Fedora 13

Security-Enhanced Linux (SELinux)

Guida Utente

Murray McAllister

Scott Radvan

Daniel Walsh

Dominick Grift

Eric Paris

James Morris

Fedora 13 Security-Enhanced Linux (SELinux)
Guida Utente
Edizione 1.5

Autore	Murray McAllister	mmcallis@redhat.com
Autore	Scott Radvan	sradvan@redhat.com
Autore	Daniel Walsh	dwalsh@redhat.com
Autore	Dominick Grift	domg472@gmail.com
Autore	Eric Paris	eparis@parisplace.org
Autore	James Morris	jmorris@redhat.com

La guida utente di SELinux assiste gli utenti e gli amministratori ad amministrare e utilizzare Security-Enhanced Linux®.

Prefazione

La Guida Utente di SELinux per Fedora 13 , è rivolta a chi ha una minima o nessuna esperienza con SELinux. Anche se non è necessaria un esperienza da amministratori di sistema, il contenuto in questa guida è scritto per compiti di amministrazione del sistema. Questa guida fornisce una introduzione ai concetti fondamentali di SELinux, ed alle sue.applicazioni pratiche. Dopo aver letto la guida, si dovrebbe avere una discreta conoscenza di SELinux.

Grazie a tutti coloro che hanno ci hanno incoraggiati, fornito assistenza e critiche costruttive - ogni contributo è stato molto apprezzato. Un.ringraziamento particolare a:

- Dominick Grift, Stephen Smalley e Russell Coker per i loro contributi, il loro aiuto e la loro pazienza.
- Karsten Wade per il suo aiuto, per aver aggiunto su *Red Hat Bugzilla*[1] un componente per questa guida, e per aver ordinato i contenuti su *http://docs.fedoraproject.org/*.
- Il *Fedora Infrastructure Team*[2] per aver fornito lo spazio web.
- Jens-Ulrik Petersen il quale assicura che la sede Red Hat di Brisbane abbia mirrors di Fedora aggiornati.

1. Convenzioni del documento

Questo manuale utilizza numerose convenzioni per evidenziare parole e frasi, ponendo attenzione su informazioni specifiche.

Nelle edizioni PDF e cartacea questo manuale utilizza caratteri presenti nel set *Font Liberation*[3]. Il set Font Liberation viene anche utilizzato nelle edizioni HTML se il set stesso è stato installato sul vostro sistema. In caso contrario, verranno mostrati caratteri alternativi ma equivalenti. Da notare: Red Hat Enterprise Linux 5 e versioni più recenti, includono per default il set Font Liberation.

1.1. Convenzioni tipografiche

Vengono utilizzate quattro convenzioni tipografiche per richiamare l'attenzione su parole e frasi specifiche. Queste convenzioni, e le circostanze alle quali vengono applicate, sono le seguenti.

`Neretto monospazio`

Usato per evidenziare l'input del sistema, incluso i comandi della shell, i nomi dei file ed i percorsi. Utilizzato anche per evidenziare tasti e combinazione di tasti. Per esempio:

> Per visualizzare i contenuti del file **`my_next_bestselling_novel`** nella vostra directory di lavoro corrente, inserire il comando **`cat my_next_bestselling_novel`** al prompt della shell e premere **`Invio`** per eseguire il comando.

Quanto sopra riportato include il nome del file, un comando della shell ed un tasto, il tutto riportato in neretto monospazio e distinguibile grazie al contesto.

Le combinazioni di tasti possono essere distinte dai tasti tramite il trattino che collega ogni parte della combinazione. Per esempio:

[3] https://fedorahosted.org/liberation-fonts/

Premere **Invio** per eseguire il comando.

Premere **Ctrl**+**Alt**+**F1** per smistarsi sul primo virtual terminal. Premere **Ctrl**+**Alt**+**F7** per ritornare alla sessione X-Windows.

Il primo paragrafo evidenzia il tasto specifico singolo da premere. Il secondo riporta due combinazioni di tasti, (ognuno dei quali è un set di tre tasti premuti contemporaneamente).

Se si discute del codice sorgente, i nomi della classe, i metodi, le funzioni i nomi della variabile ed i valori ritornati indicati all'interno di un paragrafo, essi verranno indicati come sopra, e cioè in **neretto monospazio**. Per esempio:

Le classi relative ad un file includono **filesystem** per file system, **file** per file, e **dir** per directory. Ogni classe possiede il proprio set associato di permessi.

Proportional Bold

Ciò denota le parole e le frasi incontrate su di un sistema, incluso i nomi delle applicazioni; il testo delle caselle di dialogo; i pulsanti etichettati; le caselle e le etichette per pulsanti di selezione, titoli del menu e dei sottomenu. Per esempio:

Selezionare **Sistema** → **Preferenze** → **Mouse** dalla barra del menu principale per lanciare **Preferenze del Mouse**. Nella scheda **Pulsanti**, fate clic sulla casella di dialogo **mouse per mancini**, e successivamente fate clic su **Chiudi** per cambiare il pulsante primario del mouse da sinistra a destra (rendendo così il mouse idoneo per un utilizzo con la mano sinistra).

Per inserire un carattere speciale in un file **gedit**, selezionare **Applicazioni** → **Accessori** → **Mappa carattere** dalla barra menu principale. Successivamente, selezionare **Cerca** → **Trova…** dalla barra del menu **Mappa carattere**, inserire il nome del carattere nel campo **Cerca** e cliccare **Successivo**. Il carattere ricercato verrà evidenziato nella **Tabella caratteri**. Fare un doppio clic sul carattere evidenziato per posizionarlo nel campo **Testo da copiare**, e successivamente fare clic sul pulsante **Copia**. Ritornare ora al documento e selezionare **Modifica** → **Incolla** dalla barra del menu di **gedit**.

Il testo sopra riportato include i nomi delle applicazioni; nomi ed oggetti del menu per l'intero sistema; nomi del menu specifici alle applicazioni; e pulsanti e testo trovati all'interno di una interfaccia GUI, tutti presentati in neretto proporzionale e distinguibili dal contesto.

Corsivo neretto monospazio o ***Corsivo neretto proporzionale***

Sia se si tratta di neretto monospazio o neretto proporzionale, l'aggiunta del carattere corsivo indica un testo variabile o sostituibile . Il carattere corsivo denota un testo che non viene inserito letteralmente, o visualizzato che varia a seconda delle circostanze. Per esempio:

Per collegarsi ad una macchina remota utilizzando ssh, digitare **ssh *username*@*domain.name*** al prompt della shell. Se la macchina remota è **example.com** ed il nome utente sulla macchina interessata è john, digitare **ssh john@example.com**.

Il comando **mount -o remount *file-system*** rimonta il file system indicato. Per esempio, per rimontare il file system **/home**, il comando è **mount -o remount /home**.

> Per visualizzare la versione di un pacchetto attualmente installato, utilizzare il comando **`rpm -q package`**. Esso ritornerà il seguente risultato: ***`package-version-release`***.

Da notare la parola in Corsivo neretto — nome utente, domain.name, file-system, pacchetto, versione e release. Ogni parola racchiude il testo da voi inserito durante l'emissione di un comando o per il testo mostrato dal sistema.

Oltre all'utilizzo normale per la presentazione di un titolo, il carattere Corsivo denota il primo utilizzo di un termine nuovo ed importante. Per esempio:

> Publican è un sistema di pubblicazione per *DocBook*.

1.2. Convenzioni del documento

Gli elenchi originati dal codice sorgente e l'output del terminale vengono evidenziati rispetto al testo circostante.

L'output inviato ad un terminale è impostato su **`tondo monospazio`** e così presentato:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Gli elenchi del codice sorgente sono impostati in **`tondo monospazio`** ma vengono presentati ed evidenziati nel modo seguente:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Note ed avvertimenti

E per finire, tre stili vengono usati per richiamare l'attenzione su informazioni che in caso contrario potrebbero essere ignorate.

Nota Bene

Una nota è un suggerimento o un approccio alternativo per il compito da svolgere. Non dovrebbe verificarsi alcuna conseguenza negativa se la nota viene ignorata, ma al tempo stesso potreste non usufruire di qualche trucco in grado di facilitarvi il compito.

Importante

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled 'Important' won't cause data loss but may cause irritation and frustration.

Avvertenza

Un Avvertimento non dovrebbe essere ignorato. Se ignorato, potrebbe verificarsi una perdita di dati.

2. Inviateci i vostri commenti!

Se individuate degli errori di battitura in questo manuale, o se pensate di poter contribuire al suo miglioramento, contattateci subito! Inviate i vostri suggerimenti tramite Bugzilla: *http://bugzilla.redhat.com/bugzilla/* sul componente **Fedora Documentation.**

Quando inviate un bug report, assicuratevi di indicare l'identificatore del manuale: *selinux-user-guide*

Se inviate un suggerimento per contribuire al miglioramento della guida, cercate di essere il più specifici possibile. Se avete individuato un errore, indicate il numero della sezione e alcune righe di testo, in modo da agevolare la ricerca dell'errore.

Informazioni sui Marchi

Linux® è un marchio di Linus Torvalds, registrato negli Stati Uniti e in altri Stati.

UNIX è un marchio registrato di The Open Group.

Type Enforcement è un marchio, di Secure Computing, LLC, una affiliata di McAfee, Inc., registrato negli Stati Uniti e in altri Stati. Sia McAfee sia Secure Computing, LLC hanno permesso di usare o di fare riferimento al marchio citato limitatamente a questa guida.

Apache è un marchio di The Apache Software Foundation.

MySQL è un marchio di MYSQL AB registrato negli Stati Uniti e in altri Stati.

Altri prodotti menzionati potrebbero essere marchi delle rispettive società.

1.1. Codice sorgente

Il sorgente XML per questa guida sono disponibili su *http://svn.fedorahosted.org/svn/selinuxguide/*

Introduzione

Security-Enhanced Linux (SELinux) è un implementazione di un meccanismo *mandatory access control* nel kernel Linux, che controlla quali operazioni sono consentite dopo che i controlli *discretionary access controls* sono stati effettuati. E' stato creato dalla National Security Agency e può forzare regole su file e processi in un sistema Linux, e sulle loro azioni, basandosi su una policy definita

Quando viene utilizzato SELinux, i file, incluse le directory ed i dispositivi sono definiti come oggetti. I processi, come un utente che esegue un comando o un l'applicazione Mozilla® Firefox®, sono definiti soggetti. La maggior parte dei sistemi operativi usano un sistema di controllo degli accessi discrezionale (DAC), che stabilisce come i soggetti interagiscono tra di loro e con gli oggetti. Nei sistemi operativi che usano DAC, gli utenti controllano i permessi sui file (oggetti) di loro proprietà. Per esempio, sui sistemi operativi Linux®, gli utenti possono rendere la loro home directory leggibile a tutti, dando accesso agli altri utenti ed ai processi ad informazioni potenzialmente sensibili.

Relying on DAC mechanisms alone is fundamentally inadequate for strong system security. DAC access decisions are only based on user identity and ownership, ignoring other security-relevant information such as the role of the user, the function and trustworthiness of the program, and the sensitivity and integrity of the data. Each user has complete discretion over their files, making it impossible to enforce a system-wide security policy. Furthermore, every program run by a user inherits all of the permissions granted to the user and is free to change access to the user's files, so no protection is provided against malicious software. Many system services and privileged programs must run with coarse-grained privileges that far exceed their requirements, so that a flaw in any one of these programs could be exploited to obtain further system access.[1]

Il seguente è un esempio di permessi usati sui sistemi operativi Linux che non eseguono Security-Enhanced Linux (SELinux). I permessi e l'output in questi esempi possono differire dal sistema usato. Usare il comando **`ls -l`** per vedere i permessi sui file:

```
$ ls -l file1
-rw-rw-r--. 1 user1 group1 0 May 11 10:46 file1
```

The first three permission bits, **`rw`**, control the access the Linux **`user1`** user (in this case, the owner) has to **`file1`**. The next three permission bits, **`rw-`**, control the access the Linux **`group1`** group has to **`file1`**. The last three permission bits, **`r--`**, control the access everyone else has to **`file1`**, which includes all users and processes.

Security-Enhanced Linux (SELinux) adds Mandatory Access Control (MAC) to the Linux kernel, and is enabled by default in Fedora. A general purpose MAC architecture needs the ability to enforce an administratively-set security policy over all processes and files in the system, basing decisions on labels containing a variety of security-relevant information. When properly implemented, it enables a system to adequately defend itself and offers critical support for application security by protecting against the tampering with, and bypassing of, secured applications. MAC provides strong separation of applications that permits the safe execution of untrustworthy applications. Its ability to limit the privileges associated with executing processes limits the scope of potential damage that can result from the exploitation of vulnerabilities in applications and system services. MAC enables information

"Integrating Flexible Support for Security Policies into the Linux Operating System", by Peter Loscocco and Stephen Smalley. This paper was originally prepared for the National Security Agency and is, consequently, in the public domain. Refer to the *original paper* [http://www.nsa.gov/research/_files/selinux/papers/freenix01/index.shtml] for details and the document as it was first released. Any edits and changes were done by Murray McAllister.

to be protected from legitimate users with limited authorization as well as from authorized users who have unwittingly executed malicious applications.[2]

Il seguente è un esempio di etichette contenenti informazioni rilevanti per la sicurezza, usate su processi, utenti Linux e file, in sistemi operativi che eseguono SELinux. Questa informazione viene definita *contesto* SELinux e può essere visualizzata tramite il comando **`ls -Z`**:

```
$ ls -Z file1
-rw-rw-r--. user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

Nell'esempio, SELinux fornisce un utente (**`unconfined_u`**), un ruolo (**`object_r`**), un tipo (**`user_home_t`**), ed un livello (**`s0`**). Tale informazione è usata per decidere sul controllo degli accessi. Con DAC, l'accesso è controllato soltanto dagli ID degli utenti e dei gruppi. E' importante ricordare che le regole della politica di SELinux sono verificate *dopo* le regole DAC. Se le regole DAC, in primis, negano l'accesso, le regole della policy SELinux non vengono verificate.

Utenti Linux ed utenti SELinux

On Linux operating systems that run SELinux, there are Linux users as well as SELinux users. SELinux users are part of SELinux policy. Linux users are mapped to SELinux users. To avoid confusion, this guide uses "Linux user" and "SELinux user" to differentiate between the two.

2.1. I vantaggi di eseguire SELinux

- 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, essendo 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.
- Controllo d'accesso calibrato. Superando il tradizionale sistema UNIX® dei permessi che sono impostati a discrezione dell'utente e basati sugli IDs dell'utente e di gruppo, i permessi d'accesso di SELinux si basano su tutte le informazioni disponibili, ossia utente di SELinux, ruolo, tipo e, opzionalmente livello.
- La politica di SELinux è definita amministrativamente, imposta a tutto il sistema, e non è a discrezione dell'utente.
- Ridotta vulnerabilità da attacchi pertinenti la scalata di privilegio. Un esemipo: giacché i processi sono confinati in domini, e sono perciò separati l'uno dall'altro, e poiché le regole di politica di SELinux decidono come i processi accedono ad altri processi e ai file, se un processo viene compromesso, l'attaccante può accedere soltanto alle normali funzioni di quel processo ed ai file da esso gestiti. Ad esempio se il server HTTP Apache, viene compromesso, un attaccante non potrà usare quel processo per leggere i file nelle home directory degli utenti, a meno che non sia stata specificata una regola della politica di SELinux tale da autorizzarne l'accesso.
- Confined services. SELinux ships with the ability to confine services and daemons so that they are more predictable and are only allowed access that is required for their normal operation.

"Meeting Critical Security Objectives with Security-Enhanced Linux", by Peter Loscocco and Stephen Smalley. This paper was originally prepared for the National Security Agency and is, consequently, in the public domain. Refer to the *original paper* [http://www.nsa.gov/research/_files/selinux/papers/ottawa01/index.shtml] for details and the document as it was first released. Any edits and changes were done by Murray McAllister.

- SELinux può essere usato per imporre sia la confidenzialità e l'integrità dei dati sia per salvaguardare i processi da accessi non autorizzati.

SELinux non è:

- un software antivirus.
- un sostitutivo per password, firewall, o altri sistemi di sicurezza.
- una soluzione di sicurezza omnicomprensiva.

SELinux is designed to enhance existing security solutions, not replace them. Even when running SELinux, it is important to continue to follow good security practices, such as keeping software up-to-date, using hard-to-guess passwords, firewalls, and so on.

2.2. Esempi

Gli esempi riportati di seguito illustreranno come SELinux aumenta la sicurezza:

- The default action is deny. If a specific SELinux policy rule does not exist to allow access, such as for a process opening a file, access is denied.
- SELinux può confinare gli utenti Linux. Esiste un certo numero di utenti SELinux confinati. Gli utenti di Linux si possono mappare agli utenti di SELinux per trarre vantaggio dal delle regole di sicurezza e meccanismi a loro applicati. Per esempio, se ad un utente di Linux si mappa un utente user_u di SELinux, si otterrà un utente di Linux privato della possibilità di eseguire applicazioni (a meno di volerlo configurare diversamente) set user ID (setuid), come **sudo** e **su**, e privato della possibilità di eseguire file o applicazioni nella propria home directory, onde evitare l'esecuzione di codice dannoso.
- Separazione dei processi. I processi sono eseguiti nel proprio dominio, evitando ad essi l'accesso ad altri processi ed ai file di altri processi. Per esempio, con SELinux in esecuzione nella configurazione predefinita, un attaccante non può compromettere un server Samba ed usarlo come vettore d'attacco per leggere o scrivere su file usati da altri processi, come un database usato da MySQL®.
- SELinux helps limit the damage made by configuration mistakes. *Domain Name System (DNS)*[3] servers often replicate information between each other in what is known as a zone transfer. Attackers can use zone transfers to update DNS servers with false information. When running the *Berkeley Internet Name Daemon (BIND)*[4] as a DNS server in Fedora, even if an administrator forgets to limit which servers can perform a zone transfer, the default SELinux policy prevents zone files [5] from being updated via zone transfers, by the BIND `named` daemon itself, and by other processes.
- Refer to the *Red Hat® Magazine*[6] article, *Risk report: Three years of Red Hat Enterprise Linux 4*[78], for exploits that were restricted due to the default SELinux targeted policy in Red Hat® Enterprise Linux® 4.
- Refer to the *LinuxWorld.com*[9] article, *A seatbelt for server software: SELinux blocks real-world exploits*[1011], for background information about SELinux, and information about various exploits that SELinux has prevented.

- Refer to James Morris's *SELinux mitigates remote root vulnerability in OpenPegasus*[12] blog post for information about an exploit in *OpenPegasus*[13] that was mitigated by SELinux as shipped with Red Hat Enterprise Linux 4 and 5.

Sul sito web *Tresys Technology*[14] esiste una sezione *SELinux Mitigation News*[15] che lista i recenti rischi che sono stati mitigati o prevenuti da SELinux.

2.3. L'architettura di SELinux

SELinux è un modulo di sicurezza inserito nel kernel di Linux. SELinux è governato da regole di policy caricabili. Quando si attivano accessi rilevanti per la sicurezza, come quando un processo tenta di aprire un file, l'operazione è intercettata nel kernel tramite SELinux. Se una regola della policy permette l'operazione, essa continua, altrimenti l'operazione viene bloccata ed il processo riceve un errore.

Le decisioni SELinux, come concedere o negare l'accesso, sono memorizzate in una cache. Questa cache è detta Access Vector Cache (AVC). Memorizzare le decisioni nella cache riduce quanto spesso le regole debbano essere controllate, aumentando le prestazioni. Va ricordato che se le regole DAC, in primis negano l'accesso, le regole di politica di SELinux non vengono verificate.

2.4. SELinux su altri sistemi operativi

Per informazioni sull'impiego di SELinux su altri sistemi operativi, fare riferimento ai link seguenti:

- Hardened Gentoo: *http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml*.
- Debian: *http://wiki.debian.org/SELinux*.
- Ubuntu: *https://wiki.ubuntu.com/SELinux* e *https://help.ubuntu.com/community/SELinux*.
- Red Hat Enterprise Linux: *Red Hat Enterprise Linux Deployment Guide*[16] e \n*Red Hat Enterprise Linux 4 SELinux Guide*[17].
- Fedora: *http://fedoraproject.org/wiki/SELinux* e le *Fedora Core 5 SELinux FAQ*[18].

[14] http://www.tresys.com/
[15] http://www.tresys.com/innovation.php

Contesti di SELinux

Processi e file sono contrassegnati con un contesto SELinux che contiene alcune informazioni aggiuntive come, un utente SELinux, un ruolo, un tipo ed opzionalmente un livello. Quando Selinux è in esecuzione, queste informazioni sono usate per controllare gli accessi. In Fedora, SELinux offre una combinazione di controlli Role-Based Access Control (RBAC), Type Enforcment (TE)®, ed opzionalmente, Sicurezza Multi-Level (MLS).

Il seguente è un esempio che mostra il contesto di SELinux. I contesti sono impiegati su processi, utenti e file, nei sistemi operativi Linux che utilizzano SELinux. Usare il. comando **`ls -Z`** per visualizzare il contesto di file e directory:

