host1,host2,host3 my-selinux-playbook.yml
```

Ressources supplémentaires

- Pour plus d'informations, installez le paquetage **rhel-system-roles** et consultez les répertoires **/usr/share/doc/rhel-system-roles/selinux/** et **/usr/share/ansible/roles/rhel-system-roles.selinux/**.

10.3. TRANSFÉRER LES PARAMÈTRES SELINUX VERS UN AUTRE SYSTÈME AVEC SEMANAGE

Suivez les étapes suivantes pour transférer vos paramètres SELinux personnalisés et vérifiés entre les systèmes basés sur RHEL 9.

Conditions préalables

- Le paquetage **policycoreutils-python-utils** est installé sur votre système.

Procédure

1. Exportez vos paramètres SELinux vérifiés :

```
# semanage export -f ./my-selinux-settings.mod
```

2. Copiez le fichier contenant les paramètres sur le nouveau système :

```
# scp ./my-selinux-settings.mod new-system-hostname:
```

3. Se connecter au nouveau système :

```
$ ssh root@new-system-hostname
```

4. Importer les paramètres sur le nouveau système :

```
new-system-hostname# semanage import -f ./my-selinux-settings.mod
```

Ressources supplémentaires

- **semanage-export(8)** et **semanage-import(8)** pages de manuel