Selinux tutorial

Come al solito non ho trovato nulla riguardo selinux. Solo informazioni frammentarie, monche, disordinate. E soprattutto non ho trovato (quasi) nulla in italiano. Quindi devo scrivermi il documento. E lo faro' nel modo in cui mi piacerebbe leggerlo. Nessuna pretesa di completezza, non analizzero' tutti gli usi possibili. Le prova saranno fatte con centos 7, e spero di poter replicare il tutto con la mia amata slackware.

Cosa e' selinux: inutile riscrivere le cose. Wikipedia ci offre una spiegazione valida

https://it.wikipedia.org/wiki/Security-Enhanced_Linux
https://en.wikipedia.org/wiki/Security-Enhanced_Linux

Oppure se desideriamo qualcosa di piu' completo ed ufficiale , possiamo seguire questo link

https://selinuxproject.org/page/Main_Page

Riassumendo molto brevemente possiamo dire che Selinux e' un insieme di regole, queste regole si applicano ai files, ai processi, ai socket, alle schede di rete, in pratica si applicano a tutto in un sistema linux. Dette regole consentono oppure negano una determinata azione. Sia ben chiaro che selinun NON E'

a) un antivirus
b) un ampliamento del firewall
c) un tool magico che rende sicuro al 100% un host linux.

Di default in centos 7 selinux e' abilitato. Terminata la fase di install, abbiamo selinux attivo. La directory di lavoro di selinux e'

/etc/selinux

per abilitare o disbilitare selinux occorre intervenire sul file

/etc/selinux/config

l'utilizzo del file e' elementare. Ecco il file nella sua interezza

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

Dopo aver attivato o disattivato selinux occorre riavviare il sistema. Il file-system deve essere etichettato. Dopo il riavvio – necessario per rietichettare il file system - siamo pronti ad iniziare le ns prove. Attivando selinux possiamo scegliere tre impostazioni: targeted, minimum e mls. Noi analizzeremo la modalita' targeted.
Sono presenti nella centos 7 i seguenti pacchetti:

```
selinux-policy-targeted-3.13.1-229.el7_6.9.noarch
libselinux-2.5-14.1.el7.x86_64
libselinux-python-2.5-14.1.el7.x86_64
selinux-policy-3.13.1-229.el7_6.9.noarch
selinux-policy-devel-3.13.1-229.el7_6.6.noarch
libselinux-devel-2.5-14.1.el7.x86_64
selinux-policy-doc-3.13.1-229.el7_6.9.noarch
libselinux-utils-2.5-14.1.el7.x86_64
policycoreutils-2.5-29.el7.x86_64
policycoreutils-devel-2.5-29.el7.x86_64
checkpolicy-2.5-8.el7.x86_64
policycoreutils-python-2.5-29.el7.x86_64
policycoreutils-gui-2.5-29.el7.x86_64
setools-libs-3.3.8-4.el7.x86_64
setools-libs-tcl-3.3.8-4.el7.x86_64
setools-devel-3.3.8-4.el7.x86_64
setools-console-3.3.8-4.el7.x86_64
```

Per prima cosa (immaginiamo di utilizzare la centos per la prima volta, ignorando quindi come e' stata configurata) dobbiamo capire se il tool selinux e' attivo oppure e' disattivo. Il comando e'

```
[orazio@localhost ~]$ getenforce
Enforcing
```

Possiamo lanciare detto comando anche da utente normale, non root. In questo caso ci viene confermato che selinux e' attivo, e opera nel contesto Enforcing. Se vogliamo qualche informazione aggiuntiva allora possiamo utilizzare il comando

```
[orazio@localhost etc]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      28
```

Possiamo quindi iniziare a prendere confidenza col tool.
Abbiamo due utenti nel sistema linux, Root e orazio [nome di fantasia].
Eseguiamo il login via ssh come utente orazio e lanciamo ls -l , giusto per guardarci un poco intorno:

```
[orazio@localhost ~]$ ls -la
totale 32
drwx------.  5 orazio orazio 4096 11 feb 20.36 .
drwxr-xr-x. 12 root   root   4096 11 feb 20.35 ..
-rw-r--r--.  1 orazio orazio   18 20 nov  2015 .bash_logout
-rw-r--r--.  1 orazio orazio  193 20 nov  2015 .bash_profile
-rw-r--r--.  1 orazio orazio  231 20 nov  2015 .bashrc
drwxrwxr-x.  3 orazio orazio 4096 11 feb 20.36 .cache
drwxrwxr-x.  3 orazio orazio 4096 11 feb 20.36 .config
drwxr-xr-x.  4 orazio orazio 4096  7 dic 22.05 .mozilla
[orazio@localhost ~]$
```

niente di nuovo. Vediamo quello che e' presente in un sistema ove e' la prima volta che ci colleghiamo. Creiamo un file qualunque

```
touch miofile
```

e poi rilanciamo il comando ls, questa volta pero' completiamo come segue

```
[orazio@localhost ~]$ ls -laZ
drwx------. orazio orazio unconfined_u:object_r:user_home_dir_t:s0 .
drwxr-xr-x. root   root   system_u:object_r:home_root_t:s0 ..
-rw-r--r--. orazio orazio unconfined_u:object_r:user_home_t:s0 .bash_logout
-rw-r--r--. orazio orazio unconfined_u:object_r:user_home_t:s0 .bash_profile
-rw-r--r--. orazio orazio unconfined_u:object_r:user_home_t:s0 .bashrc
drwxrwxr-x. orazio orazio unconfined_u:object_r:cache_home_t:s0 .cache
drwxrwxr-x. orazio orazio unconfined_u:object_r:config_home_t:s0 .config
-rw-rw-r--. orazio orazio unconfined_u:object_r:user_home_t:s0 miofile
drwxr-xr-x. orazio orazio unconfined_u:object_r:mozilla_home_t:s0 .mozilla
[orazio@localhost ~]$
```

le cose cominciano a complicarsi. Oltre ai permessi soliti del mondo linux, abbiamo quattro colonne sconosciute. I valori di queste quattro colonne determinano il comportamento di selinux. In pratica selinux aggiunge dei controlli in piu' rispetto ai solito controlli linux. Quando il sistema operativo linux obbedendo ai permessi tradizionali assegnati impedisca un'azione , ecco che selinux non interviene. L'azione e' stata negata dai permessi tradizionali (DAC) di linux. Se i permessi DAC concedono l'azione, ecco che entrano in gioco i controlli selinux (MAC) , i quali a loro volta possono concedere o rifiutare l'azione. Analizziamo il file miofile appena creato:

```
-rw-rw-r--. orazio orazio unconfined_u:object_r:user_home_t:s0 miofile
```

