Version : **2024.01**

Dernière mise-à-jour : 2024/10/22 10:15

RH13404 - Gestion de la Sécurité

Contenu du Module

LAB #1 - Les Droits Unix Avancés

1.1 - Les ACL

Au delà des droits étendus d'Unix, Linux utilise un système d'ACL pour permettre une meilleure gestion des droits sur des fichiers.

Pour connaître les ACL positionnés sur un fichier, il convient d'utiliser la commande **getfacl** :

```
[root@redhat9 ~]# touch tux.jpg

[root@redhat9 ~]# getfacl tux.jpg
# file: tux.jpg
# owner: root
# group: root
user::rw-
group::r--
```

```
other::r--
```

Pour positionner des ACL sur un fichier, il convient d'utiliser la commande **setfacl** :

```
[root@redhat9 ~]# setfacl --set u::rwx,g::rx,o::-,u:trainee:rw tux.jpg

[root@redhat9 ~]# getfacl tux.jpg
# file: tux.jpg
# owner: root
# group: root
user::rwx
user:trainee:rw-
group::r-x
mask::rwx
other::---
```

Important - Veuillez noter l'apparition de la ligne **mask**. Le mask indique les permissions maximales qui peuvent être acccordées à un utilisateur ou un groupe tiers.

Regardez maintenant l'effet des ACL sur un répertoire. Créez le répertoire /home/trainee/rep1 :

```
[root@redhat9 ~]# mkdir rep1
```

Positionnez des ACL le répertoire avec la commande **setfacl** :

```
[root@redhat9 ~]# setfacl --set d:u::r,d:g::-,d:o::- rep1
```

Notez l'utilisation de la lettre **d** pour indiquer une permission *par défaut*.

Créez maintenant un fichier appelé fichier1 dans /root/rep1 :

```
[root@redhat9 ~]# touch rep1/fichier1
```

Utilisez la commande **getfacl** pour visualiser le résultat :

```
[root@redhat9 ~]# getfacl rep1
# file: rep1
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::r--
default:group::---
default:other::---

[root@redhat9 ~]# getfacl rep1/fichier1
# file: rep1/fichier1
# owner: root
# group: root
user::r--
group::---
other::---
```

Notez que le fichier créé possède les ACL positionnés sur le répertoire rep1.

Dernièrement, les systèmes de sauvegarde classiques sous Linux ne peuvent pas sauvegarder les ACL, sauf l'outil **star**. Si vous n'utilisez pas **star**, il convient donc de sauvegarder les ACL dans un fichier grâce à la commande suivante :

```
[root@redhat9 ~]# cd rep1

[root@redhat9 rep1]# getfacl -R --skip-base . > backup.acl

[root@redhat9 rep1]# cat backup.acl
# file: .
```

```
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::r--
default:group::---
default:other::---
```

La restauration des ACL se fait avec la commande **setfacl** :

```
# setfacl --restore=backup.acl [Entrée]
```

Options des Commandes

Les options de la commande **getfacl** sont :

```
[root@redhat9 rep1]# getfacl --help
getfacl 2.3.1 -- get file access control lists
Usage: getfacl [-aceEsRLPtpndvh] file ...
  -a, --access            display the file access control list only
  -d, --default           display the default access control list only
  -c, --omit-header       do not display the comment header
  -e, --all-effective     print all effective rights
  -E, --no-effective      print no effective rights
  -s, --skip-base         skip files that only have the base entries
  -R, --recursive         recurse into subdirectories
  -L, --logical           logical walk, follow symbolic links
  -P, --physical          physical walk, do not follow symbolic links
  -t, --tabular           use tabular output format
  -n, --numeric           print numeric user/group identifiers
      --one-file-system   skip files on different filesystems
```

```
  -p, --absolute-names    don't strip leading '/' in pathnames
  -v, --version           print version and exit
  -h, --help              this help text
```

Les options de la commande **setfacl** sont :

```
[root@redhat9 rep1]# setfacl --help
setfacl 2.3.1 -- set file access control lists
Usage: setfacl [-bkndRLP] { -m|-M|-x|-X ... } file ...
  -m, --modify=acl        modify the current ACL(s) of file(s)
  -M, --modify-file=file  read ACL entries to modify from file
  -x, --remove=acl        remove entries from the ACL(s) of file(s)
  -X, --remove-file=file  read ACL entries to remove from file
  -b, --remove-all        remove all extended ACL entries
  -k, --remove-default    remove the default ACL
      --set=acl           set the ACL of file(s), replacing the current ACL
      --set-file=file     read ACL entries to set from file
      --mask              do recalculate the effective rights mask
  -n, --no-mask           don't recalculate the effective rights mask
  -d, --default           operations apply to the default ACL
  -R, --recursive         recurse into subdirectories
  -L, --logical           logical walk, follow symbolic links
  -P, --physical          physical walk, do not follow symbolic links
      --restore=file      restore ACLs (inverse of `getfacl -R')
      --test              test mode (ACLs are not modified)
  -v, --version           print version and exit
  -h, --help              this help text
```

1.2 - Les Attributs Etendus

Les attributs s'ajoutent aux caractéristiques classiques d'un fichier dans un système de fichiers Ext2/Ext3/Ext4, JFS, ReiserFS, XFS et Btrfs.

Les principaux attributs sont :

Attribut	Description
a	Fichier journal - uniquement l'ajout de données au fichier est permis. Le fichier ne peut pas détruit
i	Le fichier ne peut ni être modifié, ni être détruit, ni être déplacé. Le placement d'un lien sur le fichier n'est pas permis
s	Le fichier sera physiquement détruit lors de sa suppression
D	Répertoire synchrone
S	Fichier synchrone
A	La date et l'heure de dernier accès ne seront pas mises à jour

Important - Un fichier synchrone et un répertoire synchrone impliquent que les modifications seront immédiatement inscrites sur disque.

Les commandes associées avec les attributs sont :

Commande	description
chattr	Modifie les attributs
lsattr	Visualise les attributs

Pour mieux comprendre, créez le répertoire **/root/attributs/rep** :

```
[root@redhat9 rep1]# cd ..
[root@redhat9 ~]# mkdir -p attributs/rep
```

Créez ensuite les fichier **fichier** et **rep/fichier1** :

```
[root@redhat9 ~]# touch attributs/fichier
[root@redhat9 ~]# touch attributs/rep/fichier1
```

Modifiez les attributs d'une manière récursive sur le répertoire **attributs** :

```
[root@redhat9 ~]# chattr +i -R attributs/
```

Visualisez les attributs de l'arborescence **attributs** :

```
[root@redhat9 ~]# lsattr -R attributs
----i----------------- attributs/rep

attributs/rep:
----i----------------- attributs/rep/fichier1

----i----------------- attributs/fichier
```

> **Important** - Notez que l'attribut **e** sous Ext4 indique l'utilisation des **Extents**. Cet attribut ne peut pas être enlever avec la commande **chattr**. Les Extents seront couverts dans le cours **Gestion des Disques, des Systèmes de Fichiers et le Swap**.

Essayez maintenant de déplacer le fichier **fichier**. Vous obtiendrez un résultat similaire à celui-ci :

```
[root@redhat9 ~]# cd attributs; mv /root/attributs/fichier /root/attributs/rep/fichier
mv: cannot move '/root/attributs/fichier' to '/root/attributs/rep/fichier': Operation not permitted
```

LAB #2 - Mise en place de SELinux pour sécuriser le serveur

2.1 - Introducton

L'approche SELinux (*Security Enhanced Linux*) à la sécurité est une approche de type **TE**. Elle essaie aussi d'intégrer les notions des approches de type **RBAC**, **MAC** et **MLS** sous la forme de **MCS** : ur

Type de Sécurité	Nom	Description
TE	*Type enforcement*	Chaque objet a une étiquette appelé *type* pour un fichier et *domaine* pour un processus. La politique de sécurité définit l'interaction entre les types et les domaines.
RBAC	*Role Based Access Control*	Un utilisateur a un ou plusieurs rôles. Les droits sont attribués aux rôles.
MAC	*Mandatory Access Control*	L'accès aux objets est en fonction de la classification de l'objet (Très secret, Secret, Confidentiel, Public). L'administrateur définit la politique de sécurité et les utilisateurs s'y conforment.
MLS	*Multi-Level Security*	Les politiques de sécurité imposent que qu'un sujet doit dominer un objet pour pouvoir le lire tandis que l'objet doit dominer le sujet pour que ce dernier puisse y écrire.

Même quand le modèle SELinux de sécurité est actif, la sécurité type DAC est toujours active. Cependant dans le cas où la sécurité du type DAC autorise une action, SELinux va évaluer cette action par rapport à ses propres règles avant de l'autoriser.

SELinux évalue toujours des ***actions*** tentées par des ***sujets*** sur des ***objets***.

Dans le contexte de SELinux :

- un ***sujet*** est toujours un **processus**,
- un ***objet*** peut être un fichier, un répertoire, un autre processus ou une ressource système,
- une ***action*** est une **permission**.

Chaque ***classe d'objet*** possède un jeu de permissions possibles ou ***actions*** qui peuvent être uniques à la classe ou bien **héritées** d'autres classes.

2.2 - Définitions

Security Context

SELinux associe un *Security Context* (SC) à chaque ***objet*** et ***sujet*** du système.

Un SC prend la forme **identité:rôle:type:niveau** :

Nom	Descriptions
Identité	Le nom du propriétaire de l'objet. Une identité est associée à des rôles. Par défaut l'utilisateur à une identité de **user_u**.

Nom	Descriptions
Rôle	Essentiellement appliqué aux processus, le rôle est appelé une domaine. Dans le cas d'un rôle de fichier, celui-ci est toujours **object_r**. Un rôle se termine généralement par **_r**.
Type	Définit la classification de sécurité de l'objet. Un type se termine généralement par **_t**.
Niveau	Un niveau est un attribut de MLS et MCS. Une plage MLS est une paire de niveaux exprimée en utilisant la syntaxe *niveaubas-niveauhaut*. Chaque niveau est une paire exprimée en tant que sensibilitéhaut-sensibilitébas:catégoriehaut:catégoriebas par exemple s0-s0:c0.c1023. Il est important de noter que s0-s0 s'exprime aussi s0 et c0, c1, c2, c3 est exprimé c0.c3.

Sous RedHat 9, le fichier **/etc/selinux/targeted/setrans.conf** contient la correspondance entre les niveaux et leurs valeurs compréhensibles par l'utilisateur :

```
[root@redhat9 attributs]# cat /etc/selinux/targeted/setrans.conf
#
# Multi-Category Security translation table for SELinux
#
# Uncomment the following to disable translation libary
# disable=1
#
# Objects can be categorized with 0-1023 categories defined by the admin.
# Objects can be in more than one category at a time.
# Categories are stored in the system as c0-c1023.  Users can use this
# table to translate the categories into a more meaningful output.
# Examples:
# s0:c0=CompanyConfidential
# s0:c1=PatientRecord
# s0:c2=Unclassified
# s0:c3=TopSecret
# s0:c1,c3=CompanyConfidentialRedHat
s0=SystemLow
s0-s0:c0.c1023=SystemLow-SystemHigh
s0:c0.c1023=SystemHigh
```

Dans le contexte d'un SC pour un ***sujet***, le champ **identité** indique les privilèges de l'utilisateur SELinux utilisés par le ***sujet***.

Dans le contexte d'un SC pour un ***objet***, le champ **identité** indique à quel utilisateur SELinux appartient l'***objet***.

SELinux maintient sa propre liste d'utilisateurs, différente de la liste DAC de Linux. Il existe cependant une correspondance entre les deux listes de façon à ce que les utilisateurs MAC puissent être soumissent aux restrictions de SELinux :

```
[root@redhat9 attributs]# /usr/sbin/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       *
```

Domains et Types

Le **Domain** est l'endroit d'exécution d'un processus. Chaque processus a un **Domain**. Le **Domain** détermine les accès du processus.

Le **Domain** contient des ***objets*** et des ***sujets*** qui interagissent ensemble. Ce modèle, où chaque ***sujet*** se voit attribué à un **Domain** et où uniquement certaines opérations sont permises, est appelé ***Type Enforcement***.

Dans SELinux on utilise le mot :

- **Domain** pour un processus,
- **Type** pour un fichier.

Roles

Un **Rôle** est comme un utilisateur dans le système de sécurité DAC de Linux. Chaque utilisateur autorisé peut assumer l'identité du **Rôle** afin d'exécuter les commandes liées au **Rôle**.