```
$ ls -Z file1
-rw-rw-r--. user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

I contesti SELinux usano la seguente.sintassi: *utente di SELinux:ruolo:tipo:livello*:

utente SELinux

L'identita dell'utente SELinux è una identità nota alla policy, autorizzata per un insieme specifico di ruoli, e per una specifica gamma MLS. Ciascun utente Linux è mappato ad un utente SELinux attraverso la policy di SELinux. Questo permette agli utenti Linux di ereditare le restrizioni proprie dell'utente SELinux. L'identità dell'utente SELInux è usata nel contesto SELinux per i processi in quella sessione, per limitare a quali ruoli e livelli possono accedere. Usare il comando **`semanage login -l`**, come utente Linux root, per vedere un elenco di mappature fra account utente SELinux e Linux:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

L'output sarà leggermente diverso da sistema a sistema. La colonna **`Login Name`** elenca gli utenti Linux, mentre la colonna **`SELinux User`** i corrispondenti utenti SELinux. Per i processi, l'utente SELinux limita quali ruoli e livelli sono accessibili. L'ultima colonna, **`MLS/MCS Range`**, indica il livello usato dalla Multi-Level Security (MLS) e dalla Multi-Category Security (MCS). I livelli saranno trattati brevemente in seguito.

ruolo

Fa parte di SELinux il modello di sicurezza denominato controllo d'accesso basato sul ruolo (Role-Based Access Control - RBAC). Il ruolo è un attributo di RBAC. Gli utenti SELinux sono autorizzati per i ruoli e i ruoli sono autorizzati per i domini. Il ruolo funge da intermediario tra i domini e gli utenti SELinux. I ruoli che possono essere assegnati determinano a quali domini è possibile accedere - in definitiva, controllano a quali tipi di oggetti è possibile accedere. Ciò contribuisce a ridurre la vulnerabilità da attacchi che puntano a scalare i privilegi.

tipo

Il tipo è un attributo di Type Enforcement. Il tipo definisce un dominio per i processi e un tipo per i file. Le regole della policy di SELinux definiscono come i tipi accedono l'un l'altro, come un

dominio accede ad un tipo o come un dominio accede a un altro dominio. L'accesso è consentito soltanto se esiste una specifica regola della policy SELinux che lo permetta.

livello
Il livello è un attributo di MLS e della Multi-Category Security (MCS). Una gamma MLS è una coppia di livelli, indicata come *basso_livello-alto_livello* se i livelli sono diversi, o *basso_livello* se identici. (**s0-s0** equivale a **s0**). Ciascun livello rappresenta una coppia di termini sensibilità-categoria, in cui le categorie sono opzionali. Se presente il termine categoria, il livello è indicato come *sensibilità:insieme-di-categorie*. Se il termine categoria è assente, il livello si riduce a *sensibiltà*.

If the category set is a contiguous series, it can be abbreviated. For example, **c0.c3** is the same as **c0,c1,c2,c3**. The **/etc/selinux/targeted/setrans.conf** file maps levels (**s0:c0**) to human-readable form (ie. **CompanyConfidential**). Do not edit **setrans.conf** with a text editor: use **semanage** to make changes. Refer to the semanage(8) manual page for further information. In Fedora, targeted policy enforces MCS, and in MCS, there is just one sensitivity, **s0**. MCS in Fedora supports 1024 different categories: **c0** through to **c1023**. **s0-s0:c0.c1023** is sensitivity **s0** and authorized for all categories.

MLS enforces the *Bell-La Padula Mandatory Access Model*[1], and is used in Labeled Security Protection Profile (LSPP) environments. To use MLS restrictions, install the *selinux-policy-mls* package, and configure MLS to be the default SELinux policy via the **/etc/selinux/config** file. The MLS policy shipped with Fedora omits many program domains that were not part of the evaluated configuration, and therefore, MLS on a desktop workstation is unusable (no support for the X Window System); however, an MLS policy from the *upstream SELinux Reference Policy*[2] can be built that includes all program domains.

3.1. Transizioni tra Domini

Un processo transita da un dominio a un altro dominio eseguendo un'applicazione che ha un **entrypoint** nel dominio di destinazione. Il permesso del **entrypoint** è usato dalla policy SELinux e controlla quali applicazioni possono accedere a un dominio. Il seguente esempio dimostra una transizione di dominio:

1. Un utente vuole cambiare la propria password. Per farlo l'utente esegue l'applicazione **passwd**. L'eseguibile **/usr/bin/passwd** è contrassegnato dal tipo **passwd_exec_t**:

```
$ ls -Z /usr/bin/passwd
-rwsr-xr-x  root root system_u:object_r:passwd_exec_t:s0 /usr/bin/passwd
```

   L'applicazione **passwd** accede al file **/etc/shadow**, il quale è contrassegnato dal tipo **shadow_t**:

```
$ ls -Z /etc/shadow
----------. root root system_u:object_r:shadow_t:s0    /etc/shadow
```

2. Una regola della policy stabilisce che i processi in esecuzione nel dominio **passwd_t** possono leggere e scrivere in file contrassegnati con il tipo **shadow_t**. Il tipo **shadow_t** si applica solo ai

[1] http://en.wikipedia.org/wiki/Bell-LaPadula_model
[2] http://oss.tresys.com/projects/refpolicy

file interessati da un cambiamento di password. Questi includono **/etc/gshadow**, **/etc/shadow** ed i rispettivi file di backup.

3. Una regola della policy stabilisce che il dominio **passwd_t** ha un **entrypoint** per accedere al tipo **passwd_exec_t**.

4. When a user runs the **/usr/bin/passwd** application, the user's shell process transitions to the **passwd_t** domain. With SELinux, since the default action is to deny, and a rule exists that allows (among other things) applications running in the **passwd_t** domain to access files labeled with the **shadow_t** type, the **passwd** application is allowed to access **/etc/shadow**, and update the user's password.

L'esempio non è certo esaustivo, ed è usato come esempio di partenza per spiegare le transizioni di dominio. Infatti, sebbene esista una regola che permette a certi soggetti in esecuzione nel dominio **passwd_t** di accedere ad oggetti contrassegnati con l'etichetta **shadow_t**, altre regole di policy di SELinux devono essere rispettate perché un soggetto possa transitare in un nuovo dominio. In questo esempio, Type Enforcement garantisce quanto segue:

- è possibile accedere al dominio **passwd_t** soltanto eseguendo un'applicazione etichettata con il tipo **passwd_exec_t**; può essere eseguita soltanto da librerie condivise autorizzate come il tipo **lib_t**; e non può avviare alcun altra applicazione.

- soltanto i domini autorizzati, come **passwd_t** possono scrivere su file etichettati con il tipo **shadow_t**. Anche se esistono altri processi in esecuzione con privilegi di superuser, questi processi non possono scrivere su file etichettati con il tipo **shadow_t**, poiché non sono in esecuzione nel dominio **passwd_t**.

- soltanto domini autorizzati possono transitare nel dominio **passwd_t**. Per esempio, il processo `sendmail`, in esecuzione nel dominio **sendmail_t** non ha una ragione legittima per eseguire **passwd**; perciò, esso non potrà mai transitare nel dominio **passwd_t**.

- i processi in esecuzione nel dominio **passwd_t** possono soltanto leggere e scrivere su file etichettati con tipi specifici, come **etc_t** o **shadow_t**. Questo evita che l'applicazione **passwd** possa leggere o scrivere su file arbitrari.

3.2. I contesti di SELinux per i processi

Usare il comando **ps -eZ** per vedere il contesto di SELinux per i processi. Per esempio:

1. Aprire un terminale, **Applicazioni** → **Strumenti di sistema** → **Terminale**.

2. Eseguire il comando **/usr/bin/passwd**. Non inserire una nuova password.

3. In una nuova scheda, o in un altro terminale, eseguire il comando **ps -eZ | grep passwd**. L'output sarà simile a:

```
unconfined_u:unconfined_r:passwd_t:s0-s0:c0.c1023 13212 pts/1 00:00:00 passwd
```

4. Nella prima scheda/terminale, premere **Ctrl+C** per chiudere l'applicazione **passwd**.

In this example, when the **/usr/bin/passwd** application (labeled with the **passwd_exec_t** type) is executed, the user's shell process transitions to the **passwd_t** domain. Remember: the type defines a domain for processes, and a type for files.

Usare il comando **ps -eZ** per visualizzare i contesti SELinux per i processi in esecuzione. Ciò che segue è un esempio ridotto dell'output, e può essere diverso sul sistema in uso:

```
system_u:system_r:dhcpc_t:s0     1869 ?        00:00:00 dhclient
system_u:system_r:sshd_t:s0-s0:c0.c1023 1882 ? 00:00:00 sshd
system_u:system_r:gpm_t:s0       1964 ?        00:00:00 gpm
system_u:system_r:crond_t:s0-s0:c0.c1023 1973 ? 00:00:00 crond
system_u:system_r:kerneloops_t:s0 1983 ?       00:00:05 kerneloops
system_u:system_r:crond_t:s0-s0:c0.c1023 1991 ? 00:00:00 atd
```

Il ruolo **system_r** è usato per i processi di sistema, come i demoni. Type Enforcement quindi separa ciascun dominio.

3.3. I contesti di SELinux per gli utenti

Digitare il comando **id -Z**per vedere il contesto SELinux associato agli utenti di Linux:

```
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

In Fedora, gli utenti Linux sono non confinati per impostazione predefinita. Questo contesto SELinux mostra che l'utente Linux è mappato all'utente **unconfined_u** di SELinux, in esecuzione con il ruolo **unconfined_r** e nel dominio **unconfined_t**. **s0-s0** è una gamma MLS, che in questo caso coincide con **s0**. Le categorie a cui l'utente ha accesso sono definite da **c0.c1023**, cioè tutte le categorie (da **c0** a **c1023**).

Targeted Policy

La targeted policy è la politica predefinita di SELinux usata in Fedora. Usando tale politica, i processi soggetti ad essa sono eseguiti in in dominio confinato, gli altri in un dominio non confinato. Per esempio, gli utenti loggati lavorano nel dominio **unconfined_t** ed i processi di sistema avviati da init lavorano nel dominio **initrc_t**, entrambi non confinati.

I domini non confinati (come quelli confinati) sono sottoposti a controlli sulla memoria per le operazioni di scrittura o esecuzione. Per impostazione predefinita, i soggetti in esecuzione in un dominio non confinato non possono allocare memoria scrivibile ed eseguirla. Ciò riduce la vulnerabilità da attacchi di *Buffer Overflow*[1]. Questi controlli di memoria sono disattivabili impostando booleane, che permettono alla policy di SELinux di modificarsi al runtime. La configurazione delle booleane sarà trattata in seguito.

4.1. Processi confinati

Almost every service that listens on a network is confined in Fedora. Also, most processes that run as the Linux root user and perform tasks for users, such as the **passwd** application, are confined. When a process is confined, it runs in its own domain, such as the `httpd` process running in the **httpd_t** domain. If a confined process is compromised by an attacker, depending on SELinux policy configuration, an attacker's access to resources and the possible damage they can do is limited.

L'esempio seguente dimostra come SELinux impedisca al Server HTTP Apache (`httpd`) di leggere quei file che non siano correttamente etichettati, come per esempio quelli usati da Samba. Si assume che siano installati i pacchetti *httpd*, *wget*, *setroubleshoot-server*, ed *audit*, che sia in uso la targeted policy di SELinux e che quest'ultimo sia in esecuzione in modalità enforcing:

1. Eseguire il comando **sestatus** per verificare che SELinux sia abilitato, che sia in esecuzione in enforcing mode e che si stia usando la targeted policy:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

   SELinux status: enabled viene restituito quando SELinux è abilitato. **Current mode: enforcing** viene restituito quando SELinux è in esecuzione in modalità enforcing. **Policy from config file: targeted** viene restituito quando la SELinux targeted policy è utilizzata.

2. Come utente root, creare un file, eseguendo il comando **touch /var/www/html/testfile**

3. Eseguire il comando **ls -Z /var/www/html/testfile** per visualizzare il contesto di SELinux:

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/testfile
```

[1] http://en.wikipedia.org/wiki/Buffer_overflow

Per impostazione predefinita in Fedora, gli utenti Linux lavorano non confinati, perciò il file **testfile** è etichettato con l'utente **unconfined_u** di SELinux. RBAC non è impiegato per i file, ma per i processi. I ruoli non hanno senso per i file - il ruolo **object_r** è un ruolo generico usato per i file (memorizzati su disco e nei network file system). Nella directory **/proc/**, i file relativi ai processi usano il ruolo **system_r**. [2] Il tipo **httpd_sys_content_t** permette al processo `httpd` di accedere a questo file.

4. Come utente root, eseguire il comando **service httpd start** per avviare il processo `httpd`. Se il processo si avvia correttamente, l'uscita sarà:

```
# /sbin/service httpd start
Starting httpd:                                            [  OK  ]
```

5. Spostarsi in una directory dove si ha il permesso di scrittura ed eseguire il comando **wget http://localhost/testfile** . A meno di cambiamenti alla configurazione predefinita, il risultato sarà:

```
$ wget http://localhost/testfile

--2010-05-11 13:19:07--  http://localhost/testfile
Resolving localhost... ::1, 127.0.0.1
Connecting to localhost|::1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: “testfile”

    [ <=>            ] 0           --.-K/s   in 0s

2010-05-11 13:19:07 (0.00 B/s) - “testfile” saved [0/0]
```

6. Il comando **chcon** rietichetta i file; tuttavia tali modifiche vengono perdute se viene rietichettato l'intero file system. Per.salvare in modo permanente le modifiche affinchè sopravvivano a rietichettature dell'intero file system usare il comando **semanage**, discusso di seguito. Come utente root, eseguire il seguente comando per modificare il tipo del contesto in un tipo usato da Samba:

   chcon -t samba_share_t /var/www/html/testfile

   Per verificare il risultato, eseguire il comando **ls -Z /var/www/html/testfile**:

```
-rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/testfile
```

7. Nota: i permessi DAC correnti, consentono al processo `httpd` di accedere al file **testfile**. Spostarsi in una directory dove il nostro utente Linux ha il permesso di scrittura ed eseguire il comando **wget http://localhost/testfile**. A meno che non ci siano cambiamenti alla configurazione predefinita, il comando fallirà:

```
$ wget http://localhost/testfile
```

```
--2010-05-11 13:23:49--  http://localhost/testfile
Resolving localhost... ::1, 127.0.0.1
Connecting to localhost|::1|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2010-05-11 13:23:49 ERROR 403: Forbidden.
```

8. Come utente Linux root, eseguire **`rm -i /var/www/html/testfile`** per cancellare il file **`testfile`**.

9. Se il servizio `httpd`, non è necessario, come utente root eseguire.il comando **`service httpd stop`** per arrestare `httpd`:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

Questo esempio dimostra la sicurezza aggiuntiva portata impiegando SELinux. Le regole DAC consentono al processo `httpd` di accedere al file **`testfile`** al passo 7, ma poiché il file è rietichettato con un tipo cui il processo `httpd` non ha il permesso d'accesso, SELinux nega l'accesso. Dopo il passo 7, un messagio d'errore simile al seguente è registrator in **`/var/log/messages`**:

```
May 11 13:23:51 localhost setroubleshoot: SELinux is preventing /usr/sbin/httpd "getattr"
 access to /var/www/html/testfile. For complete SELinux messages. run sealert -l ca2ab0df-
fcb9-46d1-8283-037450d1efcc
```

I file di log meno recenti potrebbero usare un formato del tipo **`/var/log/messages.`*`YYYYMMDD`***. Se è in esecuzione il processo **syslog-ng**, i file di log meno recenti hanno un formato del tipo **`/var/log/messages.`*`X`***. Se i processi `setroubleshootd` ed `auditd` sono in esecuzione, errori simili al seguente sono registrati sul file **`/var/log/audit/audit.log`**:

```
type=AVC msg=audit(1220706212.937:70): avc:  denied  { getattr } for  pid=1904 comm="httpd"
 path="/var/www/html/testfile" dev=sda5 ino=247576 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0  tclass=file

type=SYSCALL msg=audit(1220706212.937:70): arch=40000003 syscall=196 success=no exit=-13
 a0=b9e21da0 a1=bf9581dc a2=555ff4 a3=2008171 items=0 ppid=1902 pid=1904 auid=500 uid=48
 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=1 comm="httpd" exe="/
usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Inoltre, un errore simile è registrato anche sul file **`/var/log/httpd/error_log`**:

```
[Tue May 11 13:23:49 2010] [error] [client ::1] (13)Permission denied: access to /testfile
 denied
```

4.2. Processi non confinati

I processi non confinati sono eseguiti in domini non confinati, per esempio i programmi init sono eseguiti nel dominio **`initrc_t`**, i processi non confinati del kernel e gli utenti non confinati sono eseguiti rispettivamente nei domini **`kernel_t`** ed **`unconfined_t`**. Ai processi non confinati, si applicano le regole della politica di SELinux, ma esistono regole tali da consentire ai processi

non confinati pressoché tutti i permessi di accesso. I processi eseguiti in tali domini, utilizzano esclusivamente le regole DAC. Se un tale processo viene compromesso, SELinux non può impedire ad un attaccante di avere accesso ai dati ed alle risorse di sistema, ma ovviamente le regole DAC restano ancora valide. SELinux è un avanzamento della sicurezza che si aggiunge alle regole DAC, - non si sostituisce.ad esse.

Il seguente esempio dimostra come il Server HTTP Apache (`httpd`) può accedere ai dati usati da Samba, quando è in esecuzione non confinato. Nota: in Fedora, il processo `httpd` è in esecuzione confinato nel dominio **`httpd_t`**, per impostazione predefinita. Questo è un esempio che non andrebbe usato su un sistema di produzione. Si assume che siano installati i pacchetti *httpd*, *wget*, *setroubleshoot-server*, *dbus* ed *audit*, che sia usata la targeted policy di SELinux e che SELinux sia in esecuzione in modalità enforcing:

1. Eseguire il comando **`sestatus`** per verificare che SELinux sia abilitato, che sia in esecuzione in enforcing mode e che si stia usando la targeted policy:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 24
Policy from config file:        targeted
```

   `SELinux status: enabled` viene restituito quando SELinux è abilitato. **`Current mode: enforcing`** viene restituito quando SELinux è in esecuzione in modalità enforcing. **`Policy from config file: targeted`** viene restituito quando la SELinux targeted policy è utilizzata.

2. Come utente root, eseguire il comando **`touch /var/www/html/test2file`** per creare un file.

3. Eseguire il comando **`ls -Z /var/www/html/test2file`** per visualizzare il contesto di SELinux:

```
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test2file
```

   Per impostazione predefinita in Fedora, gli utenti di Linux sono non confinati, perciò il file **`test2file`** è etichettato con l'utente **`unconfined_u`** di SELinux. RBAC non è impiegato per i file, ma per i processi. I ruoli non hanno significato per i file - il ruolo **`object_r`**, è un ruolo generico usato per i file (memorizzati su disco o nei network file system). Nella directory **`/proc/`**, i file relativi ai processi usano il ruolo **`system_r`**. [3] Il tipo **`httpd_sys_content_t`** permette al processo `httpd` di accedere a questo file.