unconfined_u sta per "utente non confinato", object_r sta per "ruolo oggetto", user_home_t sta per "tipo utente". s0 si riferisce alla modalita' mls , che noi non analizziamo. Ora rechiamoci /etc e lanciamo il comando

```
[orazio@localhost etc]$ ls -lZa fstab
-rw-r--r--. root root system_u:object_r:etc_t:s0       fstab
```

notiamo che i tre campi inerenti la politica di selinux son diversi. Qui abbiamo system_u , object_r e etc_t . Selinux adegua il suo comportamento analizzando questi campi. Tutti i processi e file sono contrassegnati con un tipo. Un tipo definisce un dominio per i processi e un tipo per i file. Ciascun processo è separato dagli altri, viene eseguito nel proprio dominio, e le regole della politica di SELinux stabiliscono come i processi devono interagire tra loro e con i file. L'accesso è garantito solo se esiste una regola SELinux che lo permetta specificatamente. In linea di massima gli utenti SELinux sono autorizzati per i ruoli, i ruoli sono autorizzati per i domini ed i processi vengono eseguiti nei propri domini separati.
Sembra complicato e lo e'. Almeno per me. Un'altra bella sorpresa ci attende lanciano il comando

```
id
```

```
uid=1009(orazio) gid=1010(orazio) gruppi=1010(orazio) \
contesto=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

l’eseguibile “ id “ in centos e’ stato modificato per tenere conto anche del contesto selinux. Il sistema di dice che siamo utenti “ non confinati “ , che operiamo nel ruolo “ non confinato “ e che siamo di tipo “ non confinato” .
Diamo ora uno sguardo a come selinux amministra le proprie utenze (che non sono le utenze fisiche della linux-box). Occorre lanciare il comando come root.

```
[root@localhost ~]# semanage user -l

Etichettatura MLS/       MLS/
Utente di SELinux Prefisso   Livello MCS Range MCS              Ruoli SELinux

guest_u         user       s0         s0                        guest_r
root            user       s0         s0-s0:c0.c1023            staff_r sysadm_r system_r unconfined_r
staff_u         user       s0         s0-s0:c0.c1023            staff_r sysadm_r system_r unconfined_r
sysadm_u        user       s0         s0-s0:c0.c1023            sysadm_r
system_u        user       s0         s0-s0:c0.c1023            system_r unconfined_r
unconfined_u    user       s0         s0-s0:c0.c1023            system_r unconfined_r
user_u          user       s0         s0                        user_r
xguest_u        user       s0         s0                        xguest_r
```

Quindi il tool selinux mappa un’utenza fisica della linux-box ad un’utenza di selinux. Nel nostro caso specifico, il linux user “ orazio “ viene mappato all’utenza di selinux “ unconfined_u “. Ce ne sinceriamo facendo login a nome di orazio e lanciando il solido “ id “ .

```
[orazio@localhost ~]$ id
uid=1001(orazio) gid=1001(orazio) gruppi=1001(orazio) contesto=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Quindi i processi lanciati dall’user orazio saranno di tipo unconfined_t , e potranno accedere agli altri processi presenti nel dominio unconfined_t. Potranno accedere a processi eseguiti in altri domini solo se esiste una regola di selinux che permette tale accesso. In mancanza di tale regola, ecco che il transito tra domini viene negato.
Lanciamo ora (sempre come root) il comando

```
[root@localhost ~]# semanage login -l

Nome di login        Utente di SELinux    Range MLS/MCS        Servizio

__default__          unconfined_u         s0-s0:c0.c1023       *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
[root@localhost ~]#
```

Attualmente abbiamo tre mappature attive: l’utente linux-box root viene mappato come utente selinux unconfined_u , gli utenti di sistema vengono mappati come system_u , e gli utenti di default vengono mappati anche loro come unconfined_u .
Gli utenti Linux sono mappati all'accesso _default_ di SELinux per impostazione predefinita, che viene mappato all'utente SELinux unconfined_u . Tuttavia, SELinux può limitare gli utenti Linux, per sfruttare le regole di sicurezza e i meccanismi applicati a loro, mappando gli utenti Linux agli utenti SELinux. Un certo numero di utenti SELinux confinati esiste nella politica SELinux. Di seguito è riportato un elenco degli utenti SELinux confinati e dei relativi domini associati:

```
guest_u : il dominio per l'utente è guest_t.
staff_u : il dominio per l'utente è staff_t.
user_u : il dominio per l'utente è user_t.
xguest_x : il dominio per l'utente è xguest_t.
```

- Gli utenti Linux nei domini guest_t , xguest_t e user_t possono eseguire applicazioni set user ID (setuid) solo se il criterio SELinux lo consente (come passwd). Non possono eseguire le applicazioni su e sudo setuid per diventare l'utente root.

- Gli utenti Linux nel dominio guest_t non hanno accesso alla rete e possono accedere solo da un terminale. Possono accedere con ssh ma non possono usare ssh per connettersi a un altro sistema. L'unico accesso alla rete che gli utenti Linux nel dominio xguest_t hanno è Firefox per la connessione alle pagine web.

- Gli utenti Linux nei domini xguest_t, user_t e staff_t possono accedere utilizzando il sistema X Window e un terminale.

- Per impostazione predefinita, gli utenti Linux nel dominio staff_t non dispongono delle autorizzazioni per eseguire applicazioni con il comando sudo.

- Per impostazione predefinita, gli utenti Linux nei domini guest_t e xguest_t non possono eseguire applicazioni nelle loro directory home o / tmp, impedendo loro di eseguire applicazioni nelle directory a cui hanno accesso in scrittura. Ciò consente di impedire alle applicazioni imperfette o malevoli di modificare i file di cui sono proprietari.

- Per impostazione predefinita, gli utenti Linux nei domini user_t e staff_t possono eseguire

applicazioni nelle loro directory home e / tmp

Un utilizzo interessante riguardo quanto appena esposto lo vediamo in questo link:

https://docs.fedoraproject.org/it-IT/Fedora/13/html/Security-Enhanced_Linux/sect-Security-Enhanced_Linux-Confining_Users-xguest_Kiosk_Mode.html

Alla luce di quanto visto, se volessimo limitare l'utente orazio (cioe' togliere un poco di liberta' di azione) potremmo utilizzare il seguente comando

```
[root@localhost ~]# semanage login -a -s staff_u orazio
```

Cosi' facendo istruiamo selinux a trattare l'utente della linux-box " orazio " come un utente " staff_t " di selinux, con tutte le limitazioni del caso. Infatti ora l'utente orazio non puo' utilizzare il comando " su " , come vediamo

```
[orazio@localhost ~]$ su
Password:
su: Autenticazione fallita
[orazio@localhost ~]$
```