Politiques de Sécurité

Une politique de sécurité définit les SC de chaque application. Elle définit des droits d'accès des domaines aux types. Il y a deux types de politique possible :

Politique	**Description**
targeted	Les politiques de sécurité ne s'appliquent qu'à certaines applications
mls	Multi Level Security protection

Les politiques de sécurité se trouvent dans le répertoire **/etc/selinux** :

```
[root@redhat9 attributs]# ls -lR /etc/selinux/ | more
/etc/selinux/:
total 8
-rw-r--r--. 1 root root 1187 Oct 19  2023 config
-rw-r--r--. 1 root root 2668 Dec 14  2023 semanage.conf
drwxr-xr-x. 5 root root  133 Sep 25 12:04 targeted

/etc/selinux/targeted:
total 16
-rw-r--r--. 1 root root 2367 Jun  5 11:17 booleans.subs_dist
drwxr-xr-x. 4 root root 4096 Sep 25 11:58 contexts
drwxr-xr-x. 2 root root    6 Jun  5 11:17 logins
drwxr-xr-x. 2 root root   23 Sep 25 12:04 policy
-rw-r--r--. 1 root root  607 Jun  5 11:17 setrans.conf
-rw-r--r--. 1 root root   73 Sep 25 12:04 seusers

/etc/selinux/targeted/contexts:
total 72
-rw-r--r--. 1 root root  262 Sep 25 11:58 customizable_types
-rw-r--r--. 1 root root  195 Jun  5 11:17 dbus_contexts
-rw-r--r--. 1 root root 1111 Jun  5 11:17 default_contexts
-rw-r--r--. 1 root root  114 Jun  5 11:17 default_type
-rw-r--r--. 1 root root   29 Jun  5 11:17 failsafe_context
drwxr-xr-x. 2 root root 4096 Sep 25 12:04 files
```

```
--More--
```

Afin d'utiliser SELinux en ligne de commande sous RedHat 9, il est nécessaire d'installer le paquet **setools-console** :

```
[root@redhat9 attributs]# dnf install setools-console -y
```

Pour consulter les statistiques de la politique, il convient d'utiliser la commande **seinfo** :

```
[root@redhat9 attributs]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             33 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             135    Permissions:         457
  Sensitivities:         1    Categories:         1024
  Types:              5155    Attributes:          259
  Users:                 8    Roles:                15
  Booleans:            360    Cond. Expr.:         393
  Allow:             65813    Neverallow:            0
  Auditallow:          176    Dontaudit:          8692
  Type_trans:       272792    Type_change:          94
  Type_member:          37    Range_trans:        6164
  Role allow:           40    Role_trans:          419
  Constraints:          70    Validatetrans:         0
  MLS Constrain:        72    MLS Val. Tran:         0
  Permissives:           5    Polcap:                6
  Defaults:              7    Typebounds:            0
  Allowxperm:            0    Neverallowxperm:       0
  Auditallowxperm:       0    Dontauditxperm:        0
  Ibendportcon:          0    Ibpkeycon:             0
  Initial SIDs:         27    Fs_use:               35
  Genfscon:            109    Portcon:             665
  Netifcon:              0    Nodecon:               0
```

Important : Notez ici le grand nombre de la catégorie **Dontaudit**.

Langage de Politiques

Un politique est composé de centaines de directives. Les principales directives sont :

allow

allow autorise l'accès d'un processus d'un domaine à des fichiers appartenant à un type donné. Le format de la directive est :

```
allow user_t domaine_t : file (read execute getattr) ;
```

Dans cette directive :

- user_t est le type de fichier,
- domaine_t est le domaine des processus qui sont autorisés par allow,
- file (droit1 droit2 etc) est la liste des permissions accordées.

Les permissions possibles sont :

- read
- write
- append
- execute
- getattr
- setattr
- lock
- link
- unlink

- rename
- ioctl

type

La directive **type** définit un type SELinux. Le type se termine généralement par **_t**.

auditallow, dontaudit

La directive **auditallow** demande l'écriture d'un message de type **avc** dans les journaux. Elle n'est associée à aucune restriction.

L'inverse peut être obtenue avec **dontaudit**, à savoir, cette directive demande à ce qu'il n'y ait pas de journalisation après une interdiction.

type_transition

Normalement quand un fichier est créé, il hérite du SC du répertoire parent. De même quand un processus SELinux active un nouveau processus, ce dernier s'exécute dans le même domaine que son parent. La directive type_transition permet de modifier ce comportement.

Décisions de SELinux

Il existe deux types de décisions auxquelles SELinux doit faire face :

- **Décisions d'Accès**
- **Décisions de Transition**

Décisions d'Accès

Dans ce type de décision SELinux doit décider d'accorder ou non la permission à :

- un ***sujet*** de faire quelque chose à un ***objet*** existant,

- un ***sujet*** de créer de nouvelles choses dans le **Domain**.

Décisions de Transition

Dans ce type de décision SELinux doit décider d'accorder ou non la permission :

- d'invoquer un processus dans un **Domain** différent du **Domain** courant du ***sujet***,
- de créer des ***objets*** dans différents **Types** que le répertoire parent de l'***objet***.

Commandes SELinux

Commande	**Description**
chcon	Changer le SC d'un fichier
audit2allow	Générer la source de la règle de sécurité à l'origine d'une erreur
restorecon	Restaurer le SC par défaut à un ou plusieurs fichiers
setfiles -n	Vérifier si les SC sont corrects
semodule	Gèrer les modules de politiques
semodule -i	Installer un module de politiques
checkmodule	Compiler un module
semodule_package	Créer un module installable par semodule
semanage	Administrer une politique
audit2allow -M	Créer un module à partir d'un message d'audit
sesearch	Recherche des règles SELinux
seinfo	Effectuer des recherches dans la politique
getsebool	Affiche l'état d'un booléen
getsebool -a	Affiche l'état de l'ensemble des booléens
sestatus -b	Affiche l'état de l'ensemble des booléens
setsebool	Modifie l'état d'un booléen
togglesebool	Bascule la valeur d'un booléen

Les Etats de SELinux

SELinux connait trois états :

Etat	Description
disabled	SELinux est inactif.
permissive	SELinux est actif mais tout est permis. Des interdictions ne font que de générer des messages d'erreurs dans les logs.
enforcing	SELinux est actif.

L'examen du contenu du fichier **/selinux/enforce** révèle une de deux valeurs qui correspondent à l'**état** de SELinux :

Valeur	Description
0	SELinux est en mode *permissive*
1	SELinux est en mode *enforcing*

La configuration de l'activation de SELinux ainsi que son état est effectuée grâce au fichier **/etc/selinux/config** :

```
[root@redhat9 attributs]# cat /etc/selinux/config

# 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.
# See also:
#
https://docs.fedoraproject.org/en-US/quick-docs/getting-started-with-selinux/#getting-started-with-selinux-selinu
x-states-and-modes
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
```

```
# to persistently set the bootloader to boot with selinux=0:
#
#    grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
#    grubby --update-kernel ALL --remove-args selinux
#
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Afin de connaître l'état de SELinux, il convient d'utiliser la commande **getenforce** :

```
[root@redhat9 attributs]# getenforce
Enforcing
```

Pour modifier l'état de SELinux, il convient d'utiliser la commande **setenforce** :

```
[root@redhat9 attributs]# setenforce permissive

[root@redhat9 attributs]# getenforce
Permissive
```

La commande **sestatus** vous informe sur la configuration de SELinux et notamment sur la version de la politique utilisée :

```
root@redhat9 attributs]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
```

```
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
```

Les différentes versions de politiques évolue en même temps que le noyau Linux.

La commande sestatus peut aussi prendre l'option -v :

```
[root@redhat9 attributs]# sestatus -v
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33

Process contexts:
Current context:                unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Init context:                   system_u:system_r:init_t:s0
/usr/sbin/sshd                  system_u:system_r:sshd_t:s0-s0:c0.c1023

File contexts:
Controlling terminal:           unconfined_u:object_r:user_devpts_t:s0
/etc/passwd                     system_u:object_r:passwd_file_t:s0
/etc/shadow                     system_u:object_r:shadow_t:s0
/bin/bash                       system_u:object_r:shell_exec_t:s0
/bin/login                      system_u:object_r:login_exec_t:s0
```

```
/bin/sh                         system_u:object_r:bin_t:s0 -> system_u:object_r:shell_exec_t:s0
/sbin/agetty                    system_u:object_r:getty_exec_t:s0
/sbin/init                      system_u:object_r:bin_t:s0 -> system_u:object_r:init_exec_t:s0
/usr/sbin/sshd                  system_u:object_r:sshd_exec_t:s0
```

Booléens

Les booléens permettent à des ensembles de règles d'être utilisées d'une manière alternative.

Pour visualiser l'état l'ensemble des booléens, il convient d'utiliser la commande **getsebool -a** :

```
[root@redhat9 attributs]# getsebool -a | more
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
auditadm_exec_content --> on
authlogin_nsswitch_use_ldap --> off
authlogin_radius --> off
authlogin_yubikey --> off
awstats_purge_apache_log_files --> off
boinc_execmem --> on
cdrecord_read_content --> off
cluster_can_network_connect --> off
cluster_manage_all_files --> off
cluster_use_execmem --> off
cobbler_anon_write --> off
cobbler_can_network_connect --> off
cobbler_use_cifs --> off
cobbler_use_nfs --> off
collectd_tcp_network_connect --> off
colord_use_nfs --> off
```

```
condor_tcp_network_connect --> off
conman_can_network --> off
conman_use_nfs --> off
container_connect_any --> off
container_manage_cgroup --> off
container_read_certs --> off
container_use_cephfs --> off
container_use_devices --> off
container_use_dri_devices --> on
container_use_ecryptfs --> off
container_user_exec_content --> on
cron_can_relabel --> off
cron_system_cronjob_use_shares --> off
cron_userdomain_transition --> on
cups_execmem --> off
cvs_read_shadow --> off
daemons_dontaudit_scheduling --> on
daemons_dump_core --> off
daemons_enable_cluster_mode --> off
daemons_use_tcp_wrapper --> off
daemons_use_tty --> off
dbadm_exec_content --> on
dbadm_manage_user_files --> off
dbadm_read_user_files --> off
deny_bluetooth --> off
deny_execmem --> off
deny_ptrace --> off
dhcpc_exec_iptables --> off
dhcpd_use_ldap --> off
dnsmasq_use_ipset --> off
domain_can_mmap_files --> off
--More--
```

ou la commande **sestatus -b** :

```
[root@redhat9 attributs]# sestatus -b | more
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   permissive
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33

Policy booleans:
abrt_anon_write                             off
abrt_handle_event                           off
abrt_upload_watch_anon_write                on
antivirus_can_scan_system                   off
antivirus_use_jit                           off
auditadm_exec_content                       on
authlogin_nsswitch_use_ldap                 off
authlogin_radius                            off
authlogin_yubikey                           off
awstats_purge_apache_log_files              off
boinc_execmem                               on
cdrecord_read_content                       off
cluster_can_network_connect                 off
cluster_manage_all_files                    off
cluster_use_execmem                         off
cobbler_anon_write                          off
cobbler_can_network_connect                 off
cobbler_use_cifs                            off
cobbler_use_nfs                             off
collectd_tcp_network_connect                off
colord_use_nfs                              off
```

```
condor_tcp_network_connect                  off
conman_can_network                          off
conman_use_nfs                              off
container_connect_any                       off
container_manage_cgroup                     off
container_read_certs                        off
container_use_cephfs                        off
container_use_devices                       off
container_use_dri_devices                   on
container_use_ecryptfs                      off
container_user_exec_content                 on
cron_can_relabel                            off
cron_system_cronjob_use_shares              off
cron_userdomain_transition                  on
cups_execmem                                off
cvs_read_shadow                             off
daemons_dontaudit_scheduling                on
daemons_dump_core                           off
daemons_enable_cluster_mode                 off
--More--
```

Pour fixer l'état d'un booléen, il convient d'utiliser la commande setsebool :

```
[root@redhat9 attributs]# setsebool antivirus_can_scan_system 1

[root@redhat9 attributs]# getsebool antivirus_can_scan_system
antivirus_can_scan_system --> on

[root@redhat9 attributs]# setsebool antivirus_can_scan_system 0

[root@redhat9 attributs]# getsebool antivirus_can_scan_system
antivirus_can_scan_system --> off
```

LAB #3 - Travailler avec SELinux

Afin reconstruire la politique actuelle **sans** les règles **dontaudit**, utilisez la commande **semodule** :

```
[root@redhat9 attributs]# semodule -DB
```

Vérifiez qu'il ne reste aucune règle de type **dontaudit** :

```
[root@redhat9 attributs]# seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:             33 (MLS enabled)
Target Policy:              selinux
Handle unknown classes:     allow
  Classes:             135    Permissions:         457
  Sensitivities:         1    Categories:         1024
  Types:              5145    Attributes:          259
  Users:                 8    Roles:                15
  Booleans:            356    Cond. Expr.:         385
  Allow:             65504    Neverallow:            0
  Auditallow:          176    Dontaudit:             0
  Type_trans:       271770    Type_change:          94
  Type_member:          37    Range_trans:        5931
  Role allow:           40    Role_trans:          417
  Constraints:          70    Validatetrans:         0
  MLS Constrain:        72    MLS Val. Tran:         0
  Permissives:           5    Polcap:                6
  Defaults:              7    Typebounds:            0
  Allowxperm:            0    Neverallowxperm:       0
  Auditallowxperm:       0    Dontauditxperm:        0
  Ibendportcon:          0    Ibpkeycon:             0
  Initial SIDs:         27    Fs_use:               35
  Genfscon:            109    Portcon:             665
```

```
  Netifcon:               0    Nodecon:               0
```

3.1 - Copier et Déplacer des Fichiers

Créez deux fichiers **file1** et **file2** en tant que l'utilisateur **trainee** puis visualisez les SC des fichiers :