4. Il comando **`chcon`** rietichetta i file; tuttavia tali modifiche vengono perdute se viene rietichettato l'intero file system. Per.salvare in modo permanente le modifiche affinchè sopravvivano a rietichettature dell'intero file system usare il comando **`semanage`**, discusso di seguito. Come utente root, eseguire il seguente comando per modificare il tipo del contesto in un tipo usato da Samba:

   `chcon -t samba_share_t /var/www/html/test2file`

   Per verificare il risultato, eseguire il comando **`ls -Z /var/www/html/test2file`**:

```
-rw-r--r--  root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/test2file
```

5. Eseguire il comando **`service httpd status`** per verificare che il processo `httpd` non sia in esecuzione:

   ```
   $ /sbin/service httpd status
   httpd is stopped
   ```

   Se l'output è diverso, eseguire il comando **`service httpd stop`**, come utente Linux root, per arrestare il processo `httpd`:

   ```
   # /sbin/service httpd stop
   Stopping httpd:                                            [  OK  ]
   ```

6. Per fare in modo che il processo `httpd` sia in esecuzione non confinato, modificare il tipo di **`/usr/sbin/httpd`** in un tipo che non transiti in un dominio confinato, eseguendo come root il comando:

   `chcon -t unconfined_exec_t /usr/sbin/httpd`

7. Eseguire il comando **`ls -Z /usr/sbin/httpd`** per verificare che **`/usr/sbin/httpd`** sia etichettato con il tipo **`unconfined_exec_t`**:

   ```
   -rwxr-xr-x  root root system_u:object_r:unconfined_exec_t /usr/sbin/httpd
   ```

8. Come utente root, eseguire il comando **`service httpd start`** per avviare il processo `httpd`. Se il processo si avvia correttamente, l'uscita sarà:

   ```
   # /sbin/service httpd start
   Starting httpd:                                            [  OK  ]
   ```

9. Eseguire il comando **`ps -eZ | grep httpd`** per verificare che `httpd` sia in esecuzione nel dominio **`unconfined_t`**:

   ```
   $ ps -eZ | grep httpd
   unconfined_u:system_r:unconfined_t 7721 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7723 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7724 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7725 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7726 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7727 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7728 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7729 ?      00:00:00 httpd
   unconfined_u:system_r:unconfined_t 7730 ?      00:00:00 httpd
   ```

10. Spostarsi in una directory dove il nostro utente Linux abbia il permesso di scrittura ed eseguire il comando **wget http://localhost/test2file** . A meno che non ci siano stati ulteriori cambiamenti alla configurazione predefinita, il comando riuscirà:

```
--2009-05-07 01:41:10--  http://localhost/test2file
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 0 [text/plain]
Saving to: `test2file.1'

[ <=>                                 ]--.-K/s   in 0s

2009-05-07 01:41:10 (0.00 B/s) - `test2file.1' saved [0/0]
```

    Sebbene il processo `httpd` non abbia il permesso d'accesso ai file etichettati con il tipo **samba_share_t**, `httpd` è in esecuzione nel dominio non confinato **unconfined_t**, e ricorre alle regole DAC, perciò il comando **wget** ha successo. Se `httpd` fosse stato in esecuzione nel dominio confinato **httpd_t**, il comando **wget** sarebbe fallito.

11. Il comando **restorecon** ripristina il contesto predefinito di SELinux per i file. Come utente root, eseguire **restorecon -v /usr/sbin/httpd**, per ripristinare il contesto predefinito di SELinux per **/usr/sbin/httpd**:

```
# /sbin/restorecon -v /usr/sbin/httpd
restorecon reset /usr/sbin/httpd context system_u:object_r:unconfined_notrans_exec_t:s0-
>system_u:object_r:httpd_exec_t:s0
```

    Eseguire il comando **ls -Z /usr/sbin/httpd** per verificare che **/usr/sbin/httpd** sia etichettato con il tipo **httpd_exec_t**:

```
$ ls -Z /usr/sbin/httpd
-rwxr-xr-x  root root system_u:object_r:httpd_exec_t   /usr/sbin/httpd
```

12. Come utente Linux root, eseguire il comando **/sbin/service httpd restart** per riavviare `httpd`. Dopo il riavvio, eseguire **ps -eZ | grep httpd** per verificare che `httpd` sia in esecuzione nel dominio confinato **httpd_t**:

```
# /sbin/service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
# ps -eZ | grep httpd
unconfined_u:system_r:httpd_t    8880 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8882 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8883 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8884 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8885 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8886 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8887 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8888 ?        00:00:00 httpd
unconfined_u:system_r:httpd_t    8889 ?        00:00:00 httpd
```

13. Come utente Linux root, eseguire **`rm -i /var/www/html/test2file`** per cancellare il file **`test2file`**.

14. Se il servizio `httpd`, non è necessario, come utente root eseguire.il comando **`service httpd stop`** per arrestare `httpd`:

```
# /sbin/service httpd stop
Stopping httpd:                                            [  OK  ]
```

Gli esempi in queste sezioni hanno dimostrato come sia possibile proteggere i dati da un processo-confinato compromesso (protetto da SELInux), e come sia più facile per un attaccante accedere ai dati da un processo-non confinato compromesso (non protetto da SELinux).

4.3. Utenti confinati e non confinati

Ciascun utente di Linux è mappato ad un utente di SELinux per mezzo della politica di SELinux. Ciò permette agli utenti di Linux di ereditare le restrizioni degli utenti di SELinux. Questa mappatura degli utenti Linux è visibile eseguendo il comando **`semanage login -l`**, come utente Linux root:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

In Fedora 13 per impostazione predefinita, agli utenti di Linux è mappato il login **`__default__`** di SELinux, (a cui è mappato l'utente **`unconfined_u`**). Quanto segue definisce la mappatura predefinita:

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

Il seguente esempio dimostra che ogni nuovo utente di Linux sarà mappato all'utente **`unconfined_u`** di SELinux. Si assume che l'utente root stia lavorando non confinato, come già è per impostazione predefinita in Fedora 13:

1. Come utente Linux root, eseguire il comando **`/usr/sbin/useradd newuser`** per creare un nuovo utente Linux di nome newuser.

2. Come utente root, eseguire il comando **`passwd newuser`** ed assegnare una password all'utente Linux newuser:

```
# passwd newuser
Changing password for user newuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

3. Log out of your current session, and log in as the Linux newuser user. When you log in, pam_selinux maps the Linux user to an SELinux user (in this case, unconfined_u), and sets up the

resulting SELinux context. The Linux user's shell is then launched with this context. Run the **id -Z** command to view the context of a Linux user:

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

4. Log out of the Linux newuser's session, and log in with your account. If you do not want the Linux newuser user, run the **/usr/sbin/userdel -r newuser** command as the Linux root user to remove it, along with the Linux newuser's home directory.

Sia gli utenti di Linux confinati sia quelli non confinati sono sottoposti a controlli per operazioni in memoria, come scrivere o eseguire codice, oltre alle restrizioni di MCS (ed MLS, se la politica MLS è in uso). Se gli utenti Linux non confinati eseguono un'applicazione che la politica di SELinux definisce che può transitare dal dominio **unconfined_t** al proprio dominio confinato, allora anche gli utenti vengono sottoposti alle restrizioni di quel dominio. Il beneficio di sicurezza che ne deriva è che anche se un utente di Linux è non confinato, l'applicazione continua a rimanere confinata perciò l'uso di una falla nell'applicazione verrebbe limitata dalla policy. Nota: ciò tuttavia non protegge il sistema dall'utente. Invece, l'utente ed il sistema sono protetti da possibili danni causati da una falla nell'applicazione.

In Fedora 13 sono disponibili i seguenti utenti di SELinux confinati:

Utente	Dominio	Sistema X Window	su e sudo	Esecuzione in home directory e in /tmp	Rete
guest_u	guest_t	no	no	optional	no
xguest_u	xguest_t	yes	no	optional	only **Firefox**
user_u	user_t	yes	no	optional	yes
staff_u	staff_t	yes	only **sudo**	optional	yes

Tabella 4.1. Proprietà degli Utenti di SELinux

- Gli utenti di Linux nei domini **guest_t**, **xguest_t**, e **user_t**, possono eseguire applicazioni user ID (setuid) solo se permesso dalla policy di SELinux. Essi non possono eseguire applicazioni setuid come **su** e **/usr/bin/sudo**, perciò non possono diventare utente Linux root.
- Gli utenti nel dominio **guest_t** non possono accedere alla rete, e possono avviare una sessione solo da terminale (incluso `ssh`, ma non possono utilizzare `ssh` per potersi connettere ad un altro sistema).
- Gli utenti di Linux nel dominio **xguest_t** possono accedere alla rete soltanto per connettersi al web attraverso **Firefox**.
- Gli utenti di Linux nei domini **xguest_t**, **user_t** e **staff_t**, possono avviare una sessione sia da terminale sia dal sistema grafico X Windows.
- Per impostazione predefinita, gli utenti di Linux nel dominio **staff_t** non possono eseguire applicazioni setuid, tramite **/usr/bin/sudo**. Tali permessi devono essere configurati da un amministratore.

By default, Linux users in the **guest_t** and **xguest_t** domains can not execute applications in their home directories or **/tmp/**, preventing them from executing applications (which inherit users' permissions) in directories they have write access to. This helps prevent flawed or malicious applications from modifying files users' own.

Per impostazione predefinita, gli utenti di Linux nei domini **user_t** e **staff_t**, possono eseguire applicazioni nella loro home directory ed in **/tmp/**. Per informazioni su come permettere o negare agli utenti di eseguire applicazioni nella propria home directory e in **/tmp/**, fare riferimento a *Sezione 6.6, «Booleane per gli utenti che eseguono applicazioni»*.

Lavorare con SELinux

Nelle seguenti sezioni si darà una breve introduzione ai principali pacchetti di SELinux distribuiti con Fedora; come installare e aggiornare i pacchetti; quali log file sono utilizzati; il principale file di configurazione di SELinux; come abilitare e disabilitare SELinux; i modi di esecuzione di SELinux; come configurare le booleane; come cambiare in modo temporaneo o permanente le etichette a file e directory; come scavalcare i contesti dei file system usando il comando **mount**; come montare i file system NFS; e come preservare i contesti di SELinux quando si copiano file e directory.

5.1. I pacchetti di SELinux

In Fedora, the SELinux packages are installed by default in a full installation, unless they are manually excluded during installation. If performing a minimal installation in text mode, the *policycoreutils-python* package will not be installed by default. Also, by default, SELinux targeted policy is used, and SELinux runs in enforcing mode. The following is a brief description of the main SELinux packages:

policycoreutils-python: provides utilities such as **semanage**, **audit2allow**, **audit2why** and **chcat**, for operating and managing SELinux.

policycoreutils: provides utilities such as **restorecon**, **secon**, **setfiles**, **semodule**, **load_policy**, and **setsebool**, for operating and managing SELinux.

policycoreutils-gui: fornisce **system-config-selinux**, uno strumento grafico per gestire SELinux.

selinux-policy: fornisce la policy di riferimento di SELinux. La policy di riferimento è una policy di SELinux completa, ed è usata come base per altre policy, come la targeted policy. Per ulteriori informazioni fare riferimento a *SELinux Reference Policy*[1] sviluppata da Tresys Technology. Il pacchetto *selinux-policy-devel*, fornisce alcuni strumenti di sviluppo, come **/usr/share/selinux/devel/policygentool** e **/usr/share/selinux/devel/policyhelp**, oltre ad alcuni file di esempi di policy. Questo pacchetto fu assorbito nel pacchetto *selinux-policy*.

selinux-policy-nome-policy: fornisce le policy di SELinux. Per una targeted policy, installare *selinux-policy-targeted*. Per usare MLS, installare *selinux-policy-targeted*. A partire da Fedora 8, la rigida policy venne assorbita in una targeted policy, per permettere ad utenti confinati e non confinati di coesistere nello stesso sistema.

setroubleshoot-server: traduce i messaggi di divieto, generati da SELinux, in descrizioni dettagliate che possono essere analizzate con **sealert** (fornito con questo pacchetto).

setools, *setools-gui*, and *setools-console*: these packages provide the *Tresys Technology SETools distribution*[2], a number of tools and libraries for analyzing and querying policy, audit log monitoring and reporting, and file context management[3]. The *setools* package is a meta-package for SETools. The *setools-gui* package provides the **apol**, **seaudit**, and **sediffx** tools. The *setools-console* package provides the **seaudit-report**, **sechecker**, **sediff**, **seinfo**, **sesearch**, **findcon**, **replcon**, and **indexcon** command line tools. Refer to the *Tresys Technology SETools*[4] page for information about these tools.

libselinux-utils: fornisce gli strumenti **avcstat**, **getenforce**, **getsebool**, **matchpathcon**, **selinuxconlist**, **selinuxdefcon**, **selinuxenabled**, **setenforce**, e **togglesebool**.

[1] http://oss.tresys.com/projects/refpolicy
[2] http://oss.tresys.com/projects/setools
Brindle, Joshua. "Re: blurb for fedora setools packages" Email to Murray McAllister. 1 November 2008. Any edits or changes in this version were done by Murray McAllister.
[4] http://oss.tresys.com/projects/setools

mcstrans: traduce i livelli, come **s0-s0:c0.c1023**, in un formato mnemonico, come **SystemLow-SystemHigh**. Non è installato per impostazione predefinita.

Per installare i pacchetti in Fedora, come utente Linux root, eseguire il comando **yum install *nome-del-pacchetto***. Per esempio, per installare il pacchetto *mcstrans*, **yum install mcstrans**. Per aggiornare un pacchetto, usare **yum update**.

Per maggiori informazioni sull'uso di **yum** per gestire i pacchetti, fare riferimento a *Managing Software with yum*[56].

Nota

In versioni precedenti di Fedora, per creare un modulo di policy locale usando **audit2allow -M**, è richiesto il pacchetto *selinux-policy-devel*.

5.2. File usati per registrare i messaggi di SELinux

In Fedora 13, i pacchetti *dbus*, *setroubleshoot-server* e *audit* vengono installati se non sono stati rimossi dalla selezione di pacchetti predefinita.

I messaggi di divieto di SELinux, come il seguente, sono registrati in **/var/log/audit/audit.log**, per impostazione predefinita:

```
type=AVC msg=audit(1223024155.684:49): avc:  denied  { getattr } for  pid=2000 comm="httpd"
 path="/var/www/html/file1" dev=dm-0 ino=399185 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=system_u:object_r:samba_share_t:s0 tclass=file
```

Inoltre, se `setroubleshootd` è in esecuzione, i messaggi di divieto in **/var/log/audit/audit.log** sono tradotti in un formato più comprensibile ed inviati al file **/var/log/messages**:

```
May  7 18:55:56 localhost setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr"
 to /var/www/html/file1 (samba_share_t). For complete SELinux messages. run sealert -l
 de7e30d6-5488-466d-a606-92c9f40d316d
```

In Fedora 13, `setroubleshootd` no longer constantly runs as a service, however it is still used to analyze the AVC messages. Two new programs act as a method to start setroubleshoot when needed: `sedispatch` and `seapplet`. `sedispatch` runs as part of the audit subsystem, and via `dbus`, sends a message when an AVC denial occurs, which will go straight to `setroubleshootd` if it is already running, or it will start `setroubleshootd` if it is not running. `seapplet` is a tool which runs in the system's toolbar, waiting for dbus messages in `setroubleshootd`, and will launch the notification bubble, allowing the user to review the denial.

A seconda dei demoni in esecuzione, i messaggi di divieto sono inviati su diversi file:

Demone	Percorso dei log
auditd on	**/var/log/audit/audit.log**
auditd off; rsyslogd on	**/var/log/messages**

[5] http://docs.fedoraproject.org/yum/en/
"Managing Software with yum", ideato da Stuart Ellis, e scritto da Paul W. Frields, Rodrigo Menezes, e Hugo Cisneiros.

Demone	Percorso dei log
rsyslogd and auditd on	**/var/log/audit/audit.log**. Dei messaggi di diniego più facilmente leggibili sono inviati anche a **/var/log/messages**

Avvio automatico dei demoni

Per configurare i demoni `auditd`, `rsyslogd`, e `setroubleshootd` ad avviarsi automaticamente al boot, eseguire i seguenti comandi come utente root:

```
/sbin/chkconfig --levels 2345 auditd on
```

```
/sbin/chkconfig --levels 2345 rsyslog on
```

Per verificare lo stato dei servizi, usare il comando **service *nome-del-del-servizio* status**:

```
$ /sbin/service auditd status
auditd (pid  1318) is running...
```

Se i servizi non sono in esecuzione (***nome-del servizio* è interrotto**), usare il comando **service *nome-del-servizio* start**, come utente root, per avviarli. Per esempio:

```
# /sbin/service auditd start
Starting auditd:                                   [  OK  ]
```

5.3. File di configurazione principale

Il file **/etc/selinux/config** è il file di configurazione principale di SELinux. Esso controlla la modalità di esecuzione di SELinux e la policy da usare:

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

SELINUX=enforcing

L'opzione **SELINUX**imposta la modalità di esecuzione. In SELinux esistono tre modalità: enforcing, permissive e disabilitato. Quando si usa la modalità enforcing, viene imposta la policy di SELinux, vietando gli accessi secondo le regole di policy. I messaggi di divieto sono registrati. Quando si usa la modalità permissive, SELinux non vieta il permesso d'accesso a certe azioni od operazioni, ma quelle azioni od operazioni che sarebbero state vietate se SELinux fosse stato in modalità enforcing, ora sono semplicemente registrate. Quando si usa la modalità disabilitato, SELinux

non è in esecuzione (il modulo SELinux non risulta registrato nel kernel), e sono usate soltanto le regole DAC.

SELINUXTYPE=targeted
L'opzione **SELINUXTYPE** imposta la policy di SELinux da usare. La targeted policy è quella predefinito. Cambiare questa opzione soltanto se si vuole usare la policy di MLS. Per usare MLS occorre installare il pacchetto *selinux-policy-mls*; impostare **SELINUXTYPE=mls** in **/etc/selinux/config** e riavviare il sistema.

Importante
Nei sistemi in cui SELinux è in esecuzione in modalità permissive o disabilitato, gli utenti possono contrassegnare i file con contesti di sicurezza sbagliati. Inoltre, i file creati con SELinux disabilitato, sono privi di contesto di sicurezza. Ciò è fonte di problemi quando si cambia in modalità enforcing. Cambiando modalità di esecuzione di SELinux da disabilitato a permissive o enforcing, per prevenire entrambi i problemi citati, i file sono automaticamente attribuiti ai contesti di sicurezza.

5.4. Abilitare e disabilitare SELinux

Usare il comando **/usr/sbin/getenforce** o **/usr/sbin/sestatus** per verificare lo stato di SELinux. Il comando **getenforce** restituisce **Enforcing**, **Permissive**, o **Disabilitato**. Il comando **getenforce** restituisce **Enforcing** se SELinux è abilitato (sono imposte le regole di policy di SELinux):

```
$ /usr/sbin/getenforce
Enforcing
```

Il comando **getenforce** restituisce **Permissive** quando SELinux è abilitato, ma non sono imposte le sue regole di policy, restando valide solo le regole DAC. Se SELinux è disabilitato, il comando **getenforce** restituisce **Disabilitato**.

Il comando **sestatus** restituisce lo stato di SELinux e la policy in uso:

```
$ /usr/sbin/sestatus
SELinux status:                 enabled
SELinuxfs mount:                /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy version:                 23
Policy from config file:        targeted
```

SELinux status: enabled viene restituito quando SELinux è abilitato. **Current mode: enforcing** viene restituito quando SELinux è in esecuzione in modalità enforcing. **Policy from config file: targeted** viene restituito quando viene utilizzata la targeted policy di SELinux.

5.4.1. Abilitare SELinux

Nei sistemi in cui SELinux è disabilitato, nel file **/etc/selinux/config** si trova l'opzione **SELINUX=disabled**:

```
# 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=disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Inoltre, il comando **`getenforce`** restituisce **`Disabilitato`**:

```
$ /usr/sbin/getenforce
Disabled
```

Per abilitare SELinux:

1. Usare i comandi **`rpm -qa | grep selinux`**, **`rpm -q policycoreutils`**, ed **`rpm -qa | grep setroubleshoot`**, per verificare che i pacchetti di SELinux sono installati. Questa guida suppone che siano installati i pacchetti *selinux-policy-targeted*, *selinux-policy*, *libselinux*, *libselinux-python*, *libselinux-utils*, *policycoreutils*, *setroubleshoot*, *setroubleshoot-server*, *setroubleshoot-plugins*. Se non sono installati, come utente root, usare il comando **`yum install nome-del-pacchetto`**. I seguenti sono opzionali: *policycoreutils-gui*, *setroubleshoot*, *selinux-policy-devel*, e *mcstrans*.