Possiamo controllare nel file /var/log/messages il log:

```
Apr 24 21:26:51 localhost dbus-daemon: dbus[1851]: [system] Successfully activated service
'net.reactivated.Fprint'
Apr 24 21:26:51 localhost systemd: Started Fingerprint Authentication Daemon.
Apr 24 21:26:51 localhost fprintd: Launching FprintObject
Apr 24 21:26:51 localhost fprintd: ** Message: D-Bus service launched with name:
net.reactivated.Fprint
Apr 24 21:26:51 localhost fprintd: ** Message: entering main loop
Apr 24 21:27:09 localhost su: FAILED SU (to root) orazio on pts/1
Apr 24 21:27:09 localhost dbus[1851]: [system] Activating service
name='org.fedoraproject.Setroubleshootd' (using servicehelper)
Apr 24 21:27:09 localhost dbus-daemon: dbus[1851]: [system] Activating service
name='org.fedoraproject.Setroubleshootd' (using servicehelper)
Apr 24 21:27:09 localhost dbus[1851]: [system] Successfully activated service
'org.fedoraproject.Setroubleshootd'
Apr 24 21:27:09 localhost dbus-daemon: dbus[1851]: [system] Successfully activated service
'org.fedoraproject.Setroubleshootd'
Apr 24 21:27:10 localhost setroubleshoot: Plugin Exception restorecon_source
Apr 24 21:27:10 localhost python: SELinux is preventing /usr/bin/su from write access on the
file /var/log/btmp.#012#012*****  Plugin catchall (100. confidence) suggests
**************************#012#012If you believe that su should be allowed write access on
the btmp file by default.#012Then you should report this as a bug.#012You can generate a
local policy module to allow this access.#012Do#012allow this access for now by
executing:#012# grep su /var/log/audit/audit.log | audit2allow -M mypol#012# semodule -i
mypol.pp#012
Apr 24 21:27:10 localhost dbus-daemon: 'list' object has no attribute 'split'
Apr 24 21:27:10 localhost setroubleshoot: SELinux is preventing /usr/bin/su from write access
on the file /var/log/btmp. For complete SELinux messages. run sealert -l c1f177c0-afb9-44c3-
91ec-d5f2cf21a976
Apr 24 21:27:21 localhost fprintd: ** Message: No devices in use, exit
```

Ci viene spiegato cosa succede, e ci viene suggerito come ottenere una maggior quantita' di informazioni, tramite il comando

```
sealert -l c1f177c0-afb9-44c3-91ec-d5f2cf21a976
```

il quale a sua volta restituisce

```
[root@localhost log]# sealert -l c1f177c0-afb9-44c3-91ec-d5f2cf21a976
SELinux is preventing /usr/bin/su from write access on the file /var/log/btmp.

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

Se si crede che su dovrebbe avere possibilità di accesso write sui btmp file in modo
predefinito, Allora si dovrebbe riportare il problema come bug.
E' possibile generare un modulo di politica locale per consentire questo accesso.
Do
consentire questo accesso per il momento eseguendo:
# grep su /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp


Additional Information:
Source Context                staff_u:staff_r:staff_t:s0-s0:c0.c1023
Target Context                system_u:object_r:faillog_t:s0
Target Objects                /var/log/btmp [ file ]
Source                        su
Source Path                   /usr/bin/su
```

```
Port                          <Unknown>
Host                          localhost.localdomain
Source RPM Packages           util-linux-2.23.2-26.el7.x86_64
Target RPM Packages           initscripts-9.49.30-1.el7.x86_64
                              systemd-219-62.el7.x86_64
Policy RPM                    selinux-policy-3.13.1-229.el7_6.9.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     localhost.localdomain
Platform                      Linux localhost.localdomain 3.10.0-327.el7.x86_64
                              #1 SMP Thu Nov 19 22:10:57 UTC 2015 x86_64 x86_64
Alert Count                   2
First Seen                    2019-04-24 21:18:53 CEST
Last Seen                     2019-04-24 21:27:09 CEST
Local ID                      c1f177c0-afb9-44c3-91ec-d5f2cf21a976

Raw Audit Messages
type=AVC msg=audit(1556134029.36:260): avc:  denied  { write } for  pid=5587 comm="su"
name="btmp" dev="dm-1" ino=293666 scontext=staff_u:staff_r:staff_t:s0-s0:c0.c1023
tcontext=system_u:object_r:faillog_t:s0 tclass=file


type=SYSCALL msg=audit(1556134029.36:260): arch=x86_64 syscall=open success=no exit=EACCES
a0=7f38ac501030 a1=1 a2=15d3 a3=5cc0b88d items=0 ppid=5226 pid=5587 auid=1001 uid=1001
gid=1001 euid=0 suid=0 fsuid=0 egid=1001 sgid=1001 fsgid=1001 tty=pts1 ses=11 comm=su
exe=/usr/bin/su subj=staff_u:staff_r:staff_t:s0-s0:c0.c1023 key=(null)

Hash: su,staff_t,faillog_t,file,write

[root@localhost log]#
```

Ora come esercizio creeremo un nuovo utente selinux e lo chiameremo prova_u , autorizzeremo prova_u ad operare all'interno del dominio sysadm_r , e poi creeremo l'utente linux pinguino, mappandolo all'utente selinux prova_u.

```
[root@localhost ~]# semanage user -a -R "sysadm_r" prova_u
[root@localhost ~]# useradd pinguino
[root@localhost ~]# semanage login -a -s prova_u pinguino
[root@localhost ~]# semanage login -l

Nome di login        Utente di SELinux    Range MLS/MCS        Servizio

__default__          unconfined_u         s0-s0:c0.c1023       *
pinguino             prova_u              s0                   *
root                 unconfined_u         s0-s0:c0.c1023       *
system_u             system_u             s0-s0:c0.c1023       *
```

Booleane:

Le booleane sono semplici direttive che possono assume solamente due valori. Zero oppure uno. Se sono impostate a zero allora impediscono una determinata azione, se sono impostare a uno allora concedono una determinata azione. Vengono utilizzate per modificare il comportamento di selinux senza dover riavviare il sistema e soprattutto senza compilare regola nuove. Possiamo avere un elenco delle booleane disponibili col semplice comando:

```
[root@localhost ~]# 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
condor_tcp_network_connect --> off
```

Cerchiamo ora qualcosa di piu' immediato:

```
[root@localhost ~]# getsebool -a | grep -i ssh
fenced_can_ssh --> off
selinuxuser_use_ssh_chroot --> off
ssh_chroot_rw_homedirs --> off
ssh_keysign --> off
ssh_sysadm_login --> off
```