```
[root@redhat9 attributs]# exit
logout

[trainee@redhat9 ~]$ touch file1 file2

[trainee@redhat9 ~]$ ls -Z file*
unconfined_u:object_r:user_home_t:s0 file1
unconfined_u:object_r:user_home_t:s0 file2
```

Notez que le type des deux fichiers est **user_home_t**.

Copiez maintenant le fichier **file1** vers **/tmp** en utilisant la commande **cp** et visualiser son SC :

```
[trainee@redhat9 ~]$ cp file1 /tmp

[trainee@redhat9 ~]$ ls -Z /tmp/file1
unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
```

Notez que le fichier ainsi copié a hérité du **type** du répertoire parent, à savoir **tmp_t**.

Déplacez maintenant le fichier **file2** dans le répertoire **/tmp** et contrôlez son SC :

```
[trainee@redhat9 ~]$ mv file2 /tmp

[trainee@redhat9 ~]$ ls -Z /tmp/file2
unconfined_u:object_r:user_home_t:s0 /tmp/file2
```

Notez que la commande **mv** maintient le **type** d'origine.

3.2 - Vérifier les SC des Processus

Il convient d'utiliser l'option **Z** avec la commande **ps** :

```
[trainee@redhat9 ~]$ ps auxZ | more
LABEL                             USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
system_u:system_r:init_t:s0       root           1  0.0  0.2 175884 21416 ?        Ss   Oct21   0:47
/usr/lib/systemd/systemd rhgb --switched-root --
system --deserialize 31
system_u:system_r:kernel_t:s0     root           2  0.0  0.0      0     0 ?        S    Oct21   0:00 [kthreadd]
system_u:system_r:kernel_t:s0     root           3  0.0  0.0      0     0 ?        I<   Oct21   0:00 [rcu_gp]
system_u:system_r:kernel_t:s0     root           4  0.0  0.0      0     0 ?        I<   Oct21   0:00 [rcu_par_gp]
system_u:system_r:kernel_t:s0     root           5  0.0  0.0      0     0 ?        I<   Oct21   0:00 [slub_flushwq]
system_u:system_r:kernel_t:s0     root           6  0.0  0.0      0     0 ?        I<   Oct21   0:00 [netns]
system_u:system_r:kernel_t:s0     root           8  0.0  0.0      0     0 ?        I<   Oct21   0:00 [kworker/0:0H-
events_highpri]
system_u:system_r:kernel_t:s0     root          10  0.0  0.0      0     0 ?        I<   Oct21   0:00 [mm_percpu_wq]
system_u:system_r:kernel_t:s0     root          12  0.0  0.0      0     0 ?        I    Oct21   0:00
[rcu_tasks_kthre]
system_u:system_r:kernel_t:s0     root          13  0.0  0.0      0     0 ?        I    Oct21   0:00
[rcu_tasks_rude_]
system_u:system_r:kernel_t:s0     root          14  0.0  0.0      0     0 ?        I    Oct21   0:00
[rcu_tasks_trace]
system_u:system_r:kernel_t:s0     root          15  0.0  0.0      0     0 ?        S    Oct21   0:00 [ksoftirqd/0]
system_u:system_r:kernel_t:s0     root          16  0.0  0.0      0     0 ?        I    Oct21   0:01 [rcu_preempt]
system_u:system_r:kernel_t:s0     root          17  0.0  0.0      0     0 ?        S    Oct21   0:00 [migration/0]
system_u:system_r:kernel_t:s0     root          18  0.0  0.0      0     0 ?        S    Oct21   0:00
[idle_inject/0]
system_u:system_r:kernel_t:s0     root          20  0.0  0.0      0     0 ?        S    Oct21   0:00 [cpuhp/0]
system_u:system_r:kernel_t:s0     root          21  0.0  0.0      0     0 ?        S    Oct21   0:00 [cpuhp/1]
system_u:system_r:kernel_t:s0     root          22  0.0  0.0      0     0 ?        S    Oct21   0:00
```

```
[idle_inject/1]
system_u:system_r:kernel_t:s0   root          23  0.0  0.0      0     0 ?        S    Oct21   0:00 [migration/1]
system_u:system_r:kernel_t:s0   root          24  0.0  0.0      0     0 ?        S    Oct21   0:00 [ksoftirqd/1]
system_u:system_r:kernel_t:s0   root          27  0.0  0.0      0     0 ?        S    Oct21   0:00 [cpuhp/2]
system_u:system_r:kernel_t:s0   root          28  0.0  0.0      0     0 ?        S    Oct21   0:00
[idle_inject/2]
system_u:system_r:kernel_t:s0   root          29  0.0  0.0      0     0 ?        S    Oct21   0:00 [migration/2]
system_u:system_r:kernel_t:s0   root          30  0.0  0.0      0     0 ?        S    Oct21   0:00 [ksoftirqd/2]
system_u:system_r:kernel_t:s0   root          32  0.0  0.0      0     0 ?        I<   Oct21   0:00 [kworker/2:0H-
events_highpri]
system_u:system_r:kernel_t:s0   root          33  0.0  0.0      0     0 ?        S    Oct21   0:00 [cpuhp/3]
system_u:system_r:kernel_t:s0   root          34  0.0  0.0      0     0 ?        S    Oct21   0:00
[idle_inject/3]
system_u:system_r:kernel_t:s0   root          35  0.0  0.0      0     0 ?        S    Oct21   0:00 [migration/3]
--More--
```

3.3 - Visualiser la SC d'un Utilisateur

Utilisez l'option **-Z** avec la commande **id** :

```
[trainee@redhat9 ~]$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Notez que vous ne pouvez pas consulter le SC d'un autre utilisateur :

```
[trainee@redhat9 ~]$ id root
uid=0(root) gid=0(root) groups=0(root)

[trainee@redhat9 ~]$ id -Z root
id: cannot print security context when user specified
```

3.4 - Vérifier la SC d'un fichier

Il convient d'utiliser la commande ls avec l'option **-Z** :

```
[trainee@redhat9 ~]$ cd /etc

[trainee@redhat9 etc]$ ls -Z l* -d
unconfined_u:object_r:ld_so_cache_t:s0 ld.so.cache              system_u:object_r:etc_t:s0 libssh
            system_u:object_r:etc_t:s0 ld.so.conf               system_u:object_r:etc_t:s0 libuser.conf
            system_u:object_r:etc_t:s0 ld.so.conf.d           system_u:object_r:locale_t:s0 locale.conf
            system_u:object_r:etc_t:s0 libaudit.conf          system_u:object_r:locale_t:s0 localtime
            system_u:object_r:etc_t:s0 libblockdev              system_u:object_r:etc_t:s0 login.defs
            system_u:object_r:etc_t:s0 libibverbs.d             system_u:object_r:etc_t:s0 logrotate.conf
            system_u:object_r:etc_t:s0 libnl                    system_u:object_r:etc_t:s0 logrotate.d
            system_u:object_r:etc_t:s0 libpaper.d               system_u:object_r:etc_t:s0 lsm
            system_u:object_r:etc_t:s0 libreport              system_u:object_r:lvm_etc_t:s0 lvm
```

3.5 - Troubleshooting SELinux

L'interprétation des messages journalisés de SELinux est souvent la clef d'un dépannage efficace et rapide.

Si le démon **auditd** est démarré, les messages de SELinux sont consignés dans le fichier **/var/log/audit/audit.log**. Dans le cas contraire, les mêmes messages sont consignés dans le fichier **/var/log/messages**. Dans les deux cas, chaque message de SELinux contient le mot clef **AVC** :

3.6 - La commande chcon

La commande **chcon** permet de modifier *temporairement* une SC.

Prenons le cas de la création d'un répertoire à la racine du système de fichiers afin d'y stocker les pages web du serveur apache :

```
[trainee@redhat9 etc]$ su -
```

```
Password: fenestros

[root@redhat9 ~]# mkdir /www
[root@redhat9 ~]# touch /www/index.html
```

Installez maintenant le serveur Apache :

```
[root@redhat9 ~]# dnf install httpd
```

Activez et démarrez le service **httpd** :

```
[root@redhat9 ~]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[root@redhat9 ~]# systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
[root@redhat9 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Tue 2024-10-22 10:15:49 CEST; 3s ago
       Docs: man:httpd.service(8)
   Main PID: 101100 (httpd)
     Status: "Started, listening on: port 80"
      Tasks: 177 (limit: 48800)
     Memory: 34.5M
        CPU: 86ms
     CGroup: /system.slice/httpd.service
             ├─101100 /usr/sbin/httpd -DFOREGROUND
             ├─101101 /usr/sbin/httpd -DFOREGROUND
             ├─101102 /usr/sbin/httpd -DFOREGROUND
             ├─101103 /usr/sbin/httpd -DFOREGROUND
```

```
             └─101104 /usr/sbin/httpd -DFOREGROUND

Oct 22 10:15:49 redhat9.ittraining.loc systemd[1]: Starting The Apache HTTP Server...
Oct 22 10:15:49 redhat9.ittraining.loc httpd[101100]: Server configured, listening on: port 80
Oct 22 10:15:49 redhat9.ittraining.loc systemd[1]: Started The Apache HTTP Server.
```

Modifiez ensuite la directive **DocumentRoot** dans le fichier **/etc/httpd/conf/httpd.conf** :

```
[root@redhat9 ~]# vi /etc/httpd/conf/httpd.conf
```

```
[...]
#DocumentRoot "/var/www/html"
DocumentRoot "/www"
[...]
```

Ajoutez les section **<Directory “/www”>**:

```
...
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

<Directory "/www">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
...
```

Créez le fichier **/www/index.html** :

```
[root@redhat9 ~]# vi /www/index.html

[root@redhat9 ~]# cat /www/index.html
<html>
<title>
This is a test
</title>
<body>
www test page
</body>
</html>
```

Modifiez ensuite le propriétaire et le groupe du répertoire **/www** et son contenu :

```
[root@redhat9 ~]# chown -R apache:apache /www
```

Dernièrement, créez un fichier index.html **vide** dans le répertoire **/var/www/html/** :

```
[root@redhat9 ~]# touch /var/www/html/index.html
```

Redémarrez maintenant le service httpd :

```
[root@redhat9 ~]# systemctl restart httpd.service
```

Installez le paquet **lynx** :

```
[root@redhat9 ~]# dnf install lynx -y
```

Consultez le site localhost en utilisant **lynx** :

```
[root@redhat9 ~]# lynx localhost
```

La commande **sealert** possède à la fois une interface graphique **et** un mode en ligne de commande :

```
[root@redhat9 ~]# sealert -a /var/log/audit/audit.log > /root/mylogfile.txt
```

Consultez le fichier **/root/mylogfile.txt** :

```
[root@redhat9 ~]# more /root/mylogfile.txt

found 24 alerts in /var/log/audit/audit.log
--------------------------------------------------------------------------------

SELinux is preventing /usr/bin/pkla-check-authorization from using the noatsecure access on a process.

*****  Plugin catchall (100. confidence) suggests   **************************

If you believe that pkla-check-authorization should be allowed noatsecure access on processes labeled
policykit_auth_t 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 'pkla-check-auth' --raw | audit2allow -M my-pklacheckauth
# semodule -X 300 -i my-pklacheckauth.pp


Additional Information:
Source Context                system_u:system_r:policykit_t:s0
Target Context                system_u:system_r:policykit_auth_t:s0
Target Objects                /lib64/ld-linux-x86-64.so.2 [ process ]
Source                        pkla-check-auth
Source Path                   /usr/bin/pkla-check-authorization
Port                          <Unknown>
Host                          <Unknown>
Source RPM Packages           polkit-pkla-compat-0.1-21.el9.x86_64
```

```
Target RPM Packages           glibc-2.34-100.el9_4.3.x86_64
SELinux Policy RPM            selinux-policy-targeted-38.1.35-2.el9_4.2.noarch
Local Policy RPM              selinux-policy-targeted-38.1.35-2.el9_4.2.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Permissive
Host Name                     redhat9.ittraining.loc
Platform                      Linux redhat9.ittraining.loc
                              5.14.0-427.37.1.el9_4.x86_64 #1 SMP
                              PREEMPT_DYNAMIC Fri Sep 13 12:41:50 EDT 2024
                              x86_64 x86_64
Alert Count                   10
First Seen                    2024-10-22 10:01:01 CEST
Last Seen                     2024-10-22 10:33:01 CEST
Local ID                      344c2abc-bac6-4064-ae22-411f0ce680cd

Raw Audit Messages
type=AVC msg=audit(1729585981.217:17543): avc:  denied  { noatsecure } for  pid=102284 comm="polkitd"
scontext=system_u:system_r:policykit_t:s0 tco
ntext=system_u:system_r:policykit_auth_t:s0 tclass=process permissive=1


type=AVC msg=audit(1729585981.217:17543): avc:  denied  { rlimitinh } for  pid=102284 comm="pkla-check-auth"
scontext=system_u:system_r:policykit_t
:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process permissive=1


--More--(1%)
```