2. Prima di abilitare SELinux, occorre garantire che ogni file abbia un'etichetta di contesto di SELinux, Tenendo presente che i domini confinati potrebbero avere un divieto d'accesso, impedendo al sistema di avviarsi correttamente, configurare **`SELINUX=permissive`** in **`/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.
   SELINUX=permissive
   # SELINUXTYPE= can take one of these two values:
   #       targeted - Targeted processes are protected,
   #       mls - Multi Level Security protection.
   SELINUXTYPE=targeted
   ```

3. Eseguire, come root, il comando **`reboot`** per riavviare il sistema. Al successivo riavvio, i file avranno una etichetta (o contesto di sicurezza). Un processo specializzato assegnerà a ciascun file un contesto di SELinux:

   ```
   *** Warning -- SELinux targeted policy relabel is required.
   *** Relabeling could take a very long time, depending on file
   *** system size and speed of hard drives.
   ****
   ```

   Ogni carattere **`*`** in basso rappresenta 1000 file già forniti di etichetta. Nell'esempio precedente, quattro caratteri **`*`** stanno ad indicare che 4000 file hanno il proprio contesto di sicurezza. Il tempo

richiesto per etichettare tutti i file, dipende dal numero di file presenti nel sistema e dalla velocità degli hard disks. Nei sistemi più moderni, ciò può richiedere all'incirca 10 minuti.

4. In permissive mode, SELinux policy is not enforced, but denials are still logged for actions that would have been denied if running in enforcing mode. Before changing to enforcing mode, as the Linux root user, run the **`grep "SELinux is preventing" /var/log/messages`** command as the Linux root user to confirm that SELinux did not deny actions during the last boot. If SELinux did not deny actions during the last boot, this command does not return any output. Refer to *Capitolo 7, Risoluzione dei problemi* for troubleshooting information if SELinux denied access during boot.

5. Se non ci sono messaggi di divieto in **`/var/log/messages`**, configurare **`SELINUX=enforcing`** in **`/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.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

6. Riavviare il sistema. Dopo il riavvio,verificare che **`getenforce`** restituisce **`Enforcing`**:

```
$ /usr/sbin/getenforce
Enforcing
```

7. Come utente root, eseguire **`/usr/sbin/semanage login -l`** per vedere gli utenti di Linux applicati agli utenti di SELinux. Il risultato sarà simile a:

```
Login Name                SELinux User              MLS/MCS Range

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

Se è il risultato è diverso, eseguire come root, i seguenti comandi per risolvere l'applicazione tra utenti di Linux e utenti di SELinux. Si consiglia di ignorare, se presente, l'avviso **`Utente di SELinux nome-utente is already defined`**, dove *`nome-utente`* può essere **`unconfined_u`**, **`guest_u`**, o **`xguest_u`**:

1. ```
   /usr/sbin/semanage user -a -S targeted -P user -R "unconfined_r system_r" -r s0-
   s0:c0.c1023 unconfined_u
   ```

2. ```
   /usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r s0-s0:c0.c1023 __default__
   ```

3.
```
/usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r s0-s0:c0.c1023 root
```

4.
```
/usr/sbin/semanage user -a -S targeted -P user -R guest_r guest_u
```

5.
```
/usr/sbin/semanage user -a -S targeted  -P user -R xguest_r xguest_u
```

Importante

Nei sistemi in cui SELinux è in esecuzione in modalità permissive o disabilitato, gli utenti possono contrassegnare i file con contesti di sicurezza sbagliati. Inoltre, i file creati con SELinux disabilitato, sono privi di contesto di sicurezza. Ciò è fonte di problemi quando si cambia in modalità enforcing. Cambiando modalità di esecuzione di SELinux da disabilitato a permissive o enforcing, per prevenire entrambi i problemi citati, i file sono automaticamente attribuiti ai contesti di sicurezza.

5.4.2. Disabilitare SELinux

Per disabilitare SELinux, impostare **`SELINUX=disabled`** in **`/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.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Riavviare il sistema. Dopo di che, verificare che **`getenforce`** restituisce **`Disabilitato`**:

```
$ /usr/sbin/getenforce
Disabled
```

5.5. Modalità di SELinux

SELinux ha tre modalità:

- Enforcing: la policy di SELinux è imposta. SELinux vieta gli accessi in base alle sue regole di policy.
- Permissive: la policy di SELinux non è imposta. SELinux non vieta l'accesso, ma vengono registrati i messaggi di divieto di quelle azioni che sarebbero vietate se SELinux fosse stato in modalità enforcing.
- Disabilitato: Selinux è disabilitato. Sono usate soltanto le regole DAC.

Usare **/usr/sbin/setenforce** per modificare il modo di esecuzione di SELinux da enforcing a permissive, o viceversa. Le modifiche fatte usando **/usr/sbin/setenforce** non persistono al riavvio del sistema. Per cambiare in modalità enforcing, come root, eseguire il comando **/usr/sbin/setenforce 1**. Per cambiare in modalità permissive, usare **/usr/sbin/setenforce 0**. Per vedere la modalità corrente, usare **/usr/sbin/getenforce**.

Informazioni su come rendere persistenti i cambiamenti sono elencate in *Sezione 5.4, «Abilitare e disabilitare SELinux»*.

5.6. Booleane

Le booleane consentono di modificare al runtime, parti della policy di SELinux, facilitando la gestione della policy. Ciò permette degli adattamenti, come consentire il permesso a quei servizi che devono accedere al file system NFS, senza dover ricaricare o ricompilare l'intera policy di SELinux.

5.6.1. Listare le booleane

Per una lista di booleane, una spiegazione di ciò che rappresentano, e per vedere se esse sono nello stato on od off, eseguire, come root, il comando **semanage boolean -l**. L'esempio seguente riporta una breve lista di booleane:

```
# /usr/sbin/semanage boolean -l
SELinux boolean                          Description

ftp_home_dir                   -> off   Allow ftp to read and write files in the user home
 directories
xen_use_nfs                    -> off   Allow xen to manage nfs files
xguest_connect_network         -> on    Allow xguest to configure Network Manager
```

La colonna **Booleana di SELinux** lista i nomi delle booleane. La colonna **Descrizione** visualizza lo stato on od off ed il suo impiego.

Nell'esempio, la booleana **ftp_home_dir** è off, per impedire al demone FTP (`vsftpd`) di leggere o scrivere nella home directory degli utenti:

```
ftp_home_dir                   -> off   Allow ftp to read and write files in the user home
 directories
```

Il comando **getsebool -a** lista le booleane, visualizzando se esse sono on od off, senza alcuna descrizione. Ecco un esempio:

```
$ /usr/sbin/getsebool -a
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

Eseguire il comando **getsebool *nome-booleana***, per verificare lo stato della booleana *nome-booleana*:

```
$ /usr/sbin/getsebool allow_console_login
allow_console_login --> off
```

Usare una lista di nomi separati da spazio,per visualizzare lo stato di più booleane:

```
$ getsebool allow_console_login allow_cvs_read_shadow allow_daemons_dump_core
allow_console_login --> off
allow_cvs_read_shadow --> off
allow_daemons_dump_core --> on
```

5.6.2. Configurare le booleane

Il comando **setsebool *nome-booleana x*** serve per impostare lo stato delle booleane, dove *nome-booleana* è il suo nome, e *x* è **on** per attivare la booleana, o **off** per disattivarla.

L'esempio seguente mostra come configurare la booleana **httpd_can_network_connect_db**:

1. Per impostazione predefinita, la booleana **httpd_can_network_connect_db** è off, impedendo agli scripts e ai moduli del Server HTTP Apache di connettersi ai server di database:

   ```
   $ /usr/sbin/getsebool httpd_can_network_connect_db
   httpd_can_network_connect_db --> off
   ```

2. Per autorizzare temporaneamente, gli scripts e i moduli del Server HTTP Apache a connettersi ai server di databases, eseguire, come root, **setsebool httpd_can_network_connect_db on**:

3. Usare il comando **getsebool httpd_can_network_connect_db** per verificare che la booleana è on:

   ```
   $ /usr/sbin/getsebool httpd_can_network_connect_db
   httpd_can_network_connect_db --> on
   ```

   Ora gli scripts e i moduli del Server HTTP Apache possono connettersi ai server di database.

4. Questa modifica non persiste al successivo riavvio del sistema. Per rendere persistenti le modifiche apportate, eseguire come root il comando **setsebool -P *nome-booleana* on**:

   ```
   # /usr/sbin/setsebool -P httpd_can_network_connect_db on
   ```

5. Per passare, temporaneamente, all'impostazione predefinito, eseguire come root **setsebool httpd_can_network_connect_db off**. Per rendere i cambiamenti persistenti al successivo riavvio, eseguire **setsebool -P httpd_can_network_connect_db off**.

5.6.3. Booleane per NFS e CIFS

I file system NFS montati sul lato client sono etichettati con un contesto predefinito dalla policy per file system NFS. Comunemente nelle politiche, questo contesto predefinito usa il tipo **nfs_t**. Anche le condivisioni del server Samba, montate sul lato client sono etichettate con un contesto predefinito definito da una policy. Comunemente nelle politiche, questo contesto predefinito usa il tipo **cifs_t**.

Dipendendo dalla configurazione della policy, alcuni servizi possono non essere in grado di leggere i file di tipo **nfs_t** o **cifs_t**. Di fatto, ciò impedisce ai file systems etichettati con questi tipi di essere

montati e letti o esportati da altri servizi. Usando le booleane è possibile controllare quali servizi possono avere accesso ai tipi **nfs_t** e **cifs_t**.

I comandi **setsebool** e **semanage** devono essere eseguiti come root. Il comando **setsebool -P** rende le modifiche persistenti. Non usare l'opzione **-P** se non si desidera rendere persistenti le modifche al successivo riavvio del sistema:

Server HTTP Apache

Per consentire l'accesso ai file systems NFS (di tipo **nfs_t**):

```
/usr/sbin/setsebool -P httpd_use_nfs on
```

Per consentire l'acceso ai file systems Samba (di tipo /usr/sbin/setsebool -P httpd_use_nfs on):

```
/usr/sbin/setsebool -P httpd_use_cifs on
```

Samba

Per esportare i file systems NFS:

```
/usr/sbin/setsebool -P samba_share_nfs on
```

FTP (vsftpd)

Per consentire l'accesso al file systems NFS:

```
/usr/sbin/setsebool -P allow_ftpd_use_nfs on
```

Per consentire l'accesso ai file systems Samba:

```
/usr/sbin/setsebool -P allow_ftpd_use_cifs on
```

Altri Servizi

Per una lista di booleane relative ad altri servizi di NFS:

```
/usr/sbin/semanage boolean -l | grep nfs
```

Per una lista di booleane relative ad altri servizi di Samba:

```
/usr/sbin/semanage boolean -l | grep cifs
```

Nota

Queste booleane esistono nella policy di SELinux distribuita con Fedora 13. Esse potrebbero essere assenti in altre versioni di Fedora o in altri sistemi operativi.

Refer to the SELinux Managing Confined Services Guide at *http://docs.fedoraproject.org* for more information regarding SELinux Booleans.

5.7. Contesti di SELinux - Etichettare i file

Nei sistemi in cui SELinux è in esecuzione, tutti i processi e file sono etichettati in modo da possedere un informazione rilevante la sicurezza. Questa informazione si chiama contesto di SELinux. Per i file, il contesto può essere visto con il comando **ls -Z**:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

Nell'esempio, SELinux fornisce un utente (**unconfined_u**), un ruolo (**object_r**), un tipo (**user_home_t**), ed un livello (**s0**). Tale informazione è usata per prendere decisioni di controllo sugli accessi. Sui sistemi DAC, l'accesso è controllato in base agli IDs degli utenti e di gruppo. Le regole di policy di SELinux vengono verificate dopo aver verificato le regole di DAC. Le regole di SELinux non vengono verificate se, in primis, sono violate le regole DAC.

Sono disponibili diversi comandi per gestire il contesto di SELInux per i file, come **chcon**, **semanage fcontext**, e **restorecon**.

5.7.1. Cambiamenti temporanei:chcon

Il comando **chcon** cambia il contesto di SELinux per i file. Questi cambiamenti fatti con il comando **chcon** non persistono ad una rietichettatura del file system, o all'esecuzione del comando **/sbin/restorecon**. La policy di SELinux controlla se gli utenti hanno il diritto di modificare il contesto di SELinux per un certo file. Quando si usa il comando **chcon**, gli utenti possono modificare una parte o un intero contesto di SELinux. Una tipizzazione sbagliata dei file, è una causa comune di divieto d'accesso da parte di SELinux.

Riferimento rapido

- Eseguire il comando **chcon -t *tipo nome-file***, per cambiare il tipo ad un file, dove *tipo* è un nome di tipo come **httpd_sys_content_t**, e *nome-file* è un file o una directory.
- Eseguire il comando **chcon -R -t *tipo nome-directory***, per cambiare il tipo ad una directory ed al suo contenuto, dove *tipo* è un nome di tipo come **httpd_sys_content_t**, e *nome-directory* è una directory.

Changing a File's or Directory's Type

L'esempio seguente, mostra solo come cambiare il tipo, e nessun altro attributo del contesto di SELinux:

1. Eseguire il comando **cd** senza argomenti per spostarsi nella propria home directory.
2. Eseguire il comando **touch file1** per creare un nuovo file. Usare il comando **ls -Z file1** per vedere il contesto di SELinux relativo a **file1**:

   ```
   $ ls -Z file1
   -rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
   ```

   In questo esempio, il contesto per **file1** include l'utente **unconfined_u** di SELinux, il ruolo **object_r**, il tipo **il tipo user_home_t**, ed il livello **s0**. Per una descrizione di ogni elemento del contesto, fare riferimento a *Capitolo 3, Contesti di SELinux*.
3. Eseguire il comando **chcon -t samba_share_t file1** per assegnare al file il tipo **samba_share_t**.L'opzione **-t** modifica soltanto il tipo. Per verificare le modifiche apportate **ls -Z file1**:

   ```
   $ ls -Z file1
   ```

```
-rw-rw-r--  user1 group1 unconfined_u:object_r:samba_share_t:s0 file1
```

4. Usare il comando **/sbin/restorecon -v file1** per ripristinare il contesto di SELinux, per **file1**. Per vedere ciò che cambia usare l'opzione **-v**:

```
$ /sbin/restorecon -v file1
restorecon reset file1 context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:user_home_t:s0
```

   Nell'esempio, il precedente tipo **samba_share_t** è sostituito con il tipo **user_home_t** corretto. Quando si usa la targeted policy (che è quella di default in Fedora 11), il comando **/sbin/restorecon** legge i file nella directory **/etc/selinux/targeted/contexts/files/**, per vedere quale contesto devono avere i file.

L'esempio è analogo per le direcories, cioè se **file1** è una directory.

Modificare il tipo ad una directory ed al suo contenuto.

The following example demonstrates creating a new directory, and changing the directory's file type (along with its contents) to a type used by the Apache HTTP Server. The configuration in this example is used if you want Apache HTTP Server to use a different document root (instead of **/var/www/html/**):

1. Come utente root, eseguire il comando **mkdir /web** per creare una nuova directory, e successivamente **touch /web/file{1,2,3}** per creare 3 file vuoti (**file1**, **file2**, e **file3**). La directory **/web/** e i suoi file sono di tipo **default_t**:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

2. Come root, eseguire **chcon -R -t httpd_sys_content_t /web/** per assegnare il tipo **httpd_sys_content_t** alla directory **/web/** (ed al suo contenuto):

```
# chcon -R -t httpd_sys_content_t /web/
# ls -dZ /web/
drwxr-xr-x  root root unconfined_u:object_r:httpd_sys_content_t:s0 /web/
# ls -lZ /web/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

3. Come root, eseguire **/sbin/restorecon -R -v /web/** per ripristinare i contesti di SELinux predefinito:

```
# /sbin/restorecon -R -v /web/
restorecon reset /web context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
```

```
restorecon reset /web/file2 context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
restorecon reset /web/file3 context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
restorecon reset /web/file1 context unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
```

Refer to the chcon(1) manual page for further information about **chcon**.

Nota

In SELinux, Type Enforcement è il principale controllo dei permessi usato in una targeted policy. Per gran parte, gli utenti SELinux e le sue regole possono essere ignorate.

5.7.2. Modifiche persistenti: semanage fcontext

Il comando **/usr/sbin/semanage fcontext** cambia il contesto di SELinux per i file. Quando si usa la targeted policy, le modifiche fatte usando questo comando sono aggiunte al file **/etc/selinux/targeted/contexts/files/file_contexts** se le modifiche riguardano file che esistono in **file_contexts**, o aggiunte al file **file_contexts.local** per nuovi file e directory, come la directory **/web/**, appena creata. Il comando **setfiles**, usato quando si etichetta un file system, ed il comando **/sbin/restorecon** che ripristina i contesti di default, leggono questi file. Quindi, i cambiamenti apportati usando **/usr/sbin/semanage fcontext** sono persistenti, anche se il file system viene rietichettato. La policy di SELinux controlla se gli utenti possono modificare il contesto per qualsiasi file.

Riferimento rapido

Per fare dei cambiamenti al contesto di SELinux che siano persistenti ad una rietichettatura del file system:

1. Eseguire il comando **/usr/sbin/semanage fcontext -a *opzioni nome-file|nome-directory***, ricordando di usare il percorso completo del file o della directory.
2. Eseguire il comando **/sbin/restorecon -v *nome-file|nome-directory*** per applicare le modifiche al contesto.

Changing a File's Type

The following example demonstrates changing a file's type, and no other attributes of the SELinux context:

1. Come utente root, eseguire **touch /etc/file1**, per creare un nuovo file. Per impostazione predefinita il nuovo file creato nella directory **/etc/** sarà etichettato con il tipo **etc_t**:

```
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0       /etc/file1
```

2. Come root, eseguire il comando **/usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1**, per modificare tipo a **file1**, assegnandoli il tipo **samba_share_t**. L'opzione **-a** aggiunge un nuovo record, e l'opzione **-t** definisce un tipo

(**samba_share_t**). Nota: eseguire solo questo comando non cambia direttamente il tipo al file - **file1** è tuttora del tipo **etc_t**:

```
# /usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0       /etc/file1
```

Il comando **/usr/sbin/semanage fcontext -a -t samba_share_t /etc/file1** aggiunge al file **/etc/selinux/targeted/contexts/files/file_contexts.local**, la seguente riga:

```
/etc/file1    unconfined_u:object_r:samba_share_t:s0
```

3. Come root, eseguire il comando **/sbin/restorecon -v /etc/file1** per modificare il tipo. Poiché il comando **semanage** ha aggiunto **/etc/file1** in **file.contexts.local**, il comando **/sbin/restorecon** modificherà il tipo in **samba_share_t**:

```
# /sbin/restorecon -v /etc/file1
restorecon reset /etc/file1 context unconfined_u:object_r:etc_t:s0-
>system_u:object_r:samba_share_t:s0
```

4. Come root, eseguire il comando **rm -i /etc/file1**, per cancellare **file1**.

5. Come utente linux root, eseguire il comando **/usr/sbin/semanage fcontext -d /etc/file1**per rimuovere il contesto aggiunto per **/etc/file1**. Quando il contesto è rimosso, eseguendo **restorecon**, il tipo da **samba_share_t**.si modificherà in **etc_t**.