Il significato e' piuttosto intuitivo. Abbiamo ottenuto l'elenco delle direttive il cui comportamento influenza le connessioni ssh. Possiamo aumentare la quantita' di informazioni utilizzando un altro comando:

```
[root@localhost ~]# semanage boolean -l | more
Booleana di SELinux           Stato  Predefinito Descrizione

privoxy_connect_any           (on   ,   on)  Determinare se privoxy può connettersi a tutte le porte tcp.
smartmon_3ware                (off  ,  off)  Determinare se smartmon può supportare i dispositivi nei controller 3ware.
mpd_enable_homedirs           (off  ,  off)  Determinare se mpd può attraversare le cartelle home utenti.
xdm_sysadm_login              (off  ,  off)  Permettere al programma per il login grafico di effettuare il login
                                              direttamente come sysadm_r:sysadm_t
xen_use_nfs                   (off  ,  off)  Permettere a xen di gestire i file nfs
mozilla_read_content          (off  ,  off)  Permettere ai browser web confinati la lettura del contenuto delle cartelle
                                              home
ssh_chroot_rw_homedirs        (off  ,  off)  Permettere a ssh con env chroot di leggere e scrivere file nelle cartelle home
mount_anyfile                 (on   ,   on)  Permettere  ai comandi mount il montaggio di una qualsiasi cartella o file.
cron_userdomain_transition    (on   ,   on)  Determinare se crond può eseguire i lavori nel dominio utente al posto del
                                              dominio cronjob generico.
xdm_write_home                (off  ,  off)  Permettere al programma del login grafico di creare file nelle cartelle HOME
                                              come xdm_home_t.
openvpn_can_network_connect   (on   ,   on)  Determinare se openvpn può connettersi alla rete TCP.
```

Come vediamo, al server openvpn e' consentito l'accesso alla rete tcp.
Per modificare il comportamento di selinux operando sulle booleane, e' necessario utilizzare il comando setsebol. Vediamo un esempio concreto: cominciamo a elencare le booleane che hanno effetto sul server ftp, col comando

```
[root@localhost ~]# getsebool -a | grep -i ftp
ftpd_anon_write --> off
ftpd_connect_all_unreserved --> off
ftpd_connect_db --> off
ftpd_full_access --> off
ftpd_use_cifs --> off
ftpd_use_fusefs --> off
ftpd_use_nfs --> off
ftpd_use_passive_mode --> off
httpd_can_connect_ftp --> off
httpd_enable_ftp_server --> off
tftp_anon_write --> off
tftp_home_dir --> off
```

ftpd_connect_db e' impostata a off (zero), quindi selinux proibisce al server ftp il collegamento a qualsiasi database. Volendo concedere tale dialogo dobbiamo impartire la direttiva

```
setsebool ftpd_connect_db 1
```

controlliamo per scrupolo

```
[root@localhost ~]# getsebool -a | grep -i ftp
ftpd_anon_write --> off
ftpd_connect_all_unreserved --> off
ftpd_connect_db --> on
ftpd_full_access --> off
ftpd_use_cifs --> off
ftpd_use_fusefs --> off
ftpd_use_nfs --> off
ftpd_use_passive_mode --> off
httpd_can_connect_ftp --> off
httpd_enable_ftp_server --> off
tftp_anon_write --> off
tftp_home_dir --> off
```

ora il server ftp ha accesso ai database. Volendo quindi possiamo creare un server ftp con utenze non di sistema (linux-box) , ma con utenze fittizie , i cui nomi di login e relative pass sono archiviate in un database. Un altro esempio, questa volta riferito al server samba:

```
[root@localhost ~]# getsebool -a | grep -i samba
samba_create_home_dirs --> off
samba_domain_controller --> off
samba_enable_home_dirs --> off
```

```
samba_export_all_ro --> off
samba_export_all_rw --> off
samba_load_libgfapi --> off
samba_portmapper --> off
samba_run_unconfined --> off
samba_share_fusefs --> off
samba_share_nfs --> off
sanlock_use_samba --> off
tmpreaper_use_samba --> off
use_samba_home_dirs --> off
virt_use_samba --> off
```

Con queste configurazioni, il server samba impedisce l'accesso alle "home dirs" . Per consentire tale accesso occorre impartire la seguente direttiva:

```
[root@localhost ~]# setsebool use_samba_home_dirs on
```

controlliamo per scrupolo che il sistema abbia recepito la direttiva

```
[root@localhost ~]# getsebool -a | grep -i samba
samba_create_home_dirs --> off
samba_domain_controller --> off
samba_enable_home_dirs --> off
samba_export_all_ro --> off
samba_export_all_rw --> off
samba_load_libgfapi --> off
samba_portmapper --> off
samba_run_unconfined --> off
samba_share_fusefs --> off
samba_share_nfs --> off
sanlock_use_samba --> off
tmpreaper_use_samba --> off
use_samba_home_dirs --> on
virt_use_samba --> off
```

La direttiva e' stata applicata. Per approfondimenti consultare https://docs.fedoraproject.org/it-IT/Fedora/13/html/Security-Enhanced_Linux/sect-Security-Enhanced_Linux-Working_with_SELinux-Booleans.html.

esempio samba share

Vediamo ora un esempio semplice che coinvolge samba. Installiamo i pacchetti samba necessari a condividere una directory in rete, quindi editiamo il file /etc/samba/smb.conf , nella maniera seguente:

```
[global]
        workgroup = GRUPPO
        security = user
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw
        guest account = orazio
        guest ok = yes

[homes]
        comment = Home Directories
        valid users = %S, %D%w%S
        browseable = No
        read only = No
        inherit acls = Yes
```

Abbiamo soltanto apportato poche modifiche rispetto al file smb.conf di esempio, presente dopo l'installazione dei pacchetti samba. L'utente della linux-box orazio deve avere una pass samba impostata tramite il comando smbpassw . Controlliamo la correttezza del file tramite il comando

```
[root@localhost system]# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[homes]"
Processing section "[printers]"
Processing section "[print$]"
Loaded services file OK.
Server role: ROLE_STANDALONE
```

Avviamo i servizi samba – se gia' non sono avviati – e proviamo ad accedere alle condivisioni linux

a partire da un pc windows qualunque collegato in rete (non abbiamo dominio windows attivo!). Otteniamo una prima connessione che ci mostra la cartella " orazio " , la quale altro non e' che la home directory dell'user fisico orazio, nella nostra linux-box. Tuttavia non riusciamo ad accedere oltre, poiche' selinux interviene , aggiungendo i suoi controlli a quelli del sistema operativo linux. Infatti se impartiamo il comando

```
[root@localhost ~]# setsebool samba_enable_home_dirs 1
```

ecco che l'accesso alle home directory (via samba) diventa possibile.

Selinux con webmin

Non vi e' molto da dire. Scaricato il pacchetto webmin.tar.gz dall'omonimo sito, espanso il pacchetto e lanciato il setup.sh , ecco che webmin si e' installato senza colpo ferire. L'accesso via web all'applicativo webmin e' risultato efficiente fin dal primo momento, anche con selinux attivato. Le policy di default quindi tengono conto di webmin.

Selinux con owncloud

Il noto tool owncloud e' molto ben documentato. Anche sotto l'aspetto selinux. Ecco un estratto della documentazione ufficiale del sito https://doc.owncloud.com/server/admin_manual/installation/selinux_configuration.html#introduction

Preparation

When you have SELinux enabled on your Linux distribution, you may run into permissions problems after a new ownCloud installation, and see permission denied errors in your ownCloud logs.