Cherchez dans le fichier la chaine **Plugin catchall** de la section concernant apache :

```
...
*****  Plugin catchall (100. confidence) suggests   **************************
```

```
If you believe that httpd should have the net_admin capability 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 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp


Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                system_u:system_r:httpd_t:s0
Target Objects                Unknown [ capability ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          <Unknown>
Source RPM Packages           httpd-core-2.4.57-11.el9_4.1.x86_64
Target RPM Packages
SELinux Policy RPM            selinux-policy-targeted-38.1.35-2.el9_4.2.noarch
Local Policy RPM              selinux-policy-targeted-38.1.35-2.el9_4.2.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Permissive
Host Name                     redhat9.ittraining.loc
Platform                      Linux redhat9.ittraining.loc
                              5.14.0-427.37.1.el9_4.x86_64 #1 SMP
                              PREEMPT_DYNAMIC Fri Sep 13 12:41:50 EDT 2024
                              x86_64 x86_64
Alert Count                   9
First Seen                    2024-10-22 10:15:49 CEST
Last Seen                     2024-10-22 10:32:34 CEST
Local ID                      15ae5915-d5a6-4849-b0d1-e4829bfcb57e
```

```
Raw Audit Messages
type=AVC msg=audit(1729585954.475:17532): avc:  denied  { net_admin } for  pid=101828 comm="httpd" capability=12
scontext=system_u:system_r:httpd_
t:s0 tcontext=system_u:system_r:httpd_t:s0 tclass=capability permissive=1


type=SYSCALL msg=audit(1729585954.475:17532): arch=x86_64 syscall=setsockopt success=yes exit=0 a0=9 a1=1 a2=20
a3=7ffeb581bbe4 items=0 ppid=1 pid=
101828 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295
comm=httpd exe=/usr/sbin/httpd subj=system
_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=setsockopt AUID=unset UID=root GID=root EUID=root SUID=root
FSUID=root EGID=root SGID=root FSG
ID=root

Hash: httpd,httpd_t,httpd_t,capability,net_admin
```

Ce message a été généré parce que le repertoire /www ainsi que le fichier index.html ne possèdent pas le **type** nécessaire pour que le service apache puisse les utiliser :

```
[root@redhat9 ~]# ls -Z /www/index.html
unconfined_u:object_r:default_t:s0 /www/index.html
```

```
[root@redhat9 ~]# ls -Z /var/www/html/index.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/index.html
```

L'exemple ci-dessus nous montre clairement que le type pour **/www/index.html** est **default_t** or apache a besoin du type **httpd_sys_content_t** pour pouvoir accéder au fichier.

Modifiez donc la SC de /www et /www/index.html en utilisant la commande **chcon** :

```
[root@redhat9 ~]# chcon -Rv --type=httpd_sys_content_t /www
changing security context of '/www/index.html'
changing security context of '/www'
```

```
[root@redhat9 ~]# ls -Z /www/index.html
unconfined_u:object_r:httpd_sys_content_t:s0 /www/index.html
```

Afin de maintenir ces SC lors d'une **restauration des SC par défaut**, il convient d'utiliser la commande **semanage** afin d'appliquer la modification d'une manière définitive :

```
[root@redhat9 ~]# semanage fcontext -a -t httpd_sys_content_t "/www(/.*)?"
```

Les options de la commande chcon sont :

```
[root@redhat9 ~]# chcon --help
Usage: chcon [OPTION]... CONTEXT FILE...
  or:  chcon [OPTION]... [-u USER] [-r ROLE] [-l RANGE] [-t TYPE] FILE...
  or:  chcon [OPTION]... --reference=RFILE FILE...
Change the SELinux security context of each FILE to CONTEXT.
With --reference, change the security context of each FILE to that of RFILE.

Mandatory arguments to long options are mandatory for short options too.
      --dereference      affect the referent of each symbolic link (this is
                         the default), rather than the symbolic link itself
  -h, --no-dereference   affect symbolic links instead of any referenced file
  -u, --user=USER        set user USER in the target security context
  -r, --role=ROLE        set role ROLE in the target security context
  -t, --type=TYPE        set type TYPE in the target security context
  -l, --range=RANGE      set range RANGE in the target security context
      --no-preserve-root  do not treat '/' specially (the default)
      --preserve-root    fail to operate recursively on '/'
      --reference=RFILE  use RFILE's security context rather than specifying
                         a CONTEXT value
  -R, --recursive        operate on files and directories recursively
  -v, --verbose          output a diagnostic for every file processed

The following options modify how a hierarchy is traversed when the -R
option is also specified.  If more than one is specified, only the final
```

```
one takes effect.

  -H                     if a command line argument is a symbolic link
                         to a directory, traverse it
  -L                     traverse every symbolic link to a directory
                         encountered
  -P                     do not traverse any symbolic links (default)

      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/chcon>
or available locally via: info '(coreutils) chcon invocation'
```

3.7 - La commande restorecon

Pour illustrer l'utilisation de cette commande, créez les fichiers copy.html et move.html dans le répertoire /tmp :

```
[root@redhat9 ~]# cd /tmp ; touch copy.html move.html

[root@redhat9 tmp]# ls -Z | grep html
 unconfined_u:object_r:user_tmp_t:s0 copy.html
 unconfined_u:object_r:user_tmp_t:s0 move.html
```

Copiez le fichier copy.html vers /var/www/html et **déplacez** le fichier move.html vers la même cible :

```
[root@redhat9 tmp]# cp copy.html /var/www/html/

[root@redhat9 tmp]# mv move.html /var/www/html/

[root@redhat9 tmp]# ls -Z /var/www/html
unconfined_u:object_r:httpd_sys_content_t:s0 copy.html            unconfined_u:object_r:user_tmp_t:s0 move.html
```

```
unconfined_u:object_r:httpd_sys_content_t:s0 index.html
```

> **Important** : Notez ici que copy.html a pris le type du répertoire de destination tandis que move.html retient le type obtenu lors de la création.

Restaurez maintenant la SC par défaut de move.html compte tenu de son emplacement en utilisant la commande **restorecon** :

```
[root@redhat9 tmp]# restorecon -v /var/www/html/move.html
Relabeled /var/www/html/move.html from unconfined_u:object_r:user_tmp_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0

[root@redhat9 tmp]# ls -Z /var/www/html
unconfined_u:object_r:httpd_sys_content_t:s0 copy.html   unconfined_u:object_r:httpd_sys_content_t:s0 move.html
unconfined_u:object_r:httpd_sys_content_t:s0 index.html
```

3.8 - Le fichier /.autorelabel

En cas de besoin il est intéressant de pouvoir restaurer les SC par défaut sur l'ensemble des objets du système. Cette procédure est très simple à mettre en oeuvre. Il convient de créer le fichier **.autorelabel** à la racine et de redémarrer le système :

```
[root@redhat9 tmp]# touch /.autorelabel

[root@redhat9 tmp]# shutdown -r now
```

3.9 - La commande semanage

Pour illustrer l'utilisation de cette commande, considérez le besoin de mettre le service apache à l'écoute du port **8090** au lieu du port standard.

SELinux gère aussi l'accès aux ports par les différents serveurs. La liste complète des ports autorisés par serveur puet être visualiser à l'aide de la

commande **semanage** :

```
[trainee@redhat9 ~]$ su -
Password: fenestros

[root@redhat9 ~]# 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
```

Notez par exemple que le serveur apache est autorisé d'utiliser les ports suivants :

```
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
```

Dans le cas où on souhaite qu'apache utilise le port **8090** par exemple, il est nécessaire de créer la règle adéquate avec la commande semanage :

```
[root@redhat9 ~]# semanage port -a -t http_port_t -p tcp 8090
```

Vous noterez que le port 8090 a été ajouté à la liste des ports reconnus comme valides par SELinux :

```
[root@redhat9 ~]# 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      8090, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
```

Les options **semanage** sont :

[root@redhat9 ~]# semanage –help usage: semanage [-h]
{import,export,login,user,port,ibpkey,ibendport,interface,module,node,fcontext,boolean,permissive,dontaudit} …

semanage is used to configure certain elements of SELinux policy with-out requiring modification or recompilation from policy source.

positional arguments:

```
{import,export,login,user,port,ibpkey,ibendport,interface,module,node,fcontext,boolean,permissive,dontaudit}
  import              Import local customizations
  export              Output local customizations
  login               Manage login mappings between linux users and SELinux confined users
  user                Manage SELinux confined users (Roles and levels for an SELinux user)
  port                Manage network port type definitions
  ibpkey              Manage infiniband ibpkey type definitions
  ibendport           Manage infiniband end port type definitions
  interface           Manage network interface type definitions
  module              Manage SELinux policy modules
  node                Manage network node type definitions
  fcontext            Manage file context mapping definitions
  boolean             Manage booleans to selectively enable functionality
  permissive          Manage process type enforcement mode
  dontaudit           Disable/Enable dontaudit rules in policy
```

optional arguments:

1. h, –help show this help message and exit

3.10 - La commande audit2allow

La création d'un module de politique personnalisé se fait en utilisant la commande **audit2allow**. L'administrateur de sécurité à recours à la création de modules quand, et uniquement quand :

- la résolution du problème n'est pas possible en utilisant une des commandes précédemment citées,
- il n'existe pas de booléen capable de régler le problème.

Pour illustrer l'utilisation de cette commande, créez un nouveau répertoire pour les documents d'apache ainsi que la page d'accueil :

```
[root@redhat9 ~]# mkdir /www1

[root@redhat9 ~]# touch /www1/index.html
```

Éditez le fichier **/etc/httpd/conf/httpd.conf** :

```
[root@redhat9 ~]# vi /etc/httpd/conf/httpd.conf
```

```
[...]
#DocumentRoot "/var/www/html"
DocumentRoot "/www1"
[...]
```

Ajoutez les section **<Directory “/www1”>**:

```
...
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

<Directory "/www1">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
...
```

Créez le fichier **/www1/index.html** :

```
[root@redhat9 ~]# vi /www1/index.html

[root@redhat9 ~]# cat /www1/index.html
<html>
<title>
This is a test
</title>
<body>
www test page
</body>
</html>
```

Modifiez ensuite le propriétaire et le groupe du répertoire **/www1** et son contenu :

```
[root@redhat9 ~]# chown -R apache:apache /www1
```

Redémarrez le service httpd :

```
[root@redhat9 ~]# systemctl restart httpd.service
```

Consultez le site localhost en utilisant **lynx** :

```
[root@redhat9 ~]# lynx --dump localhost
                                Red Hat Logo
                      Red Hat Enterprise Linux Test Page

   This page is used to test the proper operation of the HTTP server after
   it has been installed. If you can read this page, it means that the
   HTTP server installed at this site is working properly.
     ________________________________________________________________

If you are a member of the general public:
```

```
   The fact that you are seeing this page indicates that the website you
   just visited is either experiencing problems, or is undergoing routine
   maintenance.

   If you would like to let the administrators of this website know that
   you've seen this page instead of the page you expected, you should send
   them e-mail. In general, mail sent to the name "webmaster" and directed
   to the website's domain should reach the appropriate person.

   For example, if you experienced problems while visiting
   www.example.com, you should send e-mail to "webmaster@example.com".

   For information on Red Hat Enterprise Linux, please visit the [1]Red
   Hat, Inc. website. The documentation for Red Hat Enterprise Linux is
   [2]available on the Red Hat, Inc. website.

     ____________________________________________________________________

If you are the website administrator:

   You may now add content to the webroot directory. Note that until you
   do so, people visiting your website will see this page, and not your
   content.

   For systems using the Apache HTTP Server: You may now add content to
   the directory /var/www/html/. Note that until you do so, people
   visiting your website will see this page, and not your content. To
   prevent this page from ever being used, follow the instructions in the
   file /etc/httpd/conf.d/welcome.conf.

   For systems using NGINX: You should now put your content in a location
   of your choice and edit the root configuration directive in the nginx
   configuration file /etc/nginx/nginx.conf.
   [3][ Powered by Red Hat Enterprise Linux ] [ Powered by Red Hat
   Enterprise Linux ]
```

```
   [4]Apache™ is a registered trademark of [5]the Apache Software
   Foundation in the United States and/or other countries.
   [6]NGINX™ is a registered trademark of [7]F5 Networks, Inc..

References

   1. http://www.redhat.com/
   2. http://www.redhat.com/docs/manuals/enterprise/
   3. https://access.redhat.com/products/red-hat-enterprise-linux
   4. https://apache.org/
   5. https://apache.org/
   6. https://nginx.com/
   7. https://www.f5.com/
```

Notez que cette fois SELinux est en mode enforcing :

```
[root@redhat9 ~]# getenforce
Enforcing
```