Changing a Directory's Type

The following example demonstrates creating a new directory and changing that directory's file type, to a type used by Apache HTTP Server:

1. Come utente root, eseguire il comando **mkdir /web** per creare una nuova directory. Questa directory sarà etichettata con il tipo **default_t**:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
```

Il comando **ls** con l'opzione **-d** fa elencare ad **ls** le informazioni sulla directory, invece del suo contenuto, e l'opzione **-Z** fa visualizzare ad **ls** il conetsto di SELinux (nell'esempio, **unconfined_u:object_r:default_t:s0**).

2. Come utente root, eseguire il comando **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web** per modificare **/web/** al tipo **httpd_sys_content_t**. L'opzione **-a** aggiunge un nuovo record, e l'opzione **-t** definisce un tipo (**httpd_sys_content_t**). Nota: eseguire solo questo comando non modifica direttamente il tipo - **/web/** è tuttora etichettato con il tipo **default_t**:

```
# /usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web
# ls -dZ /web
```

```
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0   /web
```

Il comando **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /web** aggiunge la seguente riga al file **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/web    unconfined_u:object_r:httpd_sys_content_t:s0
```

3. Come root, eseguire il comando **/sbin/restorecon -v /web** per modificare il tipo. Poiché **semanage** ha inserito **/web** in **file.contexts.local**, il comando **/sbin/restorecon** modifica il tipo in **httpd_sys_content_t**:

```
# /sbin/restorecon -v /web
restorecon reset /web context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Per impostazione predefinita, i nuovi file e directory ereditano il tipo SELinux posseduto dalle directory padre. Nel caso in esame, e prima di rimuovere il contesto aggiunto per **/web/**, i file e le directory create in **/web/** saranno etichettate con il tipo **httpd_sys_content_t**.

4. Come root, eseguire il comando **/usr/sbin/semanage fcontext -d /web** per eliminare il contesto di SELinux aggiunto per **/web/**.

5. Come root, eseguire il comando **/sbin/restorecon -v /web** per ripristinare il contesto di SELinux predefinito.

Modificare il tipo ad una directory ed al suo contenuto.

The following example demonstrates creating a new directory, and changing the directory's file type (along with its contents) to a type used by Apache HTTP Server. The configuration in this example is used if you want Apache HTTP Server to use a different document root (instead of **/var/www/html/**):

1. Come utente root, eseguire il comando **mkdir /web** per creare una nuova directory, e successivamente **touch /web/file{1,2,3}** per creare 3 file vuoti (**file1**, **file2**, e **file3**). La directory **/web/** e i suoi file sono di tipo **default_t**:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

2. As the Linux root user, run the **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"** command to change the type of the **/web/** directory and the files in it, to **httpd_sys_content_t**. The **-a** option adds a new record, and the **-t** option defines a type (httpd_sys_content_t). The **"/web(/.*)?"** regular expression causes the **semanage** command to apply changes to the **/web/** directory, as well as the files in it. Note:

running this command does not directly change the type - **/web/** and files in it are still labeled with the **default_t** type:

```
# ls -dZ /web
drwxr-xr-x  root root unconfined_u:object_r:default_t:s0 /web
# ls -lZ /web
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:default_t:s0 file3
```

The **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"** command adds the following entry to **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/web(/.*)?    system_u:object_r:httpd_sys_content_t:s0
```

3. Come root, eseguire il comando **/sbin/restorecon -R -v /web** per modificare il tipo alla directory **/web/** ed ai suoi file. L'opzione **-R** è per la ricorsione, essa applica il tipo **httpd_sys_content_t** a tutti i file e a tutte le directory contenute in **/web/**. Poiché **semanage** ha aggiunto **/web(/.*)?** in **file.contexts.local**, il comando **/sbin/restorecon** modifica il tipo in **httpd_sys_content_t**:

```
# /sbin/restorecon -R -v /web
restorecon reset /web context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file2 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file3 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/file1 context unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Per impostazione predefinita i nuovi file e directory ereditano il tipo di SELinux dei loro genitori. In questo esempio, i file e le directory create in **/web/** saranno etichettate con il tipo **httpd_sys_content_t**.

4. As the Linux root user, run the **/usr/sbin/semanage fcontext -d "/web(/.*)?"** command to remove the context added for **"/web(/.*)?"**.

5. Come root, eseguire il comando **/sbin/restorecon -R -v /web** per ripristinare il contesto predefinito.

Eliminare un contesto aggiunto

L'esempio seguente mostra come aggiungere e rimuovere un contesto di SELinux:

1. Come utente root, eseguire il comando **/usr/sbin/semanage fcontext -a -t httpd_sys_content_t /test**. La directory **/test/** non deve esistere. Questo comando aggiunge il contesto seguente a **/etc/selinux/targeted/contexts/files/file_contexts.local**:

```
/test    system_u:object_r:httpd_sys_content_t:s0
```

2. Per rimuovere il contesto, come utente Linux root, eseguire il comando **/usr/sbin/semanage fcontext -d *nome-file|nome-directory***, dove *nome-file*|*nome-directory* è la parte iniziale in **file_contexts.local**. Il seguente è un esempio di contesto in **file_contexts.local**:

```
/test    system_u:object_r:httpd_sys_content_t:s0
```

   Divenendo la parte iniziale **/test**. Per impedire alla directory **/test/** di essere rietichettata con **httpd_sys_content_t** dopo aver lanciato il comando **/sbin/restorecon** o a causa di una rietichettatura del file system, eseguire come utente Linux root, il seguente comando per eliminare il contesto da **file_contexts.local**:

   /usr/sbin/semanage fcontext -d /test

Se il contesto fa parte di una espressione regolare, per esempio **/web(/.*)?**, usare i doppi apici intorno all'espressione:

/usr/sbin/semanage fcontext -d "/web(/.*)?"

Refer to the semanage(8) manual page for further information about **/usr/sbin/semanage**.

Importante

Quando si modifica il contesto di SELinux con **/usr/sbin/semanage fcontext -a**, usare il percorso completo ai file e le directory, per evitare che i file vengano mal etichettati dopo una rietichettatura del file system, o dopo aver eseguito il comando **/sbin/restorecon**.

5.8. I tipi file_t e default_t

Nei file systems che supportano gli attributi estesi, quando si accede ad un file privo di un contesto di SELinux, esso viene trattato come se avesse un contesto predefinito. Comunemente nelle policy, questo contesto predefinito usa il tipo **file_t**. Questo dovrebbe essere l'unico impiego di questo tipo, affinché quei file senza un contesto possano essere distinti nella policy, e generalmente lasciati inaccessibili ai domini confinati. Il tipo **file_t** non dovrebbe esistere in un file system correttamente etichettato, poichè nei sistemi in cui è in esecuzione SELinux, tutti i file dovrebbero avere un contesto, ed il tipo **file_t** non è mai usato per la configurazione di contesto dei file. [7].

Il tipo **default_t** è usato per quei file che non corrispondono a nessuna configurazione di contesto per file, così che essi possano essere individuati come file che non hanno un contesto, e generalmente tenuti inaccessibili ai domini confinati. Se si crea una directory di primo livello come **/mydirectory/**, essa sarà di tipo **default_t**. Se qualche servizio deve accedere alla directory, occorre aggiornare la configurazione dei contesti relativi ai file, inserendo la directory interessata. Per i dettagli su come aggiungere un contesto al file di configurazione, fare riferimento alla *Sezione 5.7.2, «Modifiche persistenti: semanage fcontext»*.

I file in **/etc/selinux/targeted/contexts/files/** definiscono i contesti per file e directory. I file di questa directory sono letti dal comando **restorecon** e da **setfiles** per ripristinare file e directory ai contesti predefiniti.

5.9. Montare file systems

Per impostazione predefinita, quando si monta un file system che supporta gli attributi estesi, il contesto di sicurezza di ciascun file è ottenuto dall'attributo esteso *security.selinux* del file. Ai file sui file system che non supportano attributi estesi è assegnato un singolo contesto di sicurezza predefinito dalla policy di configurazione, basato sul tipo di file system.

Usare il comando **`mount -o context`** per scavalcare gli attuali attributi estesi, o per specificare un contesto predefinito differente per i file system che non supportano gli attributi estesi. Ciò può essere utile se si presume che il file system non abbia gli attributi corretti, come per esempio, i supporti removibili usati su sistemi diversi. Il comando **`mount -o context`** può essere usato anche per fornire di un etichetta quei file systems che non supportano gli attributi estesi, come File Allocation Table (FAT) o file systems NFS. Il contesto di sicurezza specificato con l'opzione **`context`** non è scritto su disco: i contesti originali sono preservati, e sono visibili quando si montano senza un **`context`** (se il file system aveva gli attributi estesi).

For further information about file system labeling, refer to James Morris's "Filesystem Labeling in SELinux" article: *http://www.linuxjournal.com/article/7426*.

5.9.1. Montaggi contestuali

Per montare un file system con uno specifico contesto, prevaricando i contesti già presenti se esistono, o per specificare un diverso contesto predefinito per un file system che non supporta gli attributi estesi, come utente root, eseguire il comando **`mount -o context=`*`utente_SELinux:ruolo:tipo:livello`*** quando si monta il file system desiderato. I cambiamenti di contesto non sono scritti sul disco. Per impostazione predefinita i file system NFS montati dal lato client sono etichettati con un contesto predefinito definito dalla policy. Comunemente nelle policy, questo contesto predefinito usa il tipo **`nfs_t`**. Senza ulteriori opzioni di mount, ciò evita la condivisione del file system NFS mediante altri servizi, come il Server HTTP Apache. Il seguente esempio monta il file system NFS affinché possa essere condiviso mediante il Server Apache:

```
# mount server:/export /local/mount/point -o\
context="system_u:object_r:httpd_sys_content_t:s0"
```

I nuovi file e directory creati su questo file system avranno il contesto di SELinux specificato con l'opzione **`-o context`**; comunque, poichè i cambiamenti di contesto non sono scritti su disco per queste situazioni, il contesto specificato conl'opzione **`context`**, viene mantenuto solo se la stessa opzione **`context`** è usata al prossimo montaggio, e se lo stesso cintesto è specificato.

Type Enforcement è il principale controllo di permesso usato nella targeted policy di SELinux. Nella gran parte dei casi, gli utenti ed i ruoli di SELinux possono essere ignorati, perciò, quando si prevarica il contesto con l'opzione **`-o context`**, usare l'utente **`system_u`** ed il ruolo **`object_r`**, e concentrarsi sul tipo. Se non si usa la policy MLS o sicurezza multi-category, usare il livello **`s0`**.

Nota

Quando un file system è montato con un opzione **`context`**, le modifiche di contesto (da parte degli utenti o dei processi) sono proibite. Per esempio, eseguire **`chcon`** su un file system montato con l'opzione **`context`**, dà un errore **`Operation not supported`**.

5.9.2. Modificare il contesto predefinito

Come menzionato nella *Sezione 5.8, «I tipi file_t e default_t»*, a proposito dei file systems che supportano attributi estesi, se si accede ad un file privo di contesto di sicurezza, esso è considerato come se avesse un contesto predefinito definito dalla policy di SELinux. Comunemente nelle policy, questo contesto predefinito usa il tipo **file_t**. Se si sdesidera usare un altro contesto predefinito, occorre montare il file system con l'opzione **defcontext**.

Nel seguente esempio, si monta un file system appena creato (su **/dev/sda2**) nella nuova directory **/test/**. Si assume che non ci siano regole in **/etc/selinux/targeted/contexts/files/** che specificano un contesto per la directory **/test/**:

```
# mount /dev/sda2 /test/ -o defcontext="system_u:object_r:samba_share_t:s0"
```

In questo esempio:

- the **defcontext** option defines that **system_u:object_r:samba_share_t:s0** is "the default security context for unlabeled files"[8].
- una volta montata, la directory radice (**/test/**) del file system è trattata come se fosse etichettata con il contesto specificato da **defcontext** (questa etichetta non viene salvata su disco). Ciò condiziona l'etichettatura per i file creati sotto la directory **/test/**: i nuovi file ereditano il tipo **samba_share_t** e le loro etichette sono salvate su disco.
- i file creati sotto **/test/** quando il file system è statomontato con un opzione **defcontext** conservano le loro etichette.

5.9.3. Montare un file system NFS

Per impostazione predefinita, i file system NFS dal lato client sono etichettati con un contesto di sicurezza predefinito dalla policy. Comunemente nella policy, questo contesto predefinito usa il tipo **nfs_t**. A seconda della configurazione della policy, i servizi, come il Server HTTP Apache e MySQL, potrebbero non poter leggere i file etichettati con il tipo **nfs_t**. Ciò impedisce ai file system etichettati con questo tipo dall'essere montati e quindi letti o esportati da altri servizi.

Se si vuole montare un file system NFS e leggere o esportare quel file system con un altro servizio, usare l'opzione **context** durante il montaggio per prevaricare il tipo **nfs_t**. Usare la seguente opzione di contesto per montare i file system NFS in modo da poter essere condiviso con il Server HTTP Apache:

```
mount server:/export /local/mount/point -o\
context="system_u:object_r:httpd_sys_content_t:s0"
```

Poichè in tali condizioni, i cambi di contesto non sono scritti su disco, il contesto specificato con l'opzione **context** è mantenuto solo se al successivo mount viene usata la stessa opzione **context**, ed è specificato il medesimo contesto.

In alternativa all'uso dell'opzione di mount **context** per montare file systems, è possibile attivare le booleane per consentire ai servizi di accedere ai file systems che hanno il tipo **nfs_t**. Per informazioni sulla configurazione delle booleane per consentire ai servizi di accedere al tipo **nfs_t**, fare riferimento alla *Sezione 5.6.3, «Booleane per NFS e CIFS»*.

5.9.4. Montaggi NFS multipli

Quando si montano più punti di una stessa esportazione NFS, tentando di prevaricare il contesto SELinux di ciascun montaggio con un contesto diverso, i comandi di montaggio successivi falliscono. Nel seguente esempio, il server NFS ha una singola esportazione, **/export**, con due sottodirectory, **web/** e **database/**. I seguenti comandi tentano di effettuare due montaggi dalla singola esportazione NFS, e tentano di prevaricare il contesto assegnato per ciascuno di loro:

```
# mount server:/export/web /local/web -o\
context="system_u:object_r:httpd_sys_content_t:s0"

# mount server:/export/database /local/database -o\
context="system_u:object_r:mysqld_db_t:s0"
```

Il secondo comando di mount fallisce, ed il seguente messaggio è registrato in **/var/log/messages**:

```
kernel: SELinux: mount invalid.  Same superblock, different security settings for (dev 0:15,
 type nfs)
```

Per eseguire molteplici montaggi di una singola esportazione NFS, con un contesto di sicurezza differente per ciascun punto di montaggio, usare le opzioni **-o nosharecache,context**. L'esempio seguente mostra montaggi multipli da una stessa esportazione di NFS, ciascuno con il proprio contesto di sicurezza (consentendo l'accesso di un singolo servizio per ciascuno):

```
# mount server:/export/web /local/web -o\
nosharecache,context="system_u:object_r:httpd_sys_content_t:s0"

# mount server:/export/database /local/database -o\
nosharecache,context="system_u:object_r:mysqld_db_t:s0"
```

In questo esempio, **server:/export/web** è montato localmente su **/local/web/**, ed i suoi file saranno etichettati con il tipo **httpd_sys_content_t** per consentire al Server HTTP Apache di accedervi. Invece **server:/export/database** è montato localmente su **/local/database**, ed i suoi file saranno etichettati con il tipo **mysqld_db_t** per consentire a MySQL di accedervi. Tali cambiamenti di tipo non sono scritti su disco.

Importante

L'opzione **nosharecache** permette di montare la stessa sottodirectory di una esportazione, diverse volte con contesti di sicurezza differenti (per esempio, montare più volte **/export/web**). Non montare la stessa sottodirectory da una esportazione diverse volte con contesti di sicurezza diversi, poichè ciò crea una sovrapposizione tra montaggi ed i file sarebbero accessibili da due contesti differenti.

5.9.5. Rendere persistente il contesto per i file system montati

Per rendere persistenti i contesti di montaggio per i file systems, tra rimontaggi e riavvi di sistema, aggiungere una riga in **/etc/fstab** per i file systems interessati, oppure usare una mappa di

automount, ed usare il contesto di sicurezza desiderato come opzione di montaggio. Il seguente esempio aggiunge ad una riga di **/etc/fstab** il contesto di montaggio per un file system NFS:

```
server:/export /local/mount/ nfs context="system_u:object_r:httpd_sys_content_t:s0" 0 0
```

Refer to the *Red Hat Enterprise Linux 5 Deployment Guide, Section 19.2. "NFS Client Configuration"*[9] for information about mounting NFS file systems.

5.10. Mantenere le etichette di SELinux

Questa sezione descrive cosa accade ai contesti SELinux quando file e directory sono copiati, spostati o archiviati. Inoltre viene spiegato come preservare i contesti in fase di copia e archiviazione.

5.10.1. Copiare file e directory

When a file or directory is copied, a new file or directory is created if it does not exist. That new file or directory's context is based on default-labeling rules, not the original file or directory's context (unless options were used to preserve the original context). For example, files created in user home directories are labeled with the **user_home_t** type:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
```

Se il file è copiato in un'altra directory, per esempio in **/etc/**, il nuovo file è creato in accordo alle regole di etichettatura predefinite della directory **/etc/**. Copiare un file (senza opzioni aggiuntive) può non preservare il contesto originale:

```
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
# cp file1 /etc/
$ ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

Quando **file1** è copiato in **/etc/**, se esso non esiste già, si crea un nuovo file **/etc/file1**. Come indicato nell'esempio, **/etc/file1** è etichettato con il tipo **etc_t**, in accordo con le regole di etichettatura predefinite.

When a file is copied over an existing file, the existing file's context is preserved, unless the user specified **cp** options to preserve the context of the original file, such as **--preserve=context**. SELinux policy may prevent contexts from being preserved during copies.

Copiare senza preservare i contesti di SELinux

Quando si copia un file usando il comando **cp**, senza specificare alcuna opzione, il tipo è ereditato dalla directory padre in cui il file è copiato:

[9] http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/s1-nfs-client-config.html

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
# cp file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
```

In this example, **file1** is created in a user's home directory, and is labeled with the **user_home_t** type. The **/var/www/html/** directory is labeled with the **httpd_sys_content_t** type, as shown with the **ls -dZ /var/www/html/** command. When **file1** is copied to **/var/www/html/**, it inherits the **httpd_sys_content_t** type, as shown with the **ls -Z /var/www/html/file1** command.

Preservare i contesti SELinux durante la copia

Usare il comando **cp --preserve=context** per preservare i contesti di sicurezza durante la copia:

```
$ touch file1
$ ls -Z file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
$ ls -dZ /var/www/html/
drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
# cp --preserve=context file1 /var/www/html/
$ ls -Z /var/www/html/file1
-rw-r--r--  root root unconfined_u:object_r:user_home_t:s0 /var/www/html/file1
```

In this example, **file1** is created in a user's home directory, and is labeled with the **user_home_t** type. The **/var/www/html/** directory is labeled with the **httpd_sys_content_t** type, as shown with the **ls -dZ /var/www/html/** command. Using the **--preserve=context** option preserves SELinux contexts during copy operations. As shown with the **ls -Z /var/www/html/file1** command, the **file1 user_home_t** type was preserved when the file was copied to **/var/www/html/**.