The following settings should work for most SELinux systems that use the default distro profiles. Run these commands as root, and remember to adjust the filepaths in these examples for your installation

```
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud/data(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud/config(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud/apps(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud/apps-external(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud/.htaccess'
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud/.user.ini'

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

If you uninstall ownCloud you need to remove the ownCloud directory labels. To do this execute the following commands as root after uninstalling ownCloud

```
semanage fcontext -d '/var/www/html/owncloud/data(/.*)?'
semanage fcontext -d '/var/www/html/owncloud/config(/.*)?'
semanage fcontext -d '/var/www/html/owncloud/apps(/.*)?'
semanage fcontext -d '/var/www/html/owncloud/apps-external(/.*)?'
semanage fcontext -d '/var/www/html/owncloud/.htaccess'
semanage fcontext -d '/var/www/html/owncloud/.user.ini'

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

If you have customized SELinux policies and these examples do not work, you must give the HTTP server write access to these directories:

```
/var/www/html/owncloud/data
/var/www/html/owncloud/config
/var/www/html/owncloud/apps
/var/www/html/owncloud/apps-external
```

Enable updates via the web interface:

To enable updates via the ownCloud web interface, you may need this to enable writing to the ownCloud directories:

```
setsebool httpd_unified on
```

When the update is completed, disable write access:

```
setsebool -P  httpd_unified  off
```

Disallow write access to the whole web directory For security reasons it’s suggested to disable write access to all folders in /var/www/ (default):

```
setsebool -P httpd_unified off
```

Allow access to a remote database
An additional setting is needed if your installation is connecting to a remote database:

```
setsebool -P httpd_can_network_connect_db on
```

Allow access to LDAP server. Use this setting to allow LDAP connections:

```
setsebool -P httpd_can_connect_ldap on
```

Allow access to remote network. ownCloud requires access to remote networks for functions such as Server-to-Server sharing, external storages or the ownCloud Marketplace. To allow this access use the following setting:

```
setsebool -P httpd_can_network_connect on
```

Allow access to network memcache This setting is not required if httpd_can_network_connect is already on:

```
setsebool -P httpd_can_network_memcache on
```

Allow access to SMTP/sendmail If you want to allow ownCloud to send out e-mail notifications via sendmail you need to use the following setting:

```
setsebool -P httpd_can_sendmail on
```

Allow access to CIFS/SMB. If you have placed your datadir on a CIFS/SMB share use the following setting:

```
setsebool -P httpd_use_cifs on
```

Allow access to FuseFS. If your owncloud data folder resides on a Fuse Filesystem (e.g. EncFS etc), this setting is required as well:

```
setsebool -P httpd_use_fusefs on
```

Allow access to GPG for Rainloop. If you use a the rainloop webmail client app which supports GPG/PGP, you might need this:

```
setsebool -P httpd_use_gpg on
```

Troubleshooting. General Troubleshooting For general Troubleshooting of SELinux and its profiles try to install the package setroubleshoot and run:

```
sealert -a /var/log/audit/audit.log > /path/to/mylogfile.txt
```

to get a report which helps you configuring your SELinux profiles. Another tool for troubleshooting is to enable a single ruleset for your ownCloud directory:

```
semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/owncloud(/.*)?'
restorecon -RF /var/www/html/owncloud
```

It is much stronger security to have a more fine-grained ruleset as in the examples at the beginning, so use this only for testing and troubleshooting. It has a similar effect to disabling SELinux, so don’t use it on production systems.
See this discussion on GitHub to learn more about configuring SELinux correctly for ownCloud.
Redis on RHEL 7 & Derivatives. On RHEL 7 and its derivatives, if you are using Redis for both local server cache and file locking and Redis is configured to listen on a Unix socket instead of a TCP/IP port (which is recommended if Redis is running on the same system as ownCloud) you must instruct SELinux to allow daemons to enable cluster mode. You can do this using the following command:

```
setsebool -P daemons_enable_cluster_mode 1
```

Elenco delle direttive

apol (1)

apol è uno strumento grafico che consente all'utente di esaminare aspetti di una politica SELinux. Lo strumento consente all'utente di consultare i componenti della politica (tipi, classi, ruoli, utenti, ecc.), regole (TE, RBAC, MLS) e contesti del file system. Lo strumento fornisce anche analisi approfondite delle transizioni di dominio, flussi di informazioni e permessi di rietichettatura.

audit2allow (1)

genera politiche di selinux a partire dei log. Dalla pagina man:
cat /var/log/audit/audit.log | audit2allow -m local > local.te . Abbiamo cosi' ottenuto la policy locale local.te
altro esempio: audit2allow -w -a . con questo comando otteniamo in formato leggibile l'analisi dei divieti generati da Selinux, e registrati nel file /var/log/audit/audit.log

audit2why (1)

traduce i messaggi di selinux ed aggiunge info riguardo il motivo del divieto.

avcstat (8)

mostra le statistiche di avcstat.
Questo fornisce un breve output delle statistiche della cache del vettore di accesso sin dal boot. Puoi guardare le statistiche in tempo reale specificando un intervallo di tempo in secondi. Questo fornisce statistiche aggiornate dall'output iniziale. Il file delle statistiche utilizzato è / selinux / avc / cache_stats, ed è possibile specificare un diverso file di cache con -f / percorso / su / file. Ad esempio, questo potrebbe essere utile per rivedere le istantanee salvate di / selinux / avc / cache_stats.
Fonte: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/SELinux_Guide/rhlcommon-chapter-0007.html

chcat (8)

modifica la categoria di sicurezza. Vedi:
https://selinuxproject.org/page/MultiCategorySecurity

chcon (1)

cambia (modifica) il contesto di sicurezza
es: creiamo il file zzz con " touch zzz " , poi diamo il comando ls -Z zzz , otteniamo
-rw-r--r--. root root unconfined_u:object_r:admin_home_t:s0 zzz .
Ora cambiamo il contesto "_t" col comando " chcon -t samba_log_t zzz " . Controlliamo con ls -lZ
e vediamo che
-rw-r--r--. root root unconfined_u:object_r:samba_log_t:s0 zzz
se utilizziamo l'opzione " -v " abbiamo anche una maggiore verbosita'.
Per ulteriori esempi/ragguagli consultare la pagina man
man chcon

checkmodule (8)

compilatore del modulo della politica di selinux . Serve per aggiungere una policy, a partire dai sorgenti.

checkpolicy (8)

compilatore della politica di selinux.