Le fichier **/var/log/audit/audit.log** contient maintenant des notifications de type **AVC** :

```
[root@redhat9 ~]# cat /var/log/audit/audit.log | grep AVC | tail
type=AVC msg=audit(1729587121.979:17697): avc:  denied  { noatsecure } for  pid=102767 comm="polkitd"
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process
permissive=1
type=AVC msg=audit(1729587121.979:17697): avc:  denied  { rlimitinh } for  pid=102767 comm="pkla-check-auth"
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process
permissive=1
type=AVC msg=audit(1729587121.979:17697): avc:  denied  { siginh } for  pid=102767 comm="pkla-check-auth"
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process
permissive=1
type=AVC msg=audit(1729587426.204:17741): avc:  denied  { net_admin } for  pid=102906 comm="systemd-tmpfile"
capability=12  scontext=system_u:system_r:systemd_tmpfiles_t:s0 tcontext=system_u:system_r:systemd_tmpfiles_t:s0
tclass=capability permissive=1
```

```
type=AVC msg=audit(1729587481.218:17751): avc:  denied  { noatsecure } for  pid=102921 comm="polkitd"
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process
permissive=1
type=AVC msg=audit(1729587481.218:17751): avc:  denied  { rlimitinh } for  pid=102921 comm="pkla-check-auth"
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process
permissive=1
type=AVC msg=audit(1729587481.218:17751): avc:  denied  { siginh } for  pid=102921 comm="pkla-check-auth"
scontext=system_u:system_r:policykit_t:s0 tcontext=system_u:system_r:policykit_auth_t:s0 tclass=process
permissive=1
type=AVC msg=audit(1729587592.855:17788): avc:  denied  { siginh } for  pid=102990 comm="bash"
scontext=system_u:system_r:init_t:s0 tcontext=system_u:system_r:initrc_t:s0 tclass=process permissive=1
type=AVC msg=audit(1729588933.891:326): avc:  denied  { getattr } for  pid=2749 comm="httpd"
path="/www1/index.html" dev="dm-0" ino=34189841 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
type=AVC msg=audit(1729588933.891:327): avc:  denied  { getattr } for  pid=2749 comm="httpd"
path="/www1/index.html" dev="dm-0" ino=34189841 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

A l'aide de la commande grep, il convient maintenant d'envoyer les messages d'erreurs en provenance du fichier **/var/log/audit/audit.log** sur l'entrée standard de la commande **audit2allow** afin de permettre celle-ci de créer des règles permettant l'autorisation de ce qui a été précédemment interdit par SELinux :

```
[root@redhat9 ~]# grep httpd_t /var/log/audit/audit.log | audit2allow -m httpdlocal > httpdlocal.te
```

L'examen du fichier **httpdlocal.te** révèle la création de ces règles :

```
[root@redhat9 ~]# cat httpdlocal.te

module httpdlocal 1.0;

require {
        type httpd_t;
        type default_t;
        class capability net_admin;
```

```
        class file { getattr map open read };
}

#============= httpd_t ==============
allow httpd_t default_t:file { getattr open read };

#!!!! This avc can be allowed using the boolean 'domain_can_mmap_files'
allow httpd_t default_t:file map;

#!!!! This avc has a dontaudit rule in the current policy
allow httpd_t self:capability net_admin;
```

L'audit du fichier terminé, il faut maintenant utiliser audit2allow pour fabriquer un module de politique :

```
[root@redhat9 ~]# grep httpd_t /var/log/audit/audit.log | audit2allow -M httpdlocal
******************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i httpdlocal.pp
```

Chargez maintenant le module dans la politique SELinux :

```
[root@redhat9 ~]# semodule -i httpdlocal.pp
```

Vérifiez que le module est chargé :

```
[root@redhat9 ~]# semodule -l | grep httpd
httpdlocal
```

Consultez le site localhost en utilisant **lynx** :

```
[root@redhat9 ~]# lynx --dump localhost
   www test page
```

Les options **audit2allow** sont :

```
[root@redhat9 ~]# audit2allow --help
Usage: audit2allow [options]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -b, --boot            audit messages since last boot conflicts with -i
  -a, --all             read input from audit log - conflicts with -i
  -p POLICY, --policy=POLICY
                        Policy file to use for analysis
  -d, --dmesg           read input from dmesg - conflicts with --all and
                        --input
  -i INPUT, --input=INPUT
                        read input from <input> - conflicts with -a
  -l, --lastreload      read input only after the last reload
  -r, --requires        generate require statements for rules
  -m MODULE, --module=MODULE
                        set the module name - implies --requires
  -M MODULE_PACKAGE, --module-package=MODULE_PACKAGE
                        generate a module package - conflicts with -o and -m
  -o OUTPUT, --output=OUTPUT
                        append output to <filename>, conflicts with -M
  -D, --dontaudit       generate policy with dontaudit rules
  -R, --reference       generate refpolicy style output
  -N, --noreference     do not generate refpolicy style output
  -v, --verbose         explain generated output
  -e, --explain         fully explain generated output
  -t TYPE, --type=TYPE  only process messages with a type that matches this
                        regex
  --perm-map=PERM_MAP   file name of perm map
  --interface-info=INTERFACE_INFO
                        file name of interface information
```

```
  -x, --xperms          generate extended permission rules
  -w, --why             Translates SELinux audit messages into a description
                        of why the access was denied
```

LAB #4 - Le Pare-feu Netfilter/iptables

Netfilter est composé de 5 *hooks* :

- NF_IP_PRE_ROUTING
- NF_IP_LOCAL_IN
- NF_IP_LOCAL_OUT
- NF_IP_FORWARD
- NF_IP_POSTROUTING

Ces hooks sont utilisés par deux branches, la première est celle concernée par les paquets qui entrent vers des services locaux :

- NF_IP_PRE_ROUTING > NF_IP_LOCAL_IN > NF_IP_LOCAL_OUT > NF_IP_POSTROUTING

tandis que la deuxième concerne les paquets qui traversent la passerelle:

- NF_IP_PRE_ROUTING > NF_IP_FORWARD > NF_IP_POSTROUTING

Si IPTABLES a été compilé en tant que module, son utilisation nécessite le chargement de plusieurs modules supplémentaires en fonction de la situation:

- iptable_filter
- iptable_mangle
- iptable_net
- etc

Netfilter est organisé en **tables**. La commande **iptables** de netfilter permet d'insérer des **policies** dans les **chaines**:

- La table **FILTER**
  - La chaîne INPUT

        - Concerne les paquets entrants
            - Policies: ACCEPT, DROP, REJECT
    - La chaîne OUTPUT
        - Concerne les paquets sortants
            - Policies: ACCEPT, DROP, REJECT
    - La chaîne FORWARD
        - Concerne les paquets traversant le par-feu.
            - Policies: ACCEPT, DROP, REJECT

Si aucune table n'est précisée, c'est la table FILTER qui s'applique par défaut.

- La table **NAT**
    - La chaîne PREROUTING
        - Permet de faire la translation d'adresse de destination
            - Cibles: SNAT, DNAT, MASQUERADE
    - La chaîne POSTROUTING
        - Permet de faire la translation d'adresse de la source
            - Cibles: SNAT, DNAT, MASQUERADE
    - Le cas spécifique OUTPUT
        - Permet la modification de la destination des paquets générés localement
- La table **MANGLE**
    - Permet le marquage de paquets générés localement (OUTPUT) et entrants (PREROUTING)

Les **policies** sont:

- ACCEPT
    - Permet d'accepter le paquet concerné
- DROP
    - Permet de rejeter le paquet concerné sans générer un message d'erreur
- REJECT
    - Permet de rejeter le paquet concerné en générant une message d'erreur

Les **cibles** sont:

- SNAT
  - Permet de modifier l'adresse source du paquet concerné
- DNAT
  - Permet de modifier l'adresse de destination du paquet concerné
- MASQUERADE
  - Permet de remplacer l'adresse IP privée de l'expéditeur par un socket public de la passerelle.

IPTABLES peut être configuré soit par des outils tels shorewall, soit en utilisant des lignes de commandes ou un script. Dans ce dernier cas, la ligne prend la forme:

```
# IPTABLES --action CHAINE --option1 --option2
```

Les actions sont:

Action	Abréviation	Déscription
- -append	-A	Ajouter une règle à la fin de la chaîne spécifiée
- -delete	-D	Supprimer une règle en spécifiant son numéro ou la règle à supprimer
- -replace	-R	Permet de remplacer la règle spécifée par son numéro
- -insert	-I	Permet d'insérer une règle à l'endroit spécifié
- -list	-L	Permet d'afficher des règles
- -flush	-F	Permet de vider toutes les règles d'une chaîne

Les options sont:

Option	Abréviation	Déscription
- -protocol	-p	Permet de spécifier un protocol - tcp, udp, icmp, all
- -source	-s	Permet de spécifier une adresse source
- -destination	-d	Permet de spécifier une adresse de destination
- -in-interface	-i	Permet de spécifier une interface réseau d'entrée
- -out-interface	-o	Permet de spécifier une interface réseau de sortie
- -fragment	-f	Permet de ne spécifier que les paquets fragmentés
- -source-port	-sport	Permet de spécifier un port source ou une plage de ports source
- -destination-port	-dport	Permet de spécifier un port de destination ou une plage de ports de destination

Option	Abréviation	Déscription
- -tcp-flags	s/o	Permet de spécifier un flag TCP à matcher - SYN, ACK, FIN, RST, URG, PSH, ALL, NONE
- -icmp-type	s/o	Permet de spécifier un type de paquet ICMP
- -mac-source	s/o	Permet de spécifier une adresse MAC

Les options spécifiques à NET sont:

- -to-destination	s/o	Permet de spécifier l'adresse de destination d'une translation
- -to-source	s/o	Permet spécifier l'adresse source d'une translation

Les options spécifiques aux LOGS sont:

- -log-level	s/o	Permet de spécifier le niveau de logs
- -log-prefix	s/o	Permet de spécifier un préfix pour les logs

L'option spécifique au STATEFUL est:

- -state	s/o	Permet de spécifier l'état du paquet à vérifier

Ce dernier cas fait référence au STATEFUL. Le STATEFUL est la capacité du par-feu à enregistrer dans une table spécifique, l'état des différentes connexions. Cette table s'appelle une **table d'état**. Le principe du fonctionnement de STATEFUL est simple, à savoir, si le paquet entrant appartient à une communication déjà établie, celui-ci n'est pas vérifié.

Il existe 4 états:

- NEW
  - Le paquet concerne une nouvelle connexion et contient donc un flag SYN à 1
- ESTABLISHED
  - Le paquet concerne une connexion déjà établie. Le paquet ne doit contenir **ni** flag SYN à 1, **ni** flag FIN à 1
- RELATED
  - Le paquet est d'une connexion qui présente une relation avec une autre connexion
- INVALID
  - La paquet provient d'une connexion anormale.

4.1 - La Configuration par firewalld

Firewalld utilise des **zones** - des jeux de règles pré-définis dans lesquels sont placés les interfaces :

- **trusted** - un réseau fiable. Dans ce cas tous les ports sont autorisés,
- **work**, **home**, **internal** - un réseau partiellement fiable. Dans ce cas quelques ports sont autorisés,
- **dmz**, **public**, **external** - un réseau non fiable. Dans ce cas peu de ports sont autorisés,
- **block**, **drop** - tout est interdit. La zone drop n'envoie pas de messages d'erreurs.

> **Important** - Une interface ne peut être que dans une zone à la fois tandis que plusieurs interfaces peuvent être dans la même zone.

Le service firewalld doit toujours être lancé :

```
[root@redhat9 ~]# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
     Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
     Active: active (running) since Tue 2024-10-22 11:02:04 CEST; 32min ago
       Docs: man:firewalld(1)
   Main PID: 795 (firewalld)
      Tasks: 2 (limit: 48800)
     Memory: 44.6M
        CPU: 510ms
     CGroup: /system.slice/firewalld.service
             └─795 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Oct 22 11:02:01 redhat9.ittraining.loc systemd[1]: Starting firewalld - dynamic firewall daemon...
Oct 22 11:02:04 redhat9.ittraining.loc systemd[1]: Started firewalld - dynamic firewall daemon.
```

4.2 - La Configuration de Base de firewalld

La configuration par défaut de firewalld se trouve dans **/usr/lib/firewalld** :

```
[root@redhat9 ~]# ls -l /usr/lib/firewalld/
total 20
drwxr-xr-x. 2 root root 4096 Sep 25 12:05 helpers
drwxr-xr-x. 2 root root 4096 Sep 25 12:05 icmptypes
drwxr-xr-x. 2 root root   23 Sep 25 12:05 ipsets
drwxr-xr-x. 2 root root   33 Sep 25 12:05 policies
drwxr-xr-x. 2 root root 8192 Sep 25 12:06 services
drwxr-xr-x. 2 root root  184 Sep 25 12:05 zones

[root@redhat9 ~]# ls -l /usr/lib/firewalld/zones/
total 40
-rw-r--r--. 1 root root 312 Nov  6  2023 block.xml
-rw-r--r--. 1 root root 306 Nov  6  2023 dmz.xml
-rw-r--r--. 1 root root 304 Nov  6  2023 drop.xml
-rw-r--r--. 1 root root 317 Nov  6  2023 external.xml
-rw-r--r--. 1 root root 410 Nov  6  2023 home.xml
-rw-r--r--. 1 root root 425 Nov  6  2023 internal.xml
-rw-r--r--. 1 root root 729 Feb 22  2024 nm-shared.xml
-rw-r--r--. 1 root root 356 Nov  6  2023 public.xml
-rw-r--r--. 1 root root 175 Nov  6  2023 trusted.xml
-rw-r--r--. 1 root root 352 Nov  6  2023 work.xml