Copia e cambio del contesto

Use the **cp -Z** command to change the destination copy's context. The following example was performed in the user's home directory:

```
$ touch file1
$ cp -Z system_u:object_r:samba_share_t:s0 file1 file2
$ ls -Z file1 file2
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
-rw-rw-r--  user1 group1 system_u:object_r:samba_share_t:s0 file2
$ rm file1 file2
```

In questo esempio, il contesto di sicurezza è definito con l'opzione **-Z**. Senza l'opzione **-Z**, **file2** sarebbe stato etichettato con il contesto **unconfined_u:object_r:user_home_t**.

Copiare un file su un file esistente

When a file is copied over an existing file, the existing file's context is preserved (unless an option is used to preserve contexts). For example:

```
# touch /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
# touch /tmp/file2
# ls -Z /tmp/file2
-rw-r--r--  root root unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
# cp /tmp/file2 /etc/file1
# ls -Z /etc/file1
-rw-r--r--  root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

In questo esempio sono stati creati due file: **`/etc/file1`** che ha il tipo **`etc_t`**, e **`/tmp/file2`** che ha il tipo **`user_tmp_t`**. Il comando **`cp /tmp/file2 /etc/file1`** sostituisce **`file1`** con **`file2`**. Dopo la sostituzione, **`ls -Z /etc/file1`** mostra che **`file1`** è ora etichettato con il tipo **`etc_t`** e non il tipo **`user_tmp_t`** che ha **`/tmp/file2`**.

Importante

Copiare file e directory, piuttosto che spostarli. Ciò aiuta ad assicurare che essi siano etichettati con i corretti contesti di SELinux. I contesti di sicurezza errati possono impedire ai processi di accedere a tali file e directory.

5.10.2. Spostare file e directory

File and directories keep their current SELinux context when they are moved. In many cases, this is incorrect for the location they are being moved to. The following example demonstrates moving a file from a user's home directory to **`/var/www/html/`**, which is used by the Apache HTTP Server. Since the file is moved, it does not inherit the correct SELinux context:

1. Eseguire il comando **`cd`** senza alcun argomento per spostarsi nella home directory. Da qui, creare un file, con il comando **`touch file1`**. Il file sarà etichettato con il tipo **`user_home_t`**:

   ```
   $ ls -Z file1
   -rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 file1
   ```

2. Eseguire, **`ls -dZ /var/www/html/`** per visualizzare il contesto di SELinux per la directory **`/var/www/html/`**:

   ```
   $ ls -dZ /var/www/html/
   drwxr-xr-x  root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
   ```

   Per impostazione predefinita la directory **`/var/www/html/`** è etichettata con il tipo **`httpd_sys_content_t`**. I file e le directory create in **`/var/www/html/`** ereditano questo tipo, e sono etichettate con questo tipo.

3. Come utente Linux root, eseguire il comando **`mv file1 /var/www/html/`** per spostare **`file1`** nella directory **`/var/www/html/`**. Lo spostamento preserva il suo tipo corrente **`user_home_t`**:

```
# mv file1 /var/www/html/
# ls -Z /var/www/html/file1
-rw-rw-r--  user1 group1 unconfined_u:object_r:user_home_t:s0 /var/www/html/file1
```

Per impostazione predefinita, il server HTTP Apache non può leggere i file etichettati con il tipo **user_home_t**. Se tutti i file che formano un sito web, avessero il tipo **user_home_t**, o un altro tipo non accessibile al Server HTTP Apache, il permesso sarebbe vietato quando si tentasse di accedervi tramite Firefox o un Web browser testuale.

Importante

Spostare file e directory con **mv** può portare a contesti di SELinux errati, impedendo a processi come il Server HTTP Apache o Samba, di accedere a tali file e directory.

5.10.3. Verificare il contesto di SELinux predefinito

Use the **/usr/sbin/matchpathcon** command to check if files and directories have the correct SELinux context. From the matchpathcon(8) manual page: "**matchpathcon** queries the system policy and outputs the default security context associated with the file path."[10]. The following example demonstrates using the **/usr/sbin/matchpathcon** command to verify that files in **/var/www/html/** directory are labeled correctly:

1. Come utente Linux root, eseguire il comando **touch /var/www/html/file{1,2,3}** per creare tre file (**file1**, **file2**, e **file3**). Questi file ereditano il tipo **httpd_sys_content_t** dalla directory **/var/www/html/**:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. Come utente di Linux root, eseguire il comando **chcon -t samba_share_t /var/www/html/file1** per cambiare **file1** al tipo **samba_share_t**. Nota: il Server HTTP Apache non può leggere file che hanno il tipo **samba_share_t**.

3. L'opzione **-V** del comando **/usr/sbin/matchpathcon** confronta il contesto di SELinux corrente con il contesto corretto predefinito dalla policy di SELinux. Eseguire il comando **/usr/sbin/matchpathcon -V /var/www/html/*** per controllare tutti i file nella directory **/var/www/html/**:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/file1 has context unconfined_u:object_r:samba_share_t:s0, should be
 system_u:object_r:httpd_sys_content_t:s0
/var/www/html/file2 verified.
/var/www/html/file3 verified.
```

The matchpathcon(8) manual page, as shipped with the *libselinux-utils* package in Fedora, is written by Daniel Walsh. Any edits or changes in this version were done by Murray McAllister.

Il seguente output di **/usr/sbin/matchpathcon** mostra che **file1** è etichettato con il tipo **samba_share_t**, ma dovrebbe essere etichettato con il tipo **httpd_sys_content_t**:

```
/var/www/html/file1 has context unconfined_u:object_r:samba_share_t:s0, should be
 system_u:object_r:httpd_sys_content_t:s0
```

Per risolvere il problema di etichettatura e permettere al Server HTTP Apache di accedere a **file1**, come utente Linux root, eseguire il comando **/sbin/restorecon -v /var/www/html/file1**:

```
# /sbin/restorecon -v /var/www/html/file1
restorecon reset /var/www/html/file1 context unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

5.10.4. Archiviare file con tar

Il comando **tar**non conserva attributi estesi, per impostazione predefinita. Poiché i contesti di SELinux sono memorizzati in attributi estesi, i contesti di sicurezza possono andare perduti quando si archiviano i file. Usare il comando **tar --selinux** per creare archivi che conservino i contesti di sicurezza. Se un archivio Tar contiene file senza attributi estesi, o se si vuole che gli attributi estesi si adattino alle condizioni predefinite del sistema, eseguire l'archiviazione attraverso **/sbin/restorecon**:

```
$ tar -xvf archive.tar | /sbin/restorecon -f -
```

Nota: a seconda della directory di lavoro, per eseguire **/sbin/restorecon**, potrebbe essere necessario divenire l'utente Linux root.

Il seguente esempio dimostra come creare un archivio Tar che conservi i contesti di SELinux:

1. Come utente Linux root, eseguire il comando **touch /var/www/html/file{1,2,3}** per creare tre file (**file1**, **file2**, e **file3**). Questi file ereditano il tipo **httpd_sys_content_t** dalla directory **/var/www/html/**:

   ```
   # touch /var/www/html/file{1,2,3}
   # ls -Z /var/www/html/
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
   -rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
   ```

2. Eseguire il comando **cd /var/www/html/** per spostarsi nella directory **/var/www/html/**. Da qui, come utente Linux root, eseguire **tar --selinux -cf test.tar file{1,2,3}**, per creare un archivio Tar di nome **test.tar**.

3. Come utente Linux root, eseguire il comando **mkdir /test** per creare una nuova directory quindi lanciare **chmod 777 /test/** per consentire pieno accesso a tutti gli utenti alla directory **/test/**.

4. Eseguire **cp /var/www/html/test.tar /test/** per copiare il file **test.tar** nella directory **/test/**.

5. Eseguire il comando **cd /test/** per spostarsi nella directory **/test/**. Da qui lanciare il comando **tar -xvf test.tar** per estrarre l'archivio Tar.

6. Eseguire il comando **ls -lZ /test/** per vedere i contesti SELinux. Il tipo **httpd_sys_content_t** è stato conservato invece di essere cambiato nel tipo **default_t**, che sarebbe successo se fosse mancata l'opzione **--selinux**:

```
$ ls -lZ /test/
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file3
-rw-r--r--  user1 group1 unconfined_u:object_r:default_t:s0 test.tar
```

7. Se la directory **/test/** non è più necessaria, come utente Linux root, eseguire **rm -ri /test/** per rimuoverla assieme al suo contenuto.

Refer to the tar(1) manual page for further information about **tar**, such as the **--xattrs** option that retains all extended attributes.

5.10.5. Archiviare i file con star

Il comando **star** non conserva gli attributi estesi, per impostazione predefinita. Poiché i contesti di SELinux sono conservati in attributi estesi, i contesti possono andare perduti quando si archiviano i file. Usare il comando **star -xattr -H=exustar** per creare archivi che conservino i contesti. Il pacchetto *star* non è installato per impostazione predefinita. Per installarlo, eseguire il comando **yum install star**, come utente Linux root.

Il seguente esempio dimostra come creare un archivio Star che preserva i contesti di SELinux:

1. Come utente Linux root, eseguire il comando **touch /var/www/html/file{1,2,3}** per creare tre file (**file1**, **file2**, e **file3**). Questi file ereditano il tipo **httpd_sys_content_t** dalla directory **/var/www/html/**:

```
# touch /var/www/html/file{1,2,3}
# ls -Z /var/www/html/
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  root root unconfined_u:object_r:httpd_sys_content_t:s0 file3
```

2. Eseguire il comando **cd /var/www/html/** per spostarsi nella directory **/var/www/html/**. Da qui, come utente Linux root, lanciare il comando **star -xattr -H=exustar -c -f=test.star file{1,2,3}**, per creare un archivio di nome **test.star**:

```
# star -xattr -H=exustar -c -f=test.star file{1,2,3}
star: 1 blocks + 0 bytes (total of 10240 bytes = 10.00k).
```

3. Come utente Linux root, eseguire il comando **mkdir /test** per creare una nuova directory quindi lanciare **chmod 777 /test/** per consentire pieno accesso a tutti gli utenti alla directory **/test/**.

4. Eseguire il comando **`cp /var/www/html/test.star /test/`** per copiare il file **`test.star`** nella directory **`/test/`**.

5. Eseguire **`cd /test/`** per spostarsi nella directory **`/test/`**. Da qui, lanciare il comando **`star -x -f=test.star`** per estrarre l'archivio Star:

```
$ star -x -f=test.star
star: 1 blocks + 0 bytes (total of 10240 bytes = 10.00k).
```

6. Eseguire il comando **`ls -lZ /test/`** per vedere i contesti SELinux. Il tipo **`httpd_sys_content_t`** è stato conservato invece di essere cambiato nel tipo **`default_t`**, che sarebbe successo se fosse mancata l'opzione **`--selinux`**:

```
$ ls -lZ /test/
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file1
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file2
-rw-r--r--  user1 group1 unconfined_u:object_r:httpd_sys_content_t:s0 file3
-rw-r--r--  user1 group1 unconfined_u:object_r:default_t:s0 test.star
```

7. Se la directory **`/test/`** non è più necessaria, come utente Linux root, eseguire **`rm -ri /test/`** per rimuoverla assieme al suo contenuto.

8. Se il comando **`star`** non è più necessario, come utente root, eseguire **`yum remove star`** per rimuovere il pacchetto.

Refer to the star(1) manual page for further information about **`star`**.

Confinare gli utenti

In Fedora 13 è disponibile un certo numero di utenti di SELinux confinati. Attraverso la policy di SELinux, a ciascun utente Linux è mappato un utente SELinux, per attribuire ai primi le restrizioni proprie degli utenti di SELinux, per esempio (a seconda dell'utente) non consentire: di eseguire il sistema grafico X Window; usare la rete; eseguire applicazioni setuid; o consentire di eseguire i comandi **su** o **sudo**. Ciò aiuta a proteggere il sistema dagli utenti. Per ulteriori informazioni sugli utenti confinati fare riferimento alla *Sezione 4.3, «Utenti confinati e non confinati»*.

6.1. Mappatura degli utenti Linux e SELinux

Come utente root, eseguire il comando **semanage login -l** per vedere la mappatura fra gli utenti di SELinux e gli utenti Linux:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

Per impostazione predefinita, in Fedora 13, agli utenti Linux è mappato il **__default__** login di SELinux (a cui a sua volta, è mappato l'utente **unconfined_u** di SELinux). Al momento della creazione di un utente Linux con il comando **useradd**, all'utente creato è applicato l'utente **unconfined_u** di SELinux, se non è specificata alcuna opzione. Di seguito si riporta, l'attribuzione predefinita:

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

6.2. Confinare i nuovi utenti Linux: useradd

Gli utenti di Linux mappati all'utente **unconfined_u** di SELinux operano nel dominio **unconfined_t**. Ciò può essere verificato con il comando **id -Z**, per un utente a cui sia applicato l'utente **unconfined_u**:

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

Quando gli utenti Linux lavorano nel dominio **unconfined_t**, si applicano le regole della politica di SELinux, tuttavia esistono regole di politica che permettono agli utenti Linux operanti sotto il dominio **unconfined_t** con accesso quasi completo. Se un utente di Linux non confinato esegue un'applicazione che secondo la politica può transitare dal dominio **unconfined_t** al proprio dominio confinato, allora l'utente di Linux subisce le stesse restrizioni del dominio di destinazione. Il vantaggio di ciò in termini di sicurezza, è che sebbene l'utente di Linux operi non confinato, l'applicazione rimane confinata e perciò lo sfruttamento di una falla di sicurezza nell'applicazione sarebbe limitata dalla politica. Nota: ciò non protegge il sistema dall'utente. Invece, sono l'utente ed il sistema che risultano protetti da eventuali falle presenti nell'applicazione.

Quando si creano nuovi utenti di Linux usando il comando **useradd**, si può usare l'opzione **-Z** per specificare a quali utenti di SELinux essi vadano mappati. Nel seguente esempio si crea un nuovo utente, useruuser, mappato all'utente **user_u** di SELinux. Gli utenti di Linux attribuiti all'utente **user_u** di SELinux lavorano nel dominio **user_t**. In questo dominio gli utenti di Linux non possono eseguire applicazioni setuid, e non possono eseguire i comandi **su** o **sudo**.

1. Come utente root, lanciare il comando **/usr/sbin/useradd -Z user_u useruuser**, per creare un nuovo utente (useruuser) mappato all'utente **user_u** di SELinux.

2. Come utente root, eseguire il comando **semanage login -l** per verificare la corrispondenza di **user_u** all'utente **useruuser** di Linux:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
useruuser                 user_u                    s0
```

3. Come utente root, eseguire il comando **passwd useruuser** ed assegnare una password all'utente Linux useruuser:

```
# passwd useruuser
Changing password for user useruuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

4. Log out of your current session, and log in as the Linux useruuser user. When you log in, pam_selinux maps the Linux user to an SELinux user (in this case, **user_u**), and sets up the resulting SELinux context. The Linux user's shell is then launched with this context. Run the **id -Z** command to view the context of a Linux user:

```
[useruuser@localhost ~]$ id -Z
user_u:user_r:user_t:s0
```

5. Log out of the Linux useruuser's session, and log back in with your account. If you do not want the Linux useruuser user, run the **/usr/sbin/userdel -r useruuser** command as the Linux root user to remove it, along with its home directory.

6.3. Confinare gli utenti Linux esistenti: semanage login

Se un utente Linux è mappato all'utente SELinux **unconfined_u** (per impostazione predefinita), e si desidera cambiare a quale utente SELinux sia mappato, usare il comando **semanage login**. Il seguente esempio creerà un nuovo utente, newuser, che verrà mappato all'utente **user_u** di SELinux:

1. Come utente Linux root, eseguire il comando **/usr/sbin/useradd newuser** per creare un nuovo utente Linux (newuser). Poichè il nuovo utente usa la mappatura predefinita di SELinux, il suo nome non compare nell'output di **/usr/sbin/semanage login -l**:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

2. Per mappare l'utente newuser all'utente **user_u** di SELinux, come utente Linux root, eseguire il comando:

   /usr/sbin/semanage login -a -s user_u newuser

   L'opzione **-a** inserisce un nuovo record e l'opzione **-s** specifica quale utente di SELinux usare. L'ultimo argomento **newuser**, è il nome dell'utente di Linux che si desidera mappare all'utente SELinux.

3. Come utente root eseguire il comando **semanage login -l** per verificare l'attribuzione di **user_u** all'utente newuser:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

__default__               unconfined_u              s0-s0:c0.c1023
newuser                   user_u                    s0
root                      unconfined_u              s0-s0:c0.c1023
system_u                  system_u                  s0-s0:c0.c1023
```

4. Come utente Linux root, eseguire il comando **passwd useruuser** ed assegnare una password all'utente Linux newuser:

```
# passwd newuser
Changing password for user newuser.
New UNIX password: Enter a password
Retype new UNIX password: Enter the same password again
passwd: all authentication tokens updated successfully.
```

5. Log out of your current session, and log in as the Linux newuser user. Run the **id -Z** command to view the newuser's SELinux context:

```
[newuser@rlocalhost ~]$ id -Z
user_u:user_r:user_t:s0
```

6. Log out of the Linux newuser's session, and log back in with your account. If you do not want the Linux newuser user, run the **userdel -r newuser** command as the Linux root user to remove it, along with its home directory. Also, the mapping between the Linux newuser user and **user_u** is removed:

```
# /usr/sbin/userdel -r newuser
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

6.4. Modificare la mappatura predefinita

In Fedora 13, agli utenti di Linux viene mappato il login **__default__** di SELinux (a cui si mappa a sua volta l'utente **unconfined_u** di SELinux). Se si desidera che i nuovi utenti di Linux, e gli altri utenti non esplicitamente mappati ad un utente di SELinux siano confinati per impostazione predefinita, cambiare la mappatura predefinita eseguendo il comando **semanage login**.

Per esempio, per cambiare la mappatura predefinita da **unconfined_u** a.**user_u** eseguire come root il comando:

/usr/sbin/semanage login -m -S targeted -s "user_u" -r s0 __default__

Eseguire **semanage login -l** come utente Linux root per verificare che il login **__default__** è ora attributo a **user_u**:

```
# /usr/sbin/semanage login -l

Login Name                SELinux User              MLS/MCS Range

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

Se viene creato un nuovo utente e non è specificato alcun utente di SELinux, oppure se un utente Linux esistente che non figura in alcun record dell'output di **semanage login -l** avvia una sessione, essi saranno entrambi mappati a **user_u**, come al login **__default__**.

Per ripristinare la condizione predefinita, ossia per attribuire il login **__default__** all'utente **unconfined_u** di SELinux, eseguire come root:

```
/usr/sbin/semanage login -m -S targeted -s "unconfined_u" -r\
s0-s0:c0.c1023 __default__
```

6.5. xguest: Modo chiosco

Il pacchetto *xguest* fornisce un account kiosk. Questo account è usato per proteggere quelle postazioni ad accesso pubblico che generalmente si trovano nelle librerie, banche, aeroporti, chioschi

informativi e coffee shops. L'account kiosk è molto limitato: essenzialmente consente agli utenti solo di avviare una sessione e usare **Firefox** per navigare su Internet. Ogni modifica fatta durante la sessione, come creare file, viene persa alla chiusura.

Per impostare l'account kiosk:

1. Come utente root, installare il pacchetto *xguest* con il comando di installazione **`yum install xguest`**. Se richiesto, installare anche le dipendenze.

2. Per poter usare l'account kiosk con una varietà di persone, l'account non è protetto da password; perciò l'account può essere protetto solo se SELinux è in esecuzione in modalità imposta (enforcing mode). Usare il comando **`getenforce`** per assicurarsi che SELinux sia in esecuzione in modalità imposta, prima di avviare una sessione con questo account:

   ```
   $ /usr/sbin/getenforce
   Enforcing
   ```

   Se il risultato è diverso, fare riferimento alla *Sezione 5.5, «Modalità di SELinux»* per sapere come passare in modalità enforcing. Non è possibile avviare una sessione con questo account, se SELinux è disabilitato o in modalità permissiva.

3. Con questo account è possibile avviare una sessione solo attraverso lo GNOME Display Manager (GDM). Una volta installato il pacchetto *xguest*, un account **`Guest`** è aggiunto al GDM. Per accedere con l'account kiosk, fare click su **`Guest`**.