findcon (1)

strumento di ricerca files di un determinato contesto selinux. findcon consente all'utente di cercare file con un contesto specificato. Es:
findcon -t samba_log_t /var/log/ , restituisce
/var/log/samba -d system_u:object_r:samba_log_t:s0
/var/log/samba/old -d system_u:object_r:samba_log_t:s0
L'opzione -R consente l'utilizzo delle espressioni regolari

Fixfiles (8)

correggere i contesti di sicurezza SELinux.
Vedi anche: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/SELinux_Guide/rhlcommon-chapter-0017.html

getenforce (8)

mostra l'attuale modo di lavoro di selinux (permissivo o enforcing)

newrole

Esegui una nuova shell in un nuovo contesto. Il nuovo contesto è derivato dal vecchio contesto in cui newrole è originariamente eseguito. Se viene specificata l'opzione -r o --role, il nuovo contesto avrà il ruolo specificato da ROLE. Se viene specificata l'opzione -t o --type, il nuovo contesto avrà il tipo (dominio) specificato da TYPE. Se viene specificato un ruolo, ma non viene specificato alcun tipo, il tipo predefinito viene derivato dal ruolo specificato. Se viene specificata l'opzione -l o --level, il nuovo contesto avrà il livello di sensibilità specificato da LEVEL. Se LEVEL è un intervallo, il nuovo contesto avrà il livello di sensibilità e l'autorizzazione specificati da tale intervallo. Se viene specificata l'opzione -p o --preserve-environment, la shell con il nuovo contesto SELinux conserverà le variabili di ambiente, altrimenti verrà creato un nuovo ambiente minimo. Argomenti aggiuntivi ARGS può essere fornito dopo un'opzione-, nel qual caso vengono forniti alla nuova shell. In particolare, un argomento di - -c farà in modo che l'argomento successivo venga trattato come un comando dalla maggior parte degli interpreti di comando. Se un argomento di comando viene specificato su newrole e il nome del comando viene trovato in /etc/selinux/newrole_pam.conf, verrà utilizzato il nome del servizio pam elencato in quel file per il comando anziché la normale configurazione newrole pam. Ciò consente la configurazione pam per comando quando invocato tramite newrole, ad es. saltare la fase di ri-autenticazione interattiva. La nuova shell sarà la shell specificata nella voce dell'utente nel file / etc / passwd.
-V o --version mostra la versione corrente di newrole

runcon

eseguire il comando con il contesto di sicurezza SELinux specificato. Esempio:
runcon -t initrc_t -r system_r -u user_u IL_TUO_COMANDO

seaudit (8)

strumento grafico di analise dei log di selinux

sealert

si tratta dell'interfaccia GUI di setroubleshoot

sechecker (1)

strumento di controllo delle politiche di selinux. Sechecker -l elenca i moduli ed i profili disponibili.
Es:

```
       [root@localhost ~]# sechecker -m inc_mount
Using policy: /sys/fs/selinux/policy
Using file contexts: /etc/selinux/targeted/contexts/files/file_contexts

Module name: inc_mount  Severity: med
--------------------------------------------------------------------------------
This module finds domains that have incomplete mount permissions.
In order for a mount operation to be allowed by the policy the following rules
must be present:

   1) allow somedomain_d sometype_t : filesystem  { mount };
   2) allow somedomain_d sometype_t : dir { mounton };

This module finds domains that have only one of the rules listed above.

Found 69 types.
```

```
glusterd_t, rhgb_t, automount_t, mount_t
virtd_lxc_t, insmod_t, mock_t, svirt_qemu_net_t
podsleuth_t, snapperd_t, ifconfig_t, afs_t
rpm_script_t, initrc_t, sysadm_t, init_t
rasdaemon_t, xdm_t, smbmount_t, svirt_kvm_net_t
systemd_logind_t, namespace_init_t, container_runtime_t, devicekit_disk_t
kernel_t, pegasus_t, setfiles_t, newrole_t
su_domain_type, xguest_t, polydomain, crond_t
kern_unconfined, ssh_t, mock_build_t, sshd_t
pulseaudio_t, svirt_sandbox_domain, sysadm_screen_t, pegasus_openlmi_account_t
sssd_t, nfsd_t, xguest_usertype, staff_usertype
staff_screen_t, seunshare_domain, rshd_t, user_usertype
pegasus_openlmi_logicalfile_t, rhev_agentd_consolehelper_t, udev_t, auditadm_screen_t
user_wine_t, useradd_t, neutron_t, sysadm_usertype
realmd_consolehelper_t, staff_wine_t, user_screen_t, staff_consolehelper_t
guest_t, brltty_t, unconfined_t, guest_usertype
secadm_screen_t, ricci_modstorage_t, user_t, staff_t
secadm_t
```

secon (1)

mostra il livello di contesto, a partire da un utente, un file, un processo. Es:

```
[root@localhost ~]# secon -p 1
user: system_u
role: system_r
type: init_t
sensitivity: s0
clearance: s0
mls-range: s0
```

ci ha mostrato il contesto selinux dell processo con pid = 1 , cioe' init.

```
[root@localhost ~]# secon -f zzz
user: unconfined_u
role: object_r
type: samba_log_t
sensitivity: s0
clearance: s0
mls-range: s0
```

ci ha mostrato il contesto selinux dell'ipotetico file zzz

Sediff

consente ad un utente di ispezionare le differenza tra due politiche di selinux

sedispatch

Esegue la scansione dei messaggi di controllo per i messaggi SELinux AVC, li formatta in un messaggio dbus e li invia a setroubleshootd

seinfo

consente all'utente di interrogare i componenti di una politica SELinux. Es:

```
[root@localhost ~]# seinfo -u

Users: 8
   sysadm_u
   system_u
   xguest_u
   root
   guest_u
   staff_u
   user_u
   unconfined_u
```

restituisce le utenze valide di selinux [in realta' e' molto piu' verboso ed utile utilizzare il comando semanage user -l]

```
[root@localhost ~]# seinfo -t | more

Types: 4778
   bluetooth_conf_t
   cmirrord_exec_t
   colord_exec_t
   container_auth_t
   foghorn_exec_t
   jacorb_port_t
   pki_ra_exec_t
   pki_ra_lock_t
   sosreport_t
   squid_script_exec_t
   etc_runtime_t
   fenced_tmp_t
   git_session_t
   glance_port_t
   osad_log_t
   presence_port_t
   samba_secrets_t
   snort_exec_t
   sshd_sandbox_t
   audisp_var_run_t
   auditd_var_run_t
   blktap_var_run_t
   cfengine_execd_t
   cinder_var_lib_t
   cinder_var_run_t
   colord_var_lib_t
   comsat_var_run_t
   condor_var_lib_t
   condor_var_run_t
   conman_var_run_t
   dbskkd_var_run_t
   dccifd_var_run_t
   dirsrv_var_lib_t
   dirsrv_var_run_t
   fenced_var_run_t
   gdomap_var_run_t
   git_script_exec_t
   groupd_var_run_t
   hsqldb_var_lib_t
   […..]
```

ci mostra la lista dei “ tipi “ disponibili in pratica l’elenco dei “ _t “ disponibili sul sistema.