[root@redhat9 ~]# ls -l /usr/lib/firewalld/services/
total 884
-rw-r--r--. 1 root root 352 Nov  6  2023 afp.xml
-rw-r--r--. 1 root root 399 Nov  6  2023 amanda-client.xml
-rw-r--r--. 1 root root 427 Nov  6  2023 amanda-k5-client.xml
-rw-r--r--. 1 root root 283 Nov  6  2023 amqps.xml
-rw-r--r--. 1 root root 273 Nov  6  2023 amqp.xml
```

```
-rw-r--r--. 1 root root 285 Nov  6  2023 apcupsd.xml
-rw-r--r--. 1 root root 301 Nov  6  2023 audit.xml
-rw-r--r--. 1 root root 436 Nov  6  2023 ausweisapp2.xml
-rw-r--r--. 1 root root 320 Nov  6  2023 bacula-client.xml
-rw-r--r--. 1 root root 346 Nov  6  2023 bacula.xml
-rw-r--r--. 1 root root 390 Nov  6  2023 bareos-director.xml
-rw-r--r--. 1 root root 255 Nov  6  2023 bareos-filedaemon.xml
-rw-r--r--. 1 root root 316 Nov  6  2023 bareos-storage.xml
-rw-r--r--. 1 root root 429 Nov  6  2023 bb.xml
-rw-r--r--. 1 root root 339 Nov  6  2023 bgp.xml
-rw-r--r--. 1 root root 275 Nov  6  2023 bitcoin-rpc.xml
-rw-r--r--. 1 root root 307 Nov  6  2023 bitcoin-testnet-rpc.xml
-rw-r--r--. 1 root root 281 Nov  6  2023 bitcoin-testnet.xml
-rw-r--r--. 1 root root 244 Nov  6  2023 bitcoin.xml
-rw-r--r--. 1 root root 410 Nov  6  2023 bittorrent-lsd.xml
-rw-r--r--. 1 root root 222 Nov  6  2023 ceph-exporter.xml
-rw-r--r--. 1 root root 294 Nov  6  2023 ceph-mon.xml
-rw-r--r--. 1 root root 329 Nov  6  2023 ceph.xml
-rw-r--r--. 1 root root 168 Nov  6  2023 cfengine.xml
-rw-r--r--. 1 root root 234 Nov  6  2023 checkmk-agent.xml
-rw-r--r--. 1 root root 211 Nov  6  2023 cockpit.xml
-rw-r--r--. 1 root root 296 Nov  6  2023 collectd.xml
-rw-r--r--. 1 root root 260 Nov  6  2023 condor-collector.xml
-rw-r--r--. 1 root root 343 Nov  6  2023 cratedb.xml
-rw-r--r--. 1 root root 296 Nov  6  2023 ctdb.xml
-rw-r--r--. 1 root root 981 Nov  6  2023 dds-multicast.xml
-rw-r--r--. 1 root root 947 Nov  6  2023 dds-unicast.xml
-rw-r--r--. 1 root root 574 Nov  6  2023 dds.xml
-rw-r--r--. 1 root root 305 Nov  6  2023 dhcpv6-client.xml
-rw-r--r--. 1 root root 234 Nov  6  2023 dhcpv6.xml
-rw-r--r--. 1 root root 227 Nov  6  2023 dhcp.xml
-rw-r--r--. 1 root root 205 Nov  6  2023 distcc.xml
-rw-r--r--. 1 root root 318 Nov  6  2023 dns-over-tls.xml
-rw-r--r--. 1 root root 346 Nov  6  2023 dns.xml
```

```
-rw-r--r--. 1 root root 374 Nov  6  2023 docker-registry.xml
-rw-r--r--. 1 root root 391 Nov  6  2023 docker-swarm.xml
-rw-r--r--. 1 root root 228 Nov  6  2023 dropbox-lansync.xml
-rw-r--r--. 1 root root 338 Nov  6  2023 elasticsearch.xml
-rw-r--r--. 1 root root 304 Nov  6  2023 etcd-client.xml
-rw-r--r--. 1 root root 304 Nov  6  2023 etcd-server.xml
-rw-r--r--. 1 root root 224 Nov  6  2023 finger.xml
-rw-r--r--. 1 root root 270 Nov  6  2023 foreman-proxy.xml
-rw-r--r--. 1 root root 408 Nov  6  2023 foreman.xml
-rw-r--r--. 1 root root 709 Nov  6  2023 freeipa-4.xml
-rw-r--r--. 1 root root 489 Nov  6  2023 freeipa-ldaps.xml
-rw-r--r--. 1 root root 488 Nov  6  2023 freeipa-ldap.xml
-rw-r--r--. 1 root root 242 Nov  6  2023 freeipa-replication.xml
-rw-r--r--. 1 root root 657 Nov  6  2023 freeipa-trust.xml
-rw-r--r--. 1 root root 361 Nov  6  2023 ftp.xml
-rw-r--r--. 1 root root 292 Nov  6  2023 galera.xml
-rw-r--r--. 1 root root 184 Nov  6  2023 ganglia-client.xml
-rw-r--r--. 1 root root 176 Nov  6  2023 ganglia-master.xml
-rw-r--r--. 1 root root 212 Nov  6  2023 git.xml
-rw-r--r--. 1 root root 406 Nov  6  2023 gpsd.xml
-rw-r--r--. 1 root root 218 Nov  6  2023 grafana.xml
-rw-r--r--. 1 root root 119 Nov  6  2023 gre.xml
-rw-r--r--. 1 root root 608 Nov  6  2023 high-availability.xml
-rw-r--r--. 1 root root 336 Nov  6  2023 http3.xml
-rw-r--r--. 1 root root 448 Nov  6  2023 https.xml
-rw-r--r--. 1 root root 353 Nov  6  2023 http.xml
-rw-r--r--. 1 root root 293 Nov  6  2023 ident.xml
-rw-r--r--. 1 root root 372 Nov  6  2023 imaps.xml
-rw-r--r--. 1 root root 327 Nov  6  2023 imap.xml
-rw-r--r--. 1 root root 315 Nov  6  2023 ipfs.xml
-rw-r--r--. 1 root root 454 Nov  6  2023 ipp-client.xml
-rw-r--r--. 1 root root 427 Nov  6  2023 ipp.xml
-rw-r--r--. 1 root root 895 Nov  6  2023 ipsec.xml
-rw-r--r--. 1 root root 255 Nov  6  2023 ircs.xml
```

```
-rw-r--r--. 1 root root 247 Nov  6  2023 irc.xml
-rw-r--r--. 1 root root 264 Nov  6  2023 iscsi-target.xml
-rw-r--r--. 1 root root 358 Nov  6  2023 isns.xml
-rw-r--r--. 1 root root 213 Nov  6  2023 jenkins.xml
-rw-r--r--. 1 root root 182 Nov  6  2023 kadmin.xml
-rw-r--r--. 1 root root 272 Nov  6  2023 kdeconnect.xml
-rw-r--r--. 1 root root 233 Nov  6  2023 kerberos.xml
-rw-r--r--. 1 root root 384 Nov  6  2023 kibana.xml
-rw-r--r--. 1 root root 249 Nov  6  2023 klogin.xml
-rw-r--r--. 1 root root 221 Nov  6  2023 kpasswd.xml
-rw-r--r--. 1 root root 182 Nov  6  2023 kprop.xml
-rw-r--r--. 1 root root 242 Nov  6  2023 kshell.xml
-rw-r--r--. 1 root root 308 Nov  6  2023 kube-apiserver.xml
-rw-r--r--. 1 root root 204 Nov  6  2023 kube-api.xml
-rw-r--r--. 1 root root 289 Nov  6  2023 kube-controller-manager-secure.xml
-rw-r--r--. 1 root root 280 Nov  6  2023 kube-controller-manager.xml
-rw-r--r--. 1 root root 560 Nov  6  2023 kube-control-plane-secure.xml
-rw-r--r--. 1 root root 537 Nov  6  2023 kube-control-plane.xml
-rw-r--r--. 1 root root 244 Nov  6  2023 kubelet-readonly.xml
-rw-r--r--. 1 root root 212 Nov  6  2023 kubelet-worker.xml
-rw-r--r--. 1 root root 239 Nov  6  2023 kubelet.xml
-rw-r--r--. 1 root root 224 Nov  6  2023 kube-nodeport-services.xml
-rw-r--r--. 1 root root 328 Nov  6  2023 kube-scheduler-secure.xml
-rw-r--r--. 1 root root 319 Nov  6  2023 kube-scheduler.xml
-rw-r--r--. 1 root root 374 Nov  6  2023 kube-worker.xml
-rw-r--r--. 1 root root 232 Nov  6  2023 ldaps.xml
-rw-r--r--. 1 root root 199 Nov  6  2023 ldap.xml
-rw-r--r--. 1 root root 385 Nov  6  2023 libvirt-tls.xml
-rw-r--r--. 1 root root 389 Nov  6  2023 libvirt.xml
-rw-r--r--. 1 root root 269 Nov  6  2023 lightning-network.xml
-rw-r--r--. 1 root root 468 Nov  6  2023 llmnr-client.xml
-rw-r--r--. 1 root root 410 Nov  6  2023 llmnr-tcp.xml
-rw-r--r--. 1 root root 463 Nov  6  2023 llmnr-udp.xml
-rw-r--r--. 1 root root 519 Nov  6  2023 llmnr.xml
```

```
-rw-r--r--. 1 root root 349 Nov  6  2023 managesieve.xml
-rw-r--r--. 1 root root 432 Nov  6  2023 matrix.xml
-rw-r--r--. 1 root root 424 Nov  6  2023 mdns.xml
-rw-r--r--. 1 root root 245 Nov  6  2023 memcache.xml
-rw-r--r--. 1 root root 334 Nov  6  2023 minidlna.xml
-rw-r--r--. 1 root root 237 Nov  6  2023 mongodb.xml
-rw-r--r--. 1 root root 473 Nov  6  2023 mosh.xml
-rw-r--r--. 1 root root 211 Nov  6  2023 mountd.xml
-rw-r--r--. 1 root root 296 Nov  6  2023 mqtt-tls.xml
-rw-r--r--. 1 root root 287 Nov  6  2023 mqtt.xml
-rw-r--r--. 1 root root 170 Nov  6  2023 mssql.xml
-rw-r--r--. 1 root root 180 Nov  6  2023 ms-wbt.xml
-rw-r--r--. 1 root root 242 Nov  6  2023 murmur.xml
-rw-r--r--. 1 root root 171 Nov  6  2023 mysql.xml
-rw-r--r--. 1 root root 250 Nov  6  2023 nbd.xml
-rw-r--r--. 1 root root 309 Nov  6  2023 nebula.xml
-rw-r--r--. 1 root root 262 Nov  6  2023 netbios-ns.xml
-rw-r--r--. 1 root root 243 Nov  6  2023 netdata-dashboard.xml
-rw-r--r--. 1 root root 342 Nov  6  2023 nfs3.xml
-rw-r--r--. 1 root root 324 Nov  6  2023 nfs.xml
-rw-r--r--. 1 root root 293 Nov  6  2023 nmea-0183.xml
-rw-r--r--. 1 root root 247 Nov  6  2023 nrpe.xml
-rw-r--r--. 1 root root 389 Nov  6  2023 ntp.xml
-rw-r--r--. 1 root root 368 Nov  6  2023 nut.xml
-rw-r--r--. 1 root root 335 Nov  6  2023 openvpn.xml
-rw-r--r--. 1 root root 260 Nov  6  2023 ovirt-imageio.xml
-rw-r--r--. 1 root root 343 Nov  6  2023 ovirt-storageconsole.xml
-rw-r--r--. 1 root root 235 Nov  6  2023 ovirt-vmconsole.xml
-rw-r--r--. 1 root root 869 Nov  6  2023 plex.xml
-rw-r--r--. 1 root root 433 Nov  6  2023 pmcd.xml
-rw-r--r--. 1 root root 474 Nov  6  2023 pmproxy.xml
-rw-r--r--. 1 root root 544 Nov  6  2023 pmwebapis.xml
-rw-r--r--. 1 root root 460 Nov  6  2023 pmwebapi.xml
-rw-r--r--. 1 root root 357 Nov  6  2023 pop3s.xml
```

```
-rw-r--r--. 1 root root 348 Nov  6  2023 pop3.xml
-rw-r--r--. 1 root root 181 Nov  6  2023 postgresql.xml
-rw-r--r--. 1 root root 509 Nov  6  2023 privoxy.xml
-rw-r--r--. 1 root root 226 Nov  6  2023 prometheus-node-exporter.xml
-rw-r--r--. 1 root root 213 Nov  6  2023 prometheus.xml
-rw-r--r--. 1 root root 261 Nov  6  2023 proxy-dhcp.xml
-rw-r--r--. 1 root root 262 Nov  6  2023 ps2link.xml
-rw-r--r--. 1 root root 173 Nov  6  2023 ps3netsrv.xml
-rw-r--r--. 1 root root 424 Nov  6  2023 ptp.xml
-rw-r--r--. 1 root root 414 Nov  6  2023 pulseaudio.xml
-rw-r--r--. 1 root root 297 Nov  6  2023 puppetmaster.xml
-rw-r--r--. 1 root root 273 Nov  6  2023 quassel.xml
-rw-r--r--. 1 root root 520 Nov  6  2023 radius.xml
-rw-r--r--. 1 root root 183 Nov  6  2023 rdp.xml
-rw-r--r--. 1 root root 212 Nov  6  2023 redis-sentinel.xml
-rw-r--r--. 1 root root 268 Nov  6  2023 redis.xml
-rw-r--r--. 1 root root 381 Nov  6  2023 RH-Satellite-6-capsule.xml
-rw-r--r--. 1 root root 556 Nov  6  2023 RH-Satellite-6.xml
-rw-r--r--. 1 root root 214 Nov  6  2023 rpc-bind.xml
-rw-r--r--. 1 root root 213 Nov  6  2023 rquotad.xml
-rw-r--r--. 1 root root 310 Nov  6  2023 rsh.xml
-rw-r--r--. 1 root root 311 Nov  6  2023 rsyncd.xml
-rw-r--r--. 1 root root 350 Nov  6  2023 rtsp.xml
-rw-r--r--. 1 root root 329 Nov  6  2023 salt-master.xml
-rw-r--r--. 1 root root 339 Nov  6  2023 samba-client.xml
-rw-r--r--. 1 root root 782 Nov  6  2023 samba-dc.xml
-rw-r--r--. 1 root root 382 Nov  6  2023 samba.xml
-rw-r--r--. 1 root root 324 Nov  6  2023 sane.xml
-rw-r--r--. 1 root root 283 Nov  6  2023 sips.xml
-rw-r--r--. 1 root root 496 Nov  6  2023 sip.xml
-rw-r--r--. 1 root root 299 Nov  6  2023 slp.xml
-rw-r--r--. 1 root root 231 Nov  6  2023 smtp-submission.xml
-rw-r--r--. 1 root root 577 Nov  6  2023 smtps.xml
-rw-r--r--. 1 root root 550 Nov  6  2023 smtp.xml
```

```
-rw-r--r--. 1 root root 359 Nov  6  2023 snmptls-trap.xml
-rw-r--r--. 1 root root 390 Nov  6  2023 snmptls.xml
-rw-r--r--. 1 root root 308 Nov  6  2023 snmptrap.xml
-rw-r--r--. 1 root root 342 Nov  6  2023 snmp.xml
-rw-r--r--. 1 root root 405 Nov  6  2023 spideroak-lansync.xml
-rw-r--r--. 1 root root 275 Nov  6  2023 spotify-sync.xml
-rw-r--r--. 1 root root 173 Nov  6  2023 squid.xml
-rw-r--r--. 1 root root 421 Nov  6  2023 ssdp.xml
-rw-r--r--. 1 root root 463 Nov  6  2023 ssh.xml
-rw-r--r--. 1 root root 631 Nov  6  2023 steam-streaming.xml
-rw-r--r--. 1 root root 287 Nov  6  2023 svdrp.xml
-rw-r--r--. 1 root root 231 Nov  6  2023 svn.xml
-rw-r--r--. 1 root root 297 Nov  6  2023 syncthing-gui.xml
-rw-r--r--. 1 root root 414 Nov  6  2023 syncthing-relay.xml
-rw-r--r--. 1 root root 350 Nov  6  2023 syncthing.xml
-rw-r--r--. 1 root root 496 Nov  6  2023 synergy.xml
-rw-r--r--. 1 root root 444 Nov  6  2023 syslog-tls.xml
-rw-r--r--. 1 root root 329 Nov  6  2023 syslog.xml
-rw-r--r--. 1 root root 393 Nov  6  2023 telnet.xml
-rw-r--r--. 1 root root 252 Nov  6  2023 tentacle.xml
-rw-r--r--. 1 root root 424 Nov  6  2023 tftp.xml
-rw-r--r--. 1 root root 221 Nov  6  2023 tile38.xml
-rw-r--r--. 1 root root 336 Nov  6  2023 tinc.xml
-rw-r--r--. 1 root root 771 Nov  6  2023 tor-socks.xml
-rw-r--r--. 1 root root 244 Nov  6  2023 transmission-client.xml
-rw-r--r--. 1 root root 264 Nov  6  2023 upnp-client.xml
-rw-r--r--. 1 root root 593 Nov  6  2023 vdsm.xml
-rw-r--r--. 1 root root 475 Nov  6  2023 vnc-server.xml
-rw-r--r--. 1 root root 443 Nov  6  2023 warpinator.xml
-rw-r--r--. 1 root root 310 Nov  6  2023 wbem-https.xml
-rw-r--r--. 1 root root 352 Nov  6  2023 wbem-http.xml
-rw-r--r--. 1 root root 285 Nov  6  2023 wireguard.xml
-rw-r--r--. 1 root root 355 Nov  6  2023 ws-discovery-client.xml
-rw-r--r--. 1 root root 320 Nov  6  2023 ws-discovery-tcp.xml
```

```
-rw-r--r--. 1 root root 375 Nov  6  2023 ws-discovery-udp.xml
-rw-r--r--. 1 root root 357 Nov  6  2023 ws-discovery.xml
-rw-r--r--. 1 root root 323 Nov  6  2023 wsmans.xml
-rw-r--r--. 1 root root 316 Nov  6  2023 wsman.xml
-rw-r--r--. 1 root root 329 Nov  6  2023 xdmcp.xml
-rw-r--r--. 1 root root 509 Nov  6  2023 xmpp-bosh.xml
-rw-r--r--. 1 root root 488 Nov  6  2023 xmpp-client.xml
-rw-r--r--. 1 root root 264 Nov  6  2023 xmpp-local.xml
-rw-r--r--. 1 root root 545 Nov  6  2023 xmpp-server.xml
-rw-r--r--. 1 root root 314 Nov  6  2023 zabbix-agent.xml
-rw-r--r--. 1 root root 315 Nov  6  2023 zabbix-server.xml
-rw-r--r--. 1 root root 242 Nov  6  2023 zerotier.xml

[root@redhat9 ~]# ls -l /usr/lib/firewalld/icmptypes/
total 180
-rw-r--r--. 1 root root 385 Nov  6  2023 address-unreachable.xml
-rw-r--r--. 1 root root 258 Nov  6  2023 bad-header.xml
-rw-r--r--. 1 root root 293 Nov  6  2023 beyond-scope.xml
-rw-r--r--. 1 root root 279 Nov  6  2023 communication-prohibited.xml
-rw-r--r--. 1 root root 222 Nov  6  2023 destination-unreachable.xml
-rw-r--r--. 1 root root 173 Nov  6  2023 echo-reply.xml
-rw-r--r--. 1 root root 210 Nov  6  2023 echo-request.xml
-rw-r--r--. 1 root root 261 Nov  6  2023 failed-policy.xml
-rw-r--r--. 1 root root 280 Nov  6  2023 fragmentation-needed.xml
-rw-r--r--. 1 root root 266 Nov  6  2023 host-precedence-violation.xml
-rw-r--r--. 1 root root 257 Nov  6  2023 host-prohibited.xml
-rw-r--r--. 1 root root 242 Nov  6  2023 host-redirect.xml
-rw-r--r--. 1 root root 239 Nov  6  2023 host-unknown.xml
-rw-r--r--. 1 root root 247 Nov  6  2023 host-unreachable.xml
-rw-r--r--. 1 root root 229 Nov  6  2023 ip-header-bad.xml
-rw-r--r--. 1 root root 355 Nov  6  2023 neighbour-advertisement.xml
-rw-r--r--. 1 root root 457 Nov  6  2023 neighbour-solicitation.xml
-rw-r--r--. 1 root root 250 Nov  6  2023 network-prohibited.xml
-rw-r--r--. 1 root root 248 Nov  6  2023 network-redirect.xml
```

```
-rw-r--r--. 1 root root 239 Nov  6  2023 network-unknown.xml
-rw-r--r--. 1 root root 247 Nov  6  2023 network-unreachable.xml
-rw-r--r--. 1 root root 239 Nov  6  2023 no-route.xml
-rw-r--r--. 1 root root 328 Nov  6  2023 packet-too-big.xml
-rw-r--r--. 1 root root 225 Nov  6  2023 parameter-problem.xml
-rw-r--r--. 1 root root 233 Nov  6  2023 port-unreachable.xml
-rw-r--r--. 1 root root 256 Nov  6  2023 precedence-cutoff.xml
-rw-r--r--. 1 root root 249 Nov  6  2023 protocol-unreachable.xml
-rw-r--r--. 1 root root 185 Nov  6  2023 redirect.xml
-rw-r--r--. 1 root root 244 Nov  6  2023 reject-route.xml
-rw-r--r--. 1 root root 241 Nov  6  2023 required-option-missing.xml
-rw-r--r--. 1 root root 227 Nov  6  2023 router-advertisement.xml
-rw-r--r--. 1 root root 223 Nov  6  2023 router-solicitation.xml
-rw-r--r--. 1 root root 248 Nov  6  2023 source-quench.xml
-rw-r--r--. 1 root root 236 Nov  6  2023 source-route-failed.xml
-rw-r--r--. 1 root root 253 Nov  6  2023 time-exceeded.xml
-rw-r--r--. 1 root root 233 Nov  6  2023 timestamp-reply.xml
-rw-r--r--. 1 root root 228 Nov  6  2023 timestamp-request.xml
-rw-r--r--. 1 root root 258 Nov  6  2023 tos-host-redirect.xml
-rw-r--r--. 1 root root 257 Nov  6  2023 tos-host-unreachable.xml
-rw-r--r--. 1 root root 272 Nov  6  2023 tos-network-redirect.xml
-rw-r--r--. 1 root root 269 Nov  6  2023 tos-network-unreachable.xml
-rw-r--r--. 1 root root 293 Nov  6  2023 ttl-zero-during-reassembly.xml
-rw-r--r--. 1 root root 256 Nov  6  2023 ttl-zero-during-transit.xml
-rw-r--r--. 1 root root 259 Nov  6  2023 unknown-header-type.xml
-rw-r--r--. 1 root root 249 Nov  6  2023 unknown-option.xml
```

Ces fichiers sont au format **xml**, par exemple :

```
[root@redhat9 ~]# cat /usr/lib/firewalld/zones/home.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Home</short>
  <description>For use in home areas. You mostly trust the other computers on networks to not harm your computer.
```

```
Only selected incoming connections are accepted.</description>
  <service name="ssh"/>
  <service name="mdns"/>
  <service name="samba-client"/>
  <service name="dhcpv6-client"/>
  <service name="cockpit"/>
  <forward/>
</zone>
```

La configuration de firewalld ainsi que les définitions et règles personnalisées se trouvent dans **/etc/firewalld** :

```
[root@redhat9 ~]# ls -l /etc/firewalld/
total 8
-rw-r--r--. 1 root root 2483 Nov  6  2023 firewalld.conf
drwxr-x---. 2 root root    6 Nov  6  2023 helpers
drwxr-x---. 2 root root    6 Nov  6  2023 icmptypes
drwxr-x---. 2 root root    6 Nov  6  2023 ipsets
-rw-r--r--. 1 root root  271 Nov  6  2023 lockdown-whitelist.xml
drwxr-x---. 2 root root    6 Nov  6  2023 policies
drwxr-x---. 2 root root    6 Nov  6  2023 services
drwxr-x---. 2 root root   46 Nov  6  2023 zones

[root@redhat9 ~]# ls -l /etc/firewalld/zones/
total 8
-rw-r--r--. 1 root root 356 Oct 19  2023 public.xml
-rw-r--r--. 1 root root 356 Oct 19  2023 public.xml.old

[root@redhat9 ~]# ls -l /etc/firewalld/services/
total 0

[root@redhat9 ~]# ls -l /etc/firewalld/icmptypes/
total 0
```

Le fichier de configuration de firewalld est **/etc/firewalld/firewalld.conf** :

```
[root@redhat9 ~]# cat /etc/firewalld/firewalld.conf
# firewalld config file

# default zone
# The default zone used if an empty zone string is used.
# Default: public
DefaultZone=public

# Clean up on exit
# If set to no or false the firewall configuration will not get cleaned up
# on exit or stop of firewalld.
# Default: yes
CleanupOnExit=yes

# Clean up kernel modules on exit
# If set to yes or true the firewall related kernel modules will be
# unloaded on exit or stop of firewalld. This might attempt to unload
# modules not originally loaded by firewalld.
# Default: no
CleanupModulesOnExit=no

# Lockdown
# If set to enabled, firewall changes with the D-Bus interface will be limited
# to applications that are listed in the lockdown whitelist.
# The lockdown whitelist file is lockdown-whitelist.xml
# Default: no
Lockdown=no

# IPv6_rpfilter
# Performs a reverse path filter test on a packet for IPv6. If a reply to the
# packet would be sent via the same interface that the packet arrived on, the
# packet will match and be accepted, otherwise dropped.
# The rp_filter for IPv4 is controlled using sysctl.
# Note: This feature has a performance impact. See man page FIREWALLD.CONF(5)
```

```
# for details.
# Default: yes
IPv6_rpfilter=yes

# IndividualCalls
# Do not use combined -restore calls, but individual calls. This increases the
# time that is needed to apply changes and to start the daemon, but is good for
# debugging.
# Default: no
IndividualCalls=no