Selinuxconlist

elenca tutto il contesto SELinux raggiungibile per l'utente

selinuxdefcon

descrive il contesto SELinux predefinito per l'utente

selinuxenabled

strumento da utilizzare all'interno degli script della shell per determinare se selinux è abilitato

selinuxexeccon

mostra il contesto di selinux assegnato ad un determinato file. Es:

```
[root@localhost ~]# selinuxexeccon zzz
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

mostra il contesto di sicurezza dell’ipotetico file zzz

selinux-polgengui

strumento grafico di generazione policy di selinux.

selinux_restorecon

ripristina il contesto di sicurezza di default dei files

semanage

strumento principale per l’amministrazione di selinux. Molto complesso e pieno di dettagli. [vedere spiegone dettagliato]

Semanage-boolean

strumento per l’amministazione delle “booleane” di selinux

semodule

semodule è lo strumento utilizzato per gestire i moduli della politica di SELinux, tra cui l'installazione, l'aggiornamento, l'elenco e la rimozione dei moduli. il semodule può anche essere utilizzato per forzare una ricostruzione della politica dall'archivio dei moduli e / o per forzare una ricarica della politica senza eseguire altre transazioni. semodule agisce su pacchetti di moduli creati da semodule_package. Convenzionalmente, questi file hanno un suffisso .pp (pacchetto della politica) sebbene questo non sia obbligatorio in alcun modo.

semodule_deps

semodule_deps è uno strumento di sviluppo per mostrare le dipendenze tra i pacchetti di politiche. Per ogni modulo stampa un elenco di sotto_moduli che devono essere presenti per soddisfare i requisiti di un modulo. Si occupa solo dei requisiti, non delle dipendenze opzionali.
Affinché semodule_deps fornisca informazioni utili, l'elenco dei pacchetti passati non può avere dipendenze insoddisfatte. In generale ciò significa che l'elenco dei moduli sarà in genere piuttosto lungo. Di default le opzioni del modulo di base sono escluse poiché quasi ogni modulo ha questa dipendenza. L'opzione -b includerà queste dipendenze. Oltre all'output leggibile dall'uomo, semodule_deps può generare le dipendenze nel formato punto Graphviz (http://www.graphviz.org/) usando l'opzione -g. Questo è utile per produrre un'immagine delle dipendenze.

semodule_expand

semodule_expand è uno strumento di sviluppo per l'espansione manuale di un pacchetto del modulo della politica di base in un file di criteri binari del kernel. Questo strumento non è necessario per il normale funzionamento di SELinux. Nel normale funzionamento, tale espansione viene eseguita internamente da libsemanage in risposta ai comandi semodule. I pacchetti del modulo della politica di base possono essere creati direttamente da semodule_package o semodule_link (quando si collega un insieme di pacchetti in un unico pacchetto).

semodule_link

semodule_link è uno strumento per sviluppatori che consente di collegare manualmente un insieme di pacchetti del modulo della politica SELinux in un singolo modulo della politica pacchetto. Questo strumento non è necessario per il normale funzionamento di SELinux. Nel normale funzionamento, tale collegamento viene eseguito internamente da libsemanage in risposta ai comandi semodule. I pacchetti del modulo sono creati da semodule_package.

semodule_package

semodule_package è lo strumento utilizzato per creare un pacchetto del modulo della politica SELinux da un modulo della politica binaria e facoltativamente altri dati come i contesti dei file. pacchetti semodule_package moduli di politica binaria creati da checkmodule. Il pacchetto di criteri creato da semodule_package può quindi essere installato tramite

semodule.

semodule_unpackage

semodule_unpackage è lo strumento utilizzato per estrarre il modulo della politica SELinux e il file di contesto del file da un pacchetto di criteri SELinux.

sepolgen

Genera un modello di modulo di politica SELinux iniziale

sepolgen-ifgen = ???

sepolgen-ifgen-attr-helper = ???

sepolicy

sepolicy è un set di strumenti che interrogherà la politica SELinux installata e genererà report utili, pagine man o persino nuovi moduli di policy. Vedere l'argomento pagine man specifiche per opzioni e descrizioni. Invocato senza argomenti restituisce una breve spiegazione

```
[root@localhost ~]# sepolicy
usage: sepolicy [-h] [-P POLICY]

                {booleans,communicate,generate,gui,interface,manpage,network,transition}
                ...

SELinux Policy Inspection Tool

positional arguments:
  {booleans,communicate,generate,gui,interface,manpage,network,transition}
                        comandi
    booleans            richiedere alla Policy SELinux di vedere la
                        descrizione dei booleani
    communicate         interrogare la policy SELinux per vedere se i domini
                        possono comunicare con gli altri
    generate            Generare il modello del modulo Policy SELinux
    gui                 Interfaccia Grafica Utente(GUI) per Policy SELinux
    interface           Lista interfacce Policy SELinux
    manpage             Generare pagine man di SELinux
    network             Chiedere le informazioni di rete della policy SELinux
    transition          chiedere alla Policy SELinux di vedere come il dominio
                        del processo sorgente può transitare verso il dominio
                        del processo di destinazione

optional arguments:
  -h, --help            show this help message and exit
  -P POLICY, --policy POLICY
                        Sostituire la policy SELinux, la predefinita è in
                        /sys/fs/selinux/policy
```

sesearch

strumento di interrogazione delle politiche di selinux.

sestatus

riferisce se selinux e’ attivo oppure se non lo e’.

setenforce

attiva oppure mette in pausa selinux

setfiles

Questo programma è utilizzato principalmente per inizializzare i campi del contesto di sicurezza (attributi estesi) su uno o più filesystem (o parti di essi). Di solito è inizialmente eseguito come parte del processo di installazione di SELinux (un passaggio comunemente noto come etichettatura). Può anche essere eseguito in qualsiasi altro momento per correggere etichette incoerenti, aggiungere supporto per la politica appena installata o, usando l'opzione -n, per controllare passivamente se i contesti dei file sono tutti impostati come specificato dalla politica attiva (comportamento predefinito) o da altri criteri (vedere l'opzione -c).
Se un oggetto file non ha un contesto, setfile scriverà il contesto predefinito sugli attributi estesi dell'oggetto file. Se un oggetto file ha un contesto, setfiles modificherà solo la porzione di tipo del contesto di sicurezza. L'opzione -F costringerà a sostituire l'intero contesto.

setsebool

imposta il valore di una determinata boolena di selinux

setroubleshootd

setroubleshootd è il servizio dbus nel sistema setroubleshoot. setroubleshoot è usato per diagnosticare smentite di SELinux e tenta di fornire spiegazioni facili da usare per una negazione di SELinux (ad es. AVC) e raccomandazioni su come si potrebbe regolare il sistema per prevenire il rifiuto in futuro. In una configurazione standard, setroubleshoot è composto da due componenti, sealert e setroubleshootd.
setroubleshootd è un demone di sistema che gira sotto setroubleshoot user e ascolta gli eventi di controllo emessi dal kernel relativo a SELinux. Quando il demone setroubleshootd vede una negazione di SELinux AVC, esegue una serie di plug-in di analisi che esamina i dati di audit relativi all'AVC. Registra i risultati dell'analisi e segnala a tutti i client che sono collegati al demone setroubleshootd che è stato visualizzato un nuovo avviso.

seusers

file ove selinux mappa le utenze

il comando “semanage” merita un approfondimento, poiche’ lo strumento principale con quale configurare selinux. Si compone di diversi sottocomandi:

```
    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
```

ogni sottocomando ha la rispettiva pagina man.

Semanage import

consente di importare una determinata configurazione di selinux. Utile per applicare la stessa policy a piu’ elaboratori, oppure per importare una configurazione sicuramente funzionante.

Semanage export

consente di salvare l'attuale policy di selinux. Utile per trasferire della policy ad altri elaboratori. Oppure puo' essere usata quale backup, prima di modificare una configurazione funzionante.

Semanage login

consente di amministrare la mappatura tra utenti di sistema linux ed utenti di selinux. Infatti il comando

semanage login -l

ci mostra

```
Nome di login        Utente di SELinux    Range MLS/MCS        Servizio

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

quanto sopra sta a significare che un utente standard , viene mappato nel contesto selinux come unconfined_u . l'opzione " -a " aggiunge un login, mappando detto login ad un determinato utente di selinux. l'opzione " -d " elimina un nome di login . La pagina man e' stringata, ma si trova quanto serve per gestire il tutto.

Semange user

gestisce le utenza di selinux. l'opzione " -l " ci mostra le utenze di selinux attive, ed il rispettivo dominio di competenza:

semanage user -l

```
                Etichettatura MLS/       MLS/
Utente di SELinux Prefisso   Livello MCS Range MCS                      Ruoli SELinux

guest_u user    s0     s0     guest_r
root    user    s0     s0-s0:c0.c1023 staff_r sysadm_r system_r unconfined_r
staff_u         user    s0     s0-s0:c0.c1023 staff_r sysadm_r system_r unconfined_r
sysadm_u        user           s0             s0-s0:c0.c1023                sysadm_r
system_u         user       s0         s0-s0:c0.c1023               system_r unconfined_r
unconfined_u     user       s0         s0-s0:c0.c1023               system_r unconfined_r
user_u           user       s0         s0                           user_r
xguest_u         user       s0         s0                           xguest_r
```

Anche in questo caso, l'opzione " -a " aggiunge un nuovo elemento , e l'opzione " -d " cancella un elemento.

Semanage port

istruisce selinux a concedere oppure a non concedere l'utilizzo di porte logiche a determinati programmi/processi. Il comando " selinux port -l " elenca tutte le porte di lavoro utilizzate (quindi concesse) da selinux ai vari programmi processi. Ecco un estratto dell'output

semanage port -l

```
Tipo di porta SELinux          Proto    Numero porta

afs3_callback_port_t           tcp      7001
afs3_callback_port_t           udp      7001
afs_bos_port_t                 udp      7007
afs_fs_port_t                  tcp      2040
afs_fs_port_t                  udp      7000, 7005
afs_ka_port_t                  udp      7004
afs_pt_port_t                  tcp      7002
afs_pt_port_t                  udp      7002
afs_vl_port_t                  udp      7003
agentx_port_t                  tcp      705
agentx_port_t                  udp      705
amanda_port_t                  tcp      10080-10083
amanda_port_t                  udp      10080-10082
amavisd_recv_port_t            tcp      10024
amavisd_send_port_t            tcp      10025
amqp_port_t                    tcp      15672, 5671-5672
```

```
amqp_port_t                    udp      5671-5672
aol_port_t                     tcp      5190-5193
aol_port_t                     udp      5190-5193
apc_port_t                     tcp      3052
apc_port_t                     udp      3052
```

qualcosa di piu' selettivo potrebbe essere l'esempio seguente

```
semange port -l | grep -i ssh

man semanage port | grep -i ssh
Allow sshd to listen on tcp port 8991
# semanage port -a -t ssh_port_t -p tcp 8991
```

otteniamo il suggerimento su come fare per istruire selinux a concedere al demone sshd anche la porta 8991 oltre alla canonica 22. Otteniamo conferma consultando il file /etc/ssh/sshd_config , nel cui interno troviamo le seguenti righe

```
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
```

il comando semanage port -l | grep -w http_port_t restituisce tutte le porte logiche concesse da selinux al demone httpd, in questo caso

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

interrogando il comando sepolicy network -t http_port_t , otteniamo come output

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

come ulteriore conferma.

Semanage ibpkey = non ho trovato informazioni

semange ibendport = non ho trovato informazioni

i due comandi precedenti istruiscono selinux a configurare le comunicazioni infiniband

semanage interface

istruisce selinux su come configurare le interfacce di rete, assegnando loro (oppure non assegnando)
un determinato contesto. Il seguente esempio

```
 semanage interface -a -t bin_t eth1
```

abbina il dominio bin_t all'interfaccia di rete eth1. d'ora innanzi eth1 obbedira' alle direttive del dominio bin_t .

```
semanage interface -d -t bin_t enps03
```

annulla il comando precedente .

Semanage module

installa, disinstalla oppure rimuove i moduli di selinux . Il comando " semange module -l " ci restituisce tutti i moduli disponibili nel sistema.

Semanage node

non trovo informazioni

semange fcontext

questo comando viene utilizzato per gestire l’etichettatura del filesystem. Il comando “ semange fcontext -l “ restituisce l’elenco delle etichettature del filesystem. l’opzione -a aggiunge una nuova etichettatura ma non la rende attiva. Per rendere attiva l’etichettatura appena creata, occorre lancora il comando restorecon indicando il path completo del file/directory in oggetto.

Semanage boolen

comando tramite il quale si amministrano le boolene di selinux . La solita opzione -l (oppure - - list) restituisce l’elenco delle booleane presenti nel sistema

semanage permissive

imposta oppure rimuove la polita permissiva per un determinato dominio. In pratica concede molta piu’ liberta’ ad un determinato contesto. Il comando semanage permissive -a ssh_t imposta il contesto ssh_t in modalita’ permissiva. Mentre il comando semanage permissive -d ssh_t riporta il contesto ssh_t alla modalita’ originaria .

Semanage dontaudit

accetta due opzioni : on e off . Non trovo altre utili info

fonti consultate :

https://docs.fedoraproject.org/it-IT/Fedora/13/html/Security-Enhanced_Linux/chap-Security-Enhanced_Linux-Introduction.html