# LogDenied
# Add logging rules right before reject and drop rules in the INPUT, FORWARD
# and OUTPUT chains for the default rules and also final reject and drop rules
# in zones. Possible values are: all, unicast, broadcast, multicast and off.
# Default: off
LogDenied=off

# FirewallBackend
# Selects the firewall backend implementation.
# Choices are:
#       - nftables (default)
#       - iptables (iptables, ip6tables, ebtables and ipset)
# Note: The iptables backend is deprecated. It will be removed in a future
# release.
FirewallBackend=nftables

# FlushAllOnReload
# Flush all runtime rules on a reload. In previous releases some runtime
# configuration was retained during a reload, namely; interface to zone
# assignment, and direct rules. This was confusing to users. To get the old
# behavior set this to "no".
# Default: yes
FlushAllOnReload=yes
```

```
# RFC3964_IPv4
# As per RFC 3964, filter IPv6 traffic with 6to4 destination addresses that
# correspond to IPv4 addresses that should not be routed over the public
# internet.
# Defaults to "yes".
RFC3964_IPv4=yes
```

4.3 - La Commande firewall-cmd

firewalld s'appuie sur netfilter. Pour cette raison, l'utilisation de firewall-cmd est incompatible avec l'utilisation des commandes iptables et system-config-firewall.

> **Important** - firewall-cmd est le front-end de firewalld en ligne de commande. Il existe aussi la commande **firewall-config** qui lance un outi de configuration graphique.

Pour obtenir la liste de toutes les zones prédéfinies, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --get-zones
block dmz drop external home internal nm-shared public trusted work
```

Pour obtenir la liste de toutes les services prédéfinis, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2
bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet
bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-
collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-
registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server finger foreman foreman-proxy
freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-master
```

```
git gpsd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc ircs
iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-apiserver
kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure kube-
nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker ldap
ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns
memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-
dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd
pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link
ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp
salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap
snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui
syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-
client vdsm vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-
tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server
zerotier
```

Pour obtenir la liste de toutes les types ICMP prédéfinis, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --get-icmptypes
address-unreachable bad-header beyond-scope communication-prohibited destination-unreachable echo-reply echo-
request failed-policy fragmentation-needed host-precedence-violation host-prohibited host-redirect host-unknown
host-unreachable ip-header-bad neighbour-advertisement neighbour-solicitation network-prohibited network-redirect
network-unknown network-unreachable no-route packet-too-big parameter-problem port-unreachable precedence-cutoff
protocol-unreachable redirect reject-route required-option-missing router-advertisement router-solicitation
source-quench source-route-failed time-exceeded timestamp-reply timestamp-request tos-host-redirect tos-host-
unreachable tos-network-redirect tos-network-unreachable ttl-zero-during-reassembly ttl-zero-during-transit
unknown-header-type unknown-option
```

Pour obtenir la liste des zones de la configuration courante, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --get-active-zones
public
  interfaces: ens18
```

Pour obtenir la liste des zones de la configuration courante pour une interface spécifique, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --get-zone-of-interface=ens18
public
```

Pour obtenir la liste des services autorisés pour la zone public, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=public --list-services
cockpit dhcpv6-client ssh
```

Pour obtenir toute la configuration pour la zone public, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=public --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Pour obtenir la liste complète de toutes les zones et leurs configurations, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --list-all-zones
block
  target: %%REJECT%%
  icmp-block-inversion: no
```

```
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

dmz
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

drop
  target: DROP
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
```

```
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  forward: yes
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
```

```
  source-ports:
  icmp-blocks:
  rich rules:

internal
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

nm-shared
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcp dns ssh
  ports:
  protocols: icmp ipv6-icmp
  forward: no
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule priority="32767" reject
```

```
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

trusted
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

work
  target: default
  icmp-block-inversion: no
  interfaces:
```

```
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Pour changer la zone par défaut de public à work, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --set-default-zone=work
success

[root@redhat9 ~]# firewall-cmd --get-active-zones
work
  interfaces: ens18
```

Pour ajouter l'interface ip_fixe à la zone work, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --add-interface=ip_fixe
success

[root@redhat9 ~]# firewall-cmd --get-active-zones
work
  interfaces: ens18 ip_fixe
```

Pour supprimer l'interface ip_fixe à la zone work, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --remove-interface=ip_fixe
success
```

```
[root@redhat9 ~]# firewall-cmd --get-active-zones
work
  interfaces: ens18
```

Pour ajouter le service **http** à la zone **work**, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --add-service=http
success

[root@redhat9 ~]# firewall-cmd --zone=work --list-services
cockpit dhcpv6-client http ssh
```

Pour supprimer le service **http** de la zone **work**, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --remove-service=http
success

[root@redhat9 ~]# firewall-cmd --zone=work --list-services
cockpit dhcpv6-client ssh
```

Pour ajouter un nouveau bloc ICMP, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --add-icmp-block=echo-reply
success

[root@redhat9 ~]# firewall-cmd --zone=work --list-icmp-blocks
echo-reply
```

Pour supprimer un bloc ICMP, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --remove-icmp-block=echo-reply
success

[root@redhat9 ~]# firewall-cmd --zone=work --list-icmp-blocks
```

```
[root@redhat9 ~]#
```

Pour ajouter le port 591/tcp à la zone work, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --add-port=591/tcp
success

[root@redhat9 ~]# firewall-cmd --zone=work --list-ports
591/tcp
```

Pour supprimer le port 591/tcp à la zone work, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --remove-port=591/tcp
success

[root@redhat9 ~]# firewall-cmd --zone=work --list-ports

[root@redhat9 ~]#
```

Pour créer un nouveau service, il convient de :

- copier un fichier existant se trouvant dans le répertoire **/usr/lib/firewalld/services** vers **/etc/firewalld/services**,
- modifier le fichier,
- recharger la configuration de firewalld,
- vérifier que firewalld voit le nouveau service.

Par exemple :

```
[root@redhat9 ~]# cp /usr/lib/firewalld/services/http.xml /etc/firewalld/services/filemaker.xml

[root@redhat9 ~]# vi /etc/firewalld/services/filemaker.xml

[root@redhat9 ~]# cat /etc/firewalld/services/filemaker.xml
<?xml version="1.0" encoding="utf-8"?>
```

```
<service>
  <short>FileMakerPro</short>
  <description>fichier de service firewalld pour FileMaker Pro</description>
  <port protocol="tcp" port="591"/>
</service>

[root@redhat9 ~]# firewall-cmd --reload
success

[root@redhat9 ~]# firewall-cmd --get-services | grep filemaker
RH-Satellite-6 RH-Satellite-6-capsule afp amanda-client amanda-k5-client amqp amqps apcupsd audit ausweisapp2
bacula bacula-client bareos-director bareos-filedaemon bareos-storage bb bgp bitcoin bitcoin-rpc bitcoin-testnet
bitcoin-testnet-rpc bittorrent-lsd ceph ceph-exporter ceph-mon cfengine checkmk-agent cockpit collectd condor-
collector cratedb ctdb dds dds-multicast dds-unicast dhcp dhcpv6 dhcpv6-client distcc dns dns-over-tls docker-
registry docker-swarm dropbox-lansync elasticsearch etcd-client etcd-server filemaker finger foreman foreman-
proxy freeipa-4 freeipa-ldap freeipa-ldaps freeipa-replication freeipa-trust ftp galera ganglia-client ganglia-
master git gpsd grafana gre high-availability http http3 https ident imap imaps ipfs ipp ipp-client ipsec irc
ircs iscsi-target isns jenkins kadmin kdeconnect kerberos kibana klogin kpasswd kprop kshell kube-api kube-
apiserver kube-control-plane kube-control-plane-secure kube-controller-manager kube-controller-manager-secure
kube-nodeport-services kube-scheduler kube-scheduler-secure kube-worker kubelet kubelet-readonly kubelet-worker
ldap ldaps libvirt libvirt-tls lightning-network llmnr llmnr-client llmnr-tcp llmnr-udp managesieve matrix mdns
memcache minidlna mongodb mosh mountd mqtt mqtt-tls ms-wbt mssql murmur mysql nbd nebula netbios-ns netdata-
dashboard nfs nfs3 nmea-0183 nrpe ntp nut openvpn ovirt-imageio ovirt-storageconsole ovirt-vmconsole plex pmcd
pmproxy pmwebapi pmwebapis pop3 pop3s postgresql privoxy prometheus prometheus-node-exporter proxy-dhcp ps2link
ps3netsrv ptp pulseaudio puppetmaster quassel radius rdp redis redis-sentinel rpc-bind rquotad rsh rsyncd rtsp
salt-master samba samba-client samba-dc sane sip sips slp smtp smtp-submission smtps snmp snmptls snmptls-trap
snmptrap spideroak-lansync spotify-sync squid ssdp ssh steam-streaming svdrp svn syncthing syncthing-gui
syncthing-relay synergy syslog syslog-tls telnet tentacle tftp tile38 tinc tor-socks transmission-client upnp-
client vdsm vnc-server warpinator wbem-http wbem-https wireguard ws-discovery ws-discovery-client ws-discovery-
tcp ws-discovery-udp wsman wsmans xdmcp xmpp-bosh xmpp-client xmpp-local xmpp-server zabbix-agent zabbix-server
zerotier
```

4.4 - La Configuration Avancée de firewalld

La configuration de base de firewalld ne permet que la configuration des zones, services, blocs ICMP et les ports non-standard. Cependant firewalld peut également être configuré avec des **Rich Rules** ou ***Règles Riches***. Rich Rules ou Règles Riches évaluent des **critères** pour ensuite entreprendre une **action**.

Les **Critères** sont :

- **source address="<adresse_IP>"**
- **destination address="<adresse_IP>"**,
- **rule port port="<numéro_du_port>"**,
- **service name=<nom_d'un_sevice_prédéfini>**.

Les **Actions** sont :

- **accept**,
- **reject**,
  - une Action reject peut être associée avec un message d'erreur spécifique par la clause **type="<type_d'erreur>**,
- **drop**.

Saisissez la commande suivante pour ouvrir le port 80 :

```
[root@redhat9 ~]# firewall-cmd --add-rich-rule='rule port port="80" protocol="tcp" accept'
success
```

Important - Notez que la Rich Rule doit être entourée de caractères '.

Important - Notez que la Rich Rule a créé deux règles, une pour IPv4 et une deuxième pour IPv6. Une règle peut être créée pour IPv4 seul en incluant le Critère **family=ipv4**. De la même façon, une règle peut être

créée pour IPv6 seul en incluant le Critère **family=ipv6**.

Cette nouvelle règle est écrite en mémoire mais non pas sur disque. Pour l'écrire sur disque dans le fichier zone se trouvant dans **/etc/firewalld**, il faut ajouter l'option **–permanent** :

```
[root@redhat9 ~]# firewall-cmd --add-rich-rule='rule port port="80" protocol="tcp" accept' --permanent
success

[root@redhat9 ~]# cat /etc/firewalld/zones/work.xml
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>Work</short>
  <description>For use in work areas. You mostly trust the other computers on networks to not harm your computer.
Only selected incoming connections are accepted.</description>
  <service name="ssh"/>
  <service name="dhcpv6-client"/>
  <service name="cockpit"/>
  <rule>
    <port port="80" protocol="tcp"/>
    <accept/>
  </rule>
  <forward/>
</zone>
```

Important - Attention ! La règle ajoutée avec l'option –permanent n'est pas prise en compte imédiatement mais uniquement au prochain redémmarge. Pour qu'une règle soit appliquée immédiatement **et** être écrite sur disque, il faut saisir la commande deux fois dont une avec l'option –permanent et l'autre sans l'option –permanent.

Redémarrez le service **firewalld** :

```
[root@redhat9 ~]# systemctl restart firewalld.service
```

Pour visualiser cette règle dans la configuration de firewalld, il convient de saisir la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --zone=work --list-all
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens18
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule port port="80" protocol="tcp" accept
```

Notez que la Rich Rule est créée dans la Zone par Défaut. Il est possible de créer une Rich Rule dans une autre zone en utilisant l'option **–zone=<zone>** de la commande firewall-cmd :

```
[root@redhat9 ~]# firewall-cmd --zone=public --add-rich-rule='rule port port="80" protocol="tcp" accept'
success

[root@redhat9 ~]# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
```

```
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
        rule port port="80" protocol="tcp" accept
```

Pour supprimer une Rich Rule, il faut copier la ligne entière la concernant qui se trouve dans la sortie de la commande **firewall-cmd –list-all-zones** :

```
[root@redhat9 ~]# firewall-cmd --zone=public --remove-rich-rule='rule port port="80" protocol="tcp" accept'
success

[root@redhat9 ~]# firewall-cmd --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

4.5 - Le mode Panic de firewalld

Le mode Panic de firewalld permet de bloquer tout le trafic avec une seule commande. Pour connaître l'état du mode Panic, utilisez la commande suivante :

```
[root@redhat9 ~]# firewall-cmd --query-panic
no
```

Pour activer le mode Panic, il convient de saisir la commande suivante :

```
# firewall-cmd --panic-on
```

Pour désactiver le mode Panic, il convient de saisir la commande suivante :

```
# firewall-cmd --panic-off
```