6.6. Booleane per gli utenti che eseguono applicazioni

Not allowing Linux users to execute applications (which inherit users' permissions) in their home directories and **`/tmp/`**, which they have write access to, helps prevent flawed or malicious applications from modifying files that users own. In Fedora 13, by default, Linux users in the **`guest_t`** and **`xguest_t`** domains can not execute applications in their home directories or **`/tmp/`**; however, by default, Linux users in the **`user_t`** and **`staff_t`** domains can.

Sono disponibili booleane per modificare questo comportamento e possono essere configurate con il comando **setsebool**, eseguito dall'utente Linux root. Il comando **setsebool -P** rende persistenti i cambiamenti. Non utilizzare l'opzione **-P** se non si desidera che i cambiamenti persistano ai riavvii:

guest_t

Per *autorizzare* gli utenti di Linux del dominio **guest_t** ad eseguire applicazioni nella loro home directory e nella directory **/tmp/**:

```
/usr/sbin/setsebool -P allow_guest_exec_content on
```

xguest_t

Per *autorizzare* gli utenti del dominio **xguest_t** ad eseguire applicazioni nella loro home directory e nella directory **/tmp/**:

```
/usr/sbin/setsebool -P allow_xguest_exec_content on
```

user_t

Per *vietare* agli utenti del dominio **user_t** di eseguire applicazioni nella loro home directory e nella directory **/tmp/**:

```
/usr/sbin/setsebool -P allow_user_exec_content off
```

staff_t

Per *vietare* agli utenti del dominio **staff_t** di eseguire applicazioni nella loro home directory e nella directory **/tmp/**:

```
/usr/sbin/setsebool -P allow_staff_exec_content off
```

Risoluzione dei problemi

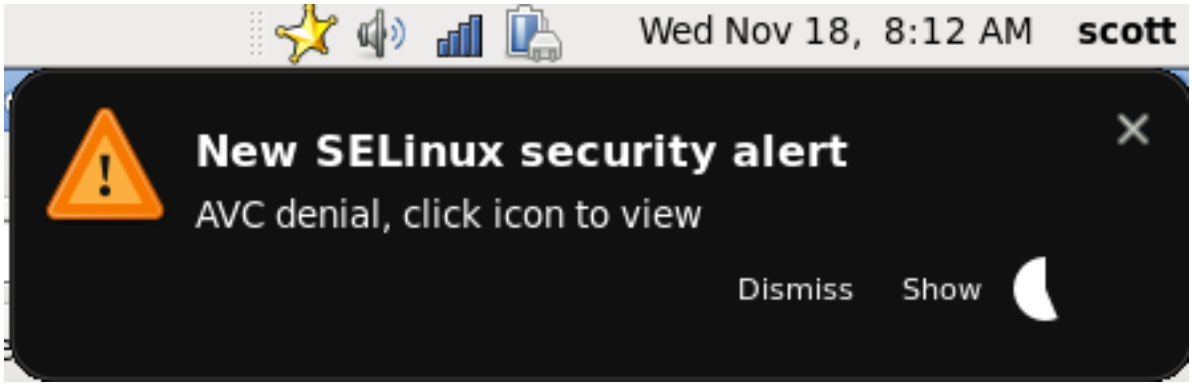
I seguenti capitoli descriveranno quel che succede quando SELinux nega l'accesso; le tre principali cause di problemi; dove trovare le informazioni per una corretta etichettatura; analizzare i dinieghi di SELinux; e creare moduli di policy personalizzati con **`audit2allow`**.

7.1. Cosa succede quando l'accesso è negato

SELinux decisions, such as allowing or disallowing access, are cached. This cache is known as the Access Vector Cache (AVC). Denial messages are logged when SELinux denies access. These denials are also known as "AVC denials", and are logged to a different location, depending on which daemons are running:

Demone	Posizione dei log
auditd on	**`/var/log/audit/audit.log`**
auditd off; rsyslogd on	**`/var/log/messages`**
setroubleshootd, rsyslogd, ed auditd on	**`/var/log/audit/audit.log`**. Dei messaggi di diniego più semplici da leggere sono anche inviati a **`/var/log/messages`**

Se è in esecuzione il sistema grafico X Windows e sono installati i pacchetti *setroubleshoot* e *setroubleshoot-server*, e sono in esecuzione i demoni `setroubleshootd` ed `auditd`, quando SELinux negherà un accesso sarà mostrato un avviso:



Clicking on 'Show' presents a detailed analysis of why SELinux denied access, and a possible solution for allowing access. If you are not running the X Window System, it is less obvious when access is denied by SELinux. For example, users browsing your website may receive an error similar to the following:

```
Forbidden

You don't have permission to access file name on this server
```

For these situations, if DAC rules (standard Linux permissions) allow access, check **`/var/log/messages`** and **`/var/log/audit/audit.log`** for **`"SELinux is preventing"`** and **`"denied"`** errors respectively. This can be done by running the following commands as the Linux root user:

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

grep "denied" /var/log/audit/audit.log
```

7.2. Le tre principali cause di problemi

Le seguenti sezioni descriveranno le tre principali cause di problemi: problemi di etichettatura, problemi sulla configurazione delle Booleane e delle porte per i servizi, e problemi legati all'evolvere delle regole di SELinux.

7.2.1. Problemi di etichettatura

Sui sistemi che eseguono SELinux, tutti i processi e i file sono marcati con una etichetta che contiene una informazione di sicurezza. Tale informazione è detta contesto di SELinux. Se le etichette sono sbagliate, l'accesso potrebbe essere vietato. Se un'applicazione ha un'etichetta sbagliata, allora anche il processo cui essa dà avvio avrà l'etichetta sbagliata, causando molto probabilmente un divieto d'accesso da parte di SELinux, e forse, la creazione di file erroneamente etichettati da parte del processo.

Una causa frequente di problemi di etichettatura si riscontra quando per un servizio si usa una una directory non-standard. Per esempio, per un sito web invece di usare **`/var/www/html/`**, un amministratore vorrebbe poter usare **`/srv/myweb/`**. In Fedora 13, la directory **`/srv/`** è etichettata con il tipo **`var_t`**. I file e le directory create in **`/srv/`** ereditano questo tipo. Inoltre, le nuove directory di primo livello (come **`/myserver/`**) saranno di tipo **`default_t`**. In tal caso, SELinux impedirà al Server HTTP Apache (`httpd`) di accedere ad entrambe le directory. Per garantire l'acesso, SELinux dovrebbe sapere che i file in **`/srv/myweb/`** devono essere accessibili ad `httpd`:

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

Il comando **`semanage`** aggiunge il contesto per la directory **`/srv/myweb/`** (inclusi i file e le sotto directory) alla configurazione del contesto per i file di SELinux [1]. Il comando **`semanage`** non modifica il contesto. Per rendere effettivo il cambiamento, come utente root, eseguire il comando **`restorecon`**:

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

Per ulteriori informazioni sull'aggiunta dei contesti per i file al file di configurazione, fare riferimento alla *Sezione 5.7.2, «Modifiche persistenti: semanage fcontext»*

7.2.1.1. Qual'è il contesto corretto?

Il comando **`matchpathcon`** verifica il contesto di un percorso e lo confronta con l'etichetta predefinita del percorso. Il seguente esempio dimostra l'uso di **`matchpathcon`** su una directory che contiene file non correttamente etichettati:

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

. I file in **`/etc/selinux/targeted/contexts/files/`** definiscono i contesti per file e directory. I file di questa directory sono letti da **`restorecon`** e **`setfiles`** per ripristinare file e directory ai loro contesti predefiniti

Nell'esempio, i file **index.html** e **page1.html** sono di tipo **user_home_t**. EssoQuesto tipo è usato per i file delle home directory degli utenti. Essendo stato usato il comando **mv** per spostare i file dalla propria home directory, essi hanno conservato il tipo **user_home_t**. Questo tipo non dovrebbe esistere al di fuori delle home directory. Per correggere i file al tipo corretto, usare il comando **restorecon**:

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

Per correggere il contesto di tutti i file presenti in una directory, usare l'opzione **-R**:

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

Per un esempio più dettagliato sull'uso di **matchpathcon**, fare riferimento a *Sezione 5.10.3, «Verificare il contesto di SELinux predefinito»*

7.2.2. Come vengono confinati i servizi in esecuzione?

I servizi possono essere eseguiti in vari modi. Per tener conto di ciò, occorre indicare in SELinux il modo in cui essi vengono eseguiti. Anche senza alcuna conoscenza su come scrivere una politica di SELinux, semplicemente usando le booleane, si possono modificare alcune parti di SELinux che si attivano al runtime. Ciò consente di apportare modifiche calibrate, come permettere l'accesso ai file systems NFS, senza dover riavviare o ricompilare la policy di SELinux. Inoltre, i servizi in esecuzione su numeri di porta diversi da quelli predefiniti, richiedono che la configurazione della policy sia aggiornata attraverso il comando **semanage**.

Per esempio, per consentire al Server HTTP Apache di comunicare con MySQL, impostare la Booleana **httpd_can_network_connect_db** su on:

```
# /usr/sbin/setsebool -P httpd_can_network_connect_db on
```

Se per un certo servizio è vietato l'accesso, usare i comandi **getsebool** e **grep** per verificare se esiste qualche booleana disponibile per permettere l'accesso. Per esempio, volendo trovare le booleane relative ad FTP, usare il comando **getsebool -a | grep ftp**:

```
$ /usr/sbin/getsebool -a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
httpd_enable_ftp_server --> off
```

```
tftp_anon_write --> off
```

Per avere un elenco delle booleane con i relativi stati on od off, eseguire il comando **/usr/sbin/getsebool -a**. Per mostrare un lista di booleane, con una spiegazione di ciò che esse rappresentano e se i loro stati sono on od off, usare il comando **/usr/sbin/semanage boolean -l**, come utente root. Per maggiori informazioni sulle liste e sulla configurazione delle booleane, fare riferimento a *Sezione 5.6, «Booleane»*

Numeri di porta

A seconda della configurazione della politica, ai servizi è concesso di lavorare soltanto su certi numeri di porta. Modificare il numero di porta senza apportare le corrette modifiche alla policy, generalmente può portare all'impossibilità di avviare il servizio. Per esempio, eseguire **semanage port -l | grep http**, come root, per avere un elenco di numeri di porta relativi al demone `http`:

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

Il tipo di porta **http_port_t** defeinisce i numeri di porta su cui il Server HTTP Apache può ascoltare, che in questo caso sono le porte TCP 80, 443, 488, 8008, 8009, e 8443. Se un amministratore configura **httpd.conf** in modo che `httpd` usi il numero di porta 9876, e trascuri di aggiornare la policy, il comando d'avvio **service httpd start** del demone fallisce:

```
# /sbin/service httpd start
Starting httpd: (13)Permission denied: make_sock: could not bind to address [::]:9876
(13)Permission denied: make_sock: could not bind to address 0.0.0.0:9876
no listening sockets available, shutting down
Unable to open logs
                  [FAILED]
```

Un diniego di SELinux simile al seguente è registrato su **/var/log/audit/audit.log**:

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

Per permettere ad `httpd` di ascoltare su un numero di porta non elencata dal tipo di porta **http_port_t**, eseguire il comando **semanage port** per aggiungere una porta nel file di configurazione della policy [2].

```
# /usr/sbin/semanage port -a -t http_port_t -p tcp 9876
```

Il comando **semanage port -a** inserisce un nuovo record nel file **/etc/selinux/targeted/modules/active/ports.local**. Nota: per impostazione predefinita, solo l'utente Linux root può leggere tale file.

L'opzione **-a** aggiunge un nuovo record; l'opzione **-t** definisce un tipo; e l'opzione **-p** un protocollo. L'ultimo argomento indica il numero di porta da aggiungere.

7.2.3. Regole di politica in evoluzione ed applicazioni malfunzionanti

Le applicazioni possono malfunzionare, provocando un divieto d'accesso da parte di SELinux. Inoltre, le stesse regole di SELinux evolvono - SELinux può negare un permesso d'accesso ad un'applicazione, che viene eseguita come richiesto ma per cui non è stato correttamente istruito. Per esempio, se viene rilasciata una nuova versione di PostgreSQL, essa potrebbe richiedere certe azioni mai contemplate dalla politica corrente, causando un divieto d'accesso.

In tali situazioni, dopo che l'accesso è stato negato, creare un modulo di politica personalizzata che garantisca l'accesso, usando il comando **audit2allow**. Per ulteriori informazioni sull'uso di **audit2allow** fare riferimento a *Sezione 7.3.8, «Consentire l'accesso: audit2allow»*

7.3. Risolvere i problemi

Le seguenti sezioni vi aiuteranno nella risoluzione dei problemi. Essi affronteranno i seguenti argomenti: verificare i permessi di Linux, usati prima di applicare le regole di SELinux; possibili cause di divieto d'accesso da parte di SELinux, ma privi di messaggi di avviso; pagine di man sui servizi, che contengono informazioni sulle etichette e sulle booleane; domini permissivi, per consentire ad un processo di essere eseguito in modalità permissiva; come vedere o trovare i messaggi di diniego; analizzare i dinieghi; e creare moduli di politica personalizzati con **audit2allow**.

7.3.1. I permessi di Linux

Quando un accesso è negato, provare a controllare i permessi di Linux canonici. Come già ricordato nel *Capitolo 2, Introduzione*, molti sistemi operativi usano un sistema discrezionale di controllo degli accessi (Discretionary Access Control o DAC), permettendo agli utenti di controllare i permessi sui loro file. Le regole della politica di SELinux non vengono prese in considerazione, se le regole DAC, in primis, negano l'accesso.

Se l'accesso è negato e non esiste alcun messaggio di avviso di SELinux, usare il comando **ls -l** per vedere i permessi standard di Linux:

```
$ ls -l /var/www/html/index.html
-rw-r----- 1 root root 0 2009-05-07 11:06 index.html
```

Nell'esempio, il file **index.html** appartiene all'utente ed al gruppo root. L'utente root ha i permessi di lettura e scrittura (**-rw**) ed i membri del gruppo root hanno il permesso di lettura (**-r-**). Gli altri non hanno alcun permesso d'accesso (**---**). Per impostazione predefinita, tali permessi non permettono `httpd` a leggere questo file. Per risolvere il problema, occorre cambiare il proprietario e il gruppo di appartenenza del file, usando il comando **chown**. Il comando deve essere eseguito dall'utente Linux root:

```
# chown apache:apache /var/www/html/index.html
```

Si assume valida la configurazione predefinita, in cui `httpd` è in esecuzione come l'utente di Linux apache. Se `httpd` è in esecuzione con un utente diverso, sostituire **`apache:apache`** con quell'utente.

Refer to the *Fedora Documentation Project "Permissions"*[3] draft for information about managing Linux permissions.

7.3.2. Possibili cause di dinieghi silenziosi

In alcune situazioni, i divieti AVC possono essere privi di messaggi di avviso quando SELinux nega l'accesso. Per svolgere i loro compiti, le applicazioni e le funzioni di libreria del sistema spesso richiedono maggiori permessi di quanto normalmente richiesto. Per mantenere i minimi privilegi senza riempire i file di log di audit con divieti AVC causati di innocenti applicazioni, la politica può essere configurata silenziare gli innumerevoli messaggi di divieto AVC senza consentire un permesso, usando le regole **`dontaudit`**. Queste regole sono comuni nella policy standard. Lo svantaggio delle regole **`dontaudit`** è di rendere più difficoltoso le operazioni di risoluzione dei problemi, in quanto ad un divieto d'accesso di SELinux il relativo messaggio non viene registrato.

Per disabilitare temporaneamente le regole **`dontaudit`**, e registrare tutti i messaggi di divieto, come utente Linux root, eseguire il seguente comando:

`/usr/sbin/semodule -DB`

L'opzione **`-D`**, disabilita le regole di **`dontaudit`**; l'opzione **`-B`** rigenera la policy. Dopo aver lanciato il comando **`semodule -DB`**, provare ad usare l'applicazione che dava origine a problemi d'accesso, e verificare se i messaggi di divieto relativi all'applicazione vengono registrati. A questo punto si può decidere con cura quali divieti dovrebbero essere permessi e quali dovrebbero essere ignorati e gestiti tramite le regole **`dontaudit`**. In caso di dubbi o per ulteriori informazioni, contattare altri utilizzatori di SELinux e gli sviluppatori presenti in una mailing list su SELinux, come *fedora-selinux-list*[4].

Per rigenerare la policy ed abilitare le regole di **`dontaudit`**, eseguire come root il comando:

`/usr/sbin/semodule -B`

Ciò ristabilisce la politica al suo stato originale. Per vedere una lista completa delle regole **`dontaudit`**, usare il comando **`sesearch --dontaudit`**. Per restringere le ricerche usare l'opzione **`-s dominio`** in pipe a **`grep`**. Per esempio:

```
$ sesearch --dontaudit -s smbd_t | grep squid
WARNING: This policy contained disabled aliases; they have been removed.
dontaudit smbd_t squid_port_t : tcp_socket name_bind ;
dontaudit smbd_t squid_port_t : udp_socket name_bind ;
```

Per informazioni sull'analisi dei divieti fare riferimento a *Sezione 7.3.6, «Messaggi Audit Raw»* e *Sezione 7.3.7, «Messaggi sealert»*

7.3.3. Pagine di man sui servizi

Le pagine di man sui servizi, contengono notevoli informazioni, come che tipo per i file usare in una data situazione, o quali booleane per cambiare l'accesso posseduto da un servizio (come permettere

[3] http://fedoraproject.org/wiki/Docs/Drafts/AdministrationGuide/Permissions
[4] http://www.redhat.com/mailman/listinfo/fedora-selinux-list

ad `httpd` di accedere al file system NFS). Questa informazione può trovarsi nelle pagine di man usuali, o in una pagina di man con il termine **`selinux`** preposto o posposto.

For example, the httpd_selinux(8) manual page has information about what file type to use for a given situation, as well as Booleans to allow scripts, sharing files, accessing directories inside user home directories, and so on. Other manual pages with SELinux information for services include:

- Samba: the samba_selinux(8) manual page describes that files and directories to be exported via Samba must be labeled with the **`samba_share_t`** type, as well as Booleans to allow files labeled with types other than **`samba_share_t`** to be exported via Samba.
- NFS: the nfs_selinux(8) manual page describes that, by default, file systems can not be exported via NFS, and that to allow file systems to be exported, Booleans such as **`nfs_export_all_ro`** or **`nfs_export_all_rw`** must be turned on.
- Berkeley Internet Name Domain (BIND): the named(8) manual page describes what file type to use for a given situation (see the **`Red Hat SELinux BIND Security Profile`** section). The named_selinux(8) manual page describes that, by default, `named` can not write to master zone files, and to allow such access, the **`named_write_master_zones`** Boolean must be turned on.

Le informazioni presenti nelle pagine di man aiutano ad assegnare correttamenti i tipi per i file e le booleane, aiutando a prevenire i dinieghi d'accesso da SELinux.

7.3.4. Domini permissivi

Quando è in esecuzione in modalità permissiva, SELinux non nega l'accesso, tuttavia registra i messaggi di quelle azioni che sarebbero vietate se SELinux fosse in modalità enforcing. Nel passato, non era possibile rendere permissivo un singolo dominio (si ricordi che i processi sono eseguiti in domini). In certe situazioni, era quindi necessario rendere l'intero sistema permissivo per risolvere i problemi.

Fedora 13 introduce i domini permissivi, in cui un amministratore può configurare l'esecuzione in modo permissivo di un singolo processo (dominio), piuttosto che l'intero sistema. I controlli di SELinux sono eseguiti anche sui domini permissivi; tuttavia, il kernel consente l'accesso e riporta un messaggio AVC per quelle situazioni in cui SELinux avrebbe negato l'accesso. I domini permissivi sono disponibili anche in Fedora 9 (applicando gli ultimi aggiornamenti).

In Red Hat Enterprise Linux 4 e 5, le booleane ***`dominio`*`_disable_trans`** sono configurabili per evitare che un'applicazione transiti in un dominio confinato, e perciò, il processo rimane in esecuzione in un dominio non confinato, come **`initrc_t`**. Attivare tali booleane può provocare ulteriori problemi. Per esempio, se la booleana **`httpd_disable_trans`** è attiva:

- `httpd` viene eseguito nel dominio **`initrc_t`** non confinato. I file creati dai processi in esecuzione sotto il dominio **`initrc_t`** potrebbero non avere le medesime regole di etichettatura applicate ai file creati da un processo eseguito nel dominio **`httpd_t`**, potenzialmente permettendo ai processi di creare file non etichettati correttamente. Questo potrebbe causare in seguito problemi d'accesso.
- I domini confinati che possono comunicare con **`httpd_t`**, non possono comunicare con il dominio **`initrc_t`**, causando possibili ulteriori problemi.

Le booleane ***`dominio`*`_disable_trans`** furono eliminate da Fedora 7, anche senza aver trovato prima un sostitutivo. I domini permissivi permettono di risolvere tali problemi: applicando le regole di transizione i file vengono creati con le corrette etichette.

I domini permissivi possono essere usati per:

- rendere permissivo un singolo processo (dominio) per risolvere un problema, piuttosto che mettere a rischio l'intero sistema ponendolo interamente in stato permissivo.
- creare policy per nuove applicazioni. In precedenza, si è raccomandato di creare una policy minimale e di porre l'intero sistema in uno stato permissivo, per consentire alle applicazioni di funzionare, mentre SELinux registra i messaggi di divieto. Poi si potrebbe usare **audit2allow** per creare la policy. In questo modo si pone a rischio l'intero sistema. Usando i domini permissivi, si può rendere permissivo soltanto quello interessato alla nuova policy, senza porre a rischio l'intero sistema.

7.3.4.1. Rendere permissivo un dominio

Eseguire il comando **semanage permissive -a *dominio*** per rendere permissivo un dominio, in cui *dominio* è il dominio che si vuole trasformare. Per esempio, per rendere permissivo il dominio **httpd_t** (in cui viene eseguito il Server HTTP Apache), lanciare come root, il comando:

/usr/sbin/semanage permissive -a httpd_t

Per visualizzare l'elenco di domini che sono stati resi permissivi, eseguire come root, **semodule -l | grep permissive**. Per esempio:

```
# /usr/sbin/semodule -l | grep permissive
permissive_httpd_t      1.0
```

Per revocare ad un dominio la permissività, come root, eseguire **semanage permissive -d *domain***. Per esempio:

/usr/sbin/semanage permissive -d httpd_t

7.3.4.2. Messaggi di diniego per i domini permissivi

Il messaggio **SYSCALL** è diverso per i domini permissivi. Il seguente è un esempio di diniego AVC (e la relativa chiamata di sistema) dal Server HTTP Apache:

```
type=AVC msg=audit(1226882736.442:86): avc:  denied  { getattr } for  pid=2427 comm="httpd"
 path="/var/www/html/file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

type=SYSCALL msg=audit(1226882736.442:86): arch=40000003 syscall=196 success=no exit=-13
 a0=b9a1e198 a1=bfc2921c a2=54dff4 a3=2008171 items=0 ppid=2425 pid=2427 auid=502 uid=48
 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/
usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Per impostazione predefinita il dominio **httpd_t** non è permissivo, perciò l'azione è vietata ed il messaggio **SYSCALL** contiene **success=no**. Il seguente è un esempio di divieto AVC per la stessa situazione, eccetto per il comando **semanage permissive -a httpd_t** che ha reso permissivo il dominio **httpd_t**:

```
type=AVC msg=audit(1226882925.714:136): avc:  denied  { read } for  pid=2512
 comm="httpd" name="file1" dev=dm-0 ino=284133 scontext=unconfined_u:system_r:httpd_t:s0
 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file
```

```
type=SYSCALL msg=audit(1226882925.714:136): arch=40000003 syscall=5 success=yes exit=11
 a0=b962a1e8 a1=8000 a2=0 a3=8000 items=0 ppid=2511 pid=2512 auid=502 uid=48 gid=48 euid=48
 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4 comm="httpd" exe="/usr/sbin/httpd"
 subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

In questo caso, nonostante sia stato registrato un divieto AVC, l'accesso non è negato, come mostrato da **`success=yes`** nel messaggio della **`SYSCALL`**.

Refer to Dan Walsh's *"Permissive Domains"*[5] blog entry for further information about permissive domains.

7.3.5. Trovare e visualizzare i dinieghi

In questa sezione si assume che siano installati i pacchetti *setroubleshoot*, *setroubleshoot-server*, ed *audit*, e che siano in esecuzione `auditd`, `rsyslogd`, e `setroubleshootd`. Per informazioni su come avviare questi demoni fare riferimento a *Sezione 5.2, «File usati per registrare i messaggi di SELinux»* . Per trovare e visualizzare i messaggi di SELinux, esistono un certo numero di strumenti come **`ausearch`**, **`aureport`**, e **`sealert`**.

ausearch

The *audit* package provides **`ausearch`**. From the ausearch(8) manual page: "**`ausearch`** is a tool that can query the audit daemon logs based for events based on different search criteria"[6]. The **`ausearch`** tool accesses **`/var/log/audit/audit.log`**, and as such, must be run as the Linux root user:

Ricerca di	Comando
tutti i divieti	**`/sbin/ausearch -m avc`**
divieti a partire da oggi	**`/sbin/ausearch -m avc -ts today`**
divieti negli ultimi 10 minuti	**`/sbin/ausearch -m avc -ts recent`**

To search for SELinux denials for a particular service, use the **`-c`** ***`comm-name`*** option, where *`comm-name`* "is the executable's name"[7], for example, `httpd` for the Apache HTTP Server, and `smbd` for Samba:

`/sbin/ausearch -m avc -c httpd`

`/sbin/ausearch -m avc -c smbd`

Refer to the ausearch(8) manual page for further **`ausearch`** options.

aureport

The *audit* package provides **`aureport`**. From the aureport(8) manual page: "**`aureport`** is a tool that produces summary reports of the audit system logs"[8]. The **`aureport`** tool accesses **`/var/log/audit/audit.log`**, and as such, must be run as the Linux root user. To view a list of SELinux denials and how often each one occurred, run the **`aureport -a`** command. The following is example output that includes two denials:

```
# /sbin/aureport -a
```

[5] http://danwalsh.livejournal.com/24537.html
From the ausearch(8) manual page, as shipped with the *audit* package in Fedora 13.
From the ausearch(8) manual page, as shipped with the *audit* package in Fedora 13.
From the aureport(8) manual page, as shipped with the *audit* package in Fedora 13.

```
AVC Report
==============================================================
# date time comm subj syscall class permission obj event
==============================================================
1. 05/01/2009 21:41:39 httpd unconfined_u:system_r:httpd_t:s0 195 file getattr
 system_u:object_r:samba_share_t:s0 denied 2
2. 05/03/2009 22:00:25 vsftpd unconfined_u:system_r:ftpd_t:s0 5 file read
 unconfined_u:object_r:cifs_t:s0 denied 4
```

Refer to the aureport(8) manual page for further **aureport** options.

sealert

Il pacchetto *setroubleshoot-server* fornisce il comando **sealert**, che legge i messaggi di divieto tradotti da *setroubleshoot-server*. Ai dinieghi sono assegnati degli ID, come si può vedere in **/var/log/messages**. Ecco un esempio di divieto tratto da **messages**:

```
setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/
file1 (samba_share_t). For complete SELinux messages. run sealert -l 84e0b04d-
d0ad-4347-8317-22e74f6cd020
```

Nell'esempio, l'ID del diniego è **84e0b04d-d0ad-4347-8317-22e74f6cd020**. L'opzione **-l** richiede un ID come argomento. Eseguendo il comando **sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020** viene presentata un'analisi dettagliata del motivo per cui SELinux ha negato l'accesso, ed una possibile soluzione per permettere l'accesso.

If you are running the X Window System, have the *setroubleshoot* and *setroubleshoot-server* packages installed, and the `setroubleshootd`, `dbus` and `auditd` daemons are running, a warning is displayed when access is denied by SELinux. Clicking on 'Show' launches the **sealert** GUI, and displays denials in HTML output:

- Eseguire il comando **sealert -b** per lanciare l'interfaccia grafica di **sealert**.
- Eseguire il comando **sealert -l *** per avere una analisi dettagliata di tutti i divieti.
- As the Linux root user, run the **sealert -a /var/log/audit/audit.log -H > audit.html** command to create a HTML version of the **sealert** analysis, as seen with the **sealert** GUI.

Refer to the sealert(8) manual page for further **sealert** options.

7.3.6. Messaggi Audit Raw

I messaggi Raw Audit sono registrati nel file **/var/log/audit/audit.log**. Il seguente è un esempio di diniego AVC (e la relativa chiamata di sistema) che si verifica quando il Server HTTP Apache (in esecuzione nel dominio **httpd_t**) tenta di accedere al file **/var/www/html/file1** (di tipo **samba_share_t**):

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

type=SYSCALL msg=audit(1226874073.147:96): arch=40000003 syscall=196 success=no exit=-13
 a0=b98df198 a1=bfec85dc a2=54dff4 a3=2008171 items=0 ppid=2463 pid=2465 auid=502 uid=48
 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=6 comm="httpd" exe="/
usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

{ getattr }
: The item in braces indicates the permission that was denied. **getattr** indicates the source process was trying to read the target file's status information. This occurs before reading files. This action is denied due to the file being accessed having the wrong label. Commonly seen permissions include **getattr**, **read**, and **write**.

comm="*httpd*"
: The executable that launched the process. The full path of the executable is found in the **exe=** section of the system call (**SYSCALL**) message, which in this case, is **exe="/usr/sbin/httpd"**.

path="*/var/www/html/file1*"
: Il percorso dell'oggetto (target) a cui il processo ha tentato di accedere.

scontext="*unconfined_u:system_r:httpd_t:s0*"
: Il contesto di SELinux del processo che ha tentato l'azione negata. In questo caso, si tratta del contesto SELInux del Server HTTP Apache, in esecuzione nel dominio **httpd_t**.

tcontext="*unconfined_u:object_r:samba_share_t:s0*"
: Il contesto di SELinux dell'oggetto (target) a cui il processo ha tentato l'accesso. In questo caso, è il contesto di **file1**. Nota: il tipo **samba_share_t** non è accessibile ai processi in esecuzione nel dominio **httpd_t**.

  In alcune situazioni, il **tcontext** può coincidere con l'**scontext**, per esempio quando un processo tenta di eseguire un servizio di sistema che cambierà le caratteristiche del processo, come l'ID dell'utente. Analogamente il **tcontext** potrà coincidere con l'**scontext**, quando un processo cerca di usare più risorse (come la memoria) di quanto normalmente consentito, causando l'attivazione di un controllo di sicurezza per verificare se il processo ha diritto di superare le risorse assegnategli.

Nei messaggi delle chiamate di sistema (**SYSCALL**), ci sono due elementi di interesse:

- **success=**: indica se è stato o no applicato il divieto (AVC). **success=no** indica che la chiamata di sistema non ha avuto successo (SELinux ha vietato l'accesso). **success=yes** indica che la chiamata di sistema ha avuto successo - ciò può verificarsi per i domini permissivi o non confinati, come **initrc_t** e **kernel_t**.
- **exe="*/usr/sbin/httpd*"**: the full path to the executable that launched the process, which in this case, is **exe="/usr/sbin/httpd"**.

An incorrect file type is a common cause for SELinux denying access. To start troubleshooting, compare the source context (**scontext**) with the target context (**tcontext**). Should the process (**scontext**) be accessing such an object (**tcontext**)? For example, the Apache HTTP Server (**httpd_t**) should only be accessing types specified in the httpd_selinux(8) manual page, such as **httpd_sys_content_t**, **public_content_t**, and so on, unless configured otherwise.

7.3.7. Messaggi sealert

Ai messaggi di divieto sono assegnati degli IDs, come si può vedere nel file **/var/log/messages**. Nel seguente esempio si mostra un divieto AVC (registrato in **messages**) che si verifica quando il Server HTTP Apache (in esecuzione nel dominio **httpd_t**) tenta di accedere al file **/var/www/html/file1** (di tipo **samba_share_t**):

```
hostname setroubleshoot: SELinux is preventing httpd (httpd_t) "getattr" to /var/www/
html/file1 (samba_share_t). For complete SELinux messages. run sealert -l 84e0b04d-
d0ad-4347-8317-22e74f6cd020
```

Come suggerito, eseguire il comando **sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020** per vedere il messaggio completo. Esso vale solo sulla macchina locale, e presenta la stessa informazione del comando **sealert**, con interfaccia grafica:

```
$ sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020

Summary:

SELinux is preventing httpd (httpd_t) "getattr" to /var/www/html/file1
(samba_share_t).

Detailed Description:

SELinux denied access to /var/www/html/file1 requested by httpd.
/var/www/html/file1 has a context used for sharing by different program. If you
would like to share /var/www/html/file1 from httpd also, you need to change its
file context to public_content_t. If you did not intend to this access, this
could signal a intrusion attempt.

Allowing Access:

You can alter the file context by executing chcon -t public_content_t
'/var/www/html/file1'

Fix Command:

chcon -t public_content_t '/var/www/html/file1'
```

```
Additional Information:

Source Context                unconfined_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:samba_share_t:s0
Target Objects                /var/www/html/file1 [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
Host                          hostname
Source RPM Packages           httpd-2.2.10-2
Target RPM Packages
Policy RPM                    selinux-policy-3.5.13-11.fc12
Selinux Enabled               True
Policy Type                   targeted
MLS Enabled                   True
Enforcing Mode                Enforcing
Plugin Name                   public_content
Host Name                     hostname
Platform                      Linux hostname 2.6.27.4-68.fc12.i686 #1 SMP Thu Oct
30 00:49:42 EDT 2008 i686 i686
Alert Count                   4
First Seen                    Wed Nov  5 18:53:05 2008
Last Seen                     Wed Nov  5 01:22:58 2008
Local ID                      84e0b04d-d0ad-4347-8317-22e74f6cd020
Line Numbers

Raw Audit Messages

node=hostname type=AVC msg=audit(1225812178.788:101): avc:  denied  { getattr }
 for  pid=2441 comm="httpd" path="/var/www/html/file1" dev=dm-0 ino=284916
 scontext=unconfined_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0
 tclass=file

node=hostname type=SYSCALL msg=audit(1225812178.788:101): arch=40000003 syscall=196 success=no
 exit=-13 a0=b8e97188 a1=bf87aaac a2=54dff4 a3=2008171 items=0 ppid=2439 pid=2441 auid=502
 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=3 comm="httpd"
 exe="/usr/sbin/httpd" subj=unconfined_u:system_r:httpd_t:s0 key=(null)
```

Sommario

Un breve sommario dell'azione vietata. Simile al divieto descritto in **`/var/log/messages`**. Nell'esempio, al processo `httpd` è stato negato l'accesso a (**`file1`**) che è di tipo **`samba_share_t`**.

Descrizione dettagliata

Una descrizione più dettagliata. In quest'esempio, **`file1`** è di tipo **`samba_share_t`**. Questo tipo è usato per quei file e directory che si vogliono esportare con Samba. La descrizione suggerisce di cambiare il tipo in un tipo che sia accessibile al Server HTTP Apache ed eventualmente, al Server Samba.

Abilitazione accesso in corso

Un suggerimento per consentire l'accesso. Ciò può ottenersi modificando il contesto per i file, settando le booleane, o realizzando un modulo di policy locale. In questo esempio, si suggerisce di cambiare il contesto con un tipo accessibile al Server HTTP Apache e Samba.

Comando per risolvere il problema

Si suggerisce un comando per consentire l'accesso e risolvere il divieto. Nell'esempio, si fornisce il comando per cambiare il tipo per **`file1`**, nel tipo **`public_content_t`**, accessibile sia al Server HTTP Apache sia a Samba.

Informazioni aggiuntive
: Informazione utile nei rapporti di errore (bug reports), come il nome e la versione del pacchetto della policy (**selinux-policy-3.5.13-11.fc12**), irrilevante però per la soluzione del problema.

Messaggi Audit Raw
: I messaggi audit raw registrati in **/var/log/audit/audit.log** e relativi al divieto. Per informazioni sui vari termini di un messaggio AVC, fare riferimento alla *Sezione 7.3.6, «Messaggi Audit Raw»*.

7.3.8. Consentire l'accesso: audit2allow

Non usare l'esempio di questa sezione su un sistema di produzione. Si dimostrerà l'uso del comando **audit2allow**.

From the audit2allow(1) manual page: "**audit2allow** - generate SELinux policy allow rules from logs of denied operations"[9]. After analyzing denials as per *Sezione 7.3.7, «Messaggi sealert»*, and if no label changes or Booleans allowed access, use **audit2allow** to create a local policy module. After access is denied by SELinux, running the **audit2allow** command presents Type Enforcement rules that allow the previously denied access.

Nell'esempio riportato di seguito si dimostra come usare **audit2allow** per creare un modulo di policy:

1. Un messaggio di divieto e la relativa chiamata di sistema sono registrati in **/var/log/audit/audit.log**:

```
type=AVC msg=audit(1226270358.848:238): avc:  denied  { write }
 for  pid=13349 comm="certwatch" name="cache" dev=dm-0 ino=218171
 scontext=system_u:system_r:certwatch_t:s0 tcontext=system_u:object_r:var_t:s0 tclass=dir

type=SYSCALL msg=audit(1226270358.848:238): arch=40000003 syscall=39 success=no exit=-13
 a0=39a2bf a1=3ff a2=3a0354 a3=94703c8 items=0 ppid=13344 pid=13349 auid=4294967295
 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295
 comm="certwatch" exe="/usr/bin/certwatch" subj=system_u:system_r:certwatch_t:s0
 key=(null)
```

   In this example, **certwatch** (**comm="certwatch"**) was denied write access (**{ write }**) to a directory labeled with the **var_t** type (**tcontext=system_u:object_r:var_t:s0**). Analyze the denial as per *Sezione 7.3.7, «Messaggi sealert»*. If no label changes or Booleans allowed access, use **audit2allow** to create a local policy module.

2. Quando si ha un messaggio di divieto, come questo.relativo a **certwatch** al passo 1, eseguire il comando **audit2allow -w -a** per generare una descrizione sul motivo del divieto. L'opzione **-a** legge tutti i messaggi di divieto registrati. L'opzione **-w** genera una descrizione. Poichè **audit2allow** accede al file **/var/log/audit/audit.log**, esso deve essere eseguito dall'utente root:

```
# audit2allow -w -a
```

From the audit2allow(1) manual page, as shipped with the *policycoreutils* package in Fedora 13.

```
type=AVC msg=audit(1226270358.848:238): avc:  denied  { write }
 for  pid=13349 comm="certwatch" name="cache" dev=dm-0 ino=218171
 scontext=system_u:system_r:certwatch_t:s0 tcontext=system_u:object_r:var_t:s0 tclass=dir
 Was caused by:
  Missing type enforcement (TE) allow rule.

 You can use audit2allow to generate a loadable module to allow this access.
```

Come mostrato, l'accesso è vietato in quanto manca una regola di Type Enforcement.

3. Eseguire il comando **audit2allow -a** per vedere la regola di Type Enforcement che autorizza l'accesso:

```
# audit2allow -a


#============= certwatch_t ==============
allow certwatch_t var_t:dir write;
```

Importante

Quando manca una regola di Type Enforcement, generalmente si tratta di un baco(bug) nella policy di SELinux che dovrebbe essere segnalato su *Red Hat Bugzilla*[10]. Per Fedora, creare un nuovo bug per il prodotto **Fedora**, e selezionare il componente **selinux-policy**. Allegare a tale segnalazione l'uscita di **audit2allow -w -a** e di **audit2allow -a**.

4. Per usare la regola suggerita da **audit2allow -a**, eseguire, come utente root, il comando **audit2allow -a -M *mycertwatch*** per creare un modulo di policy personalizzato. L'opzione **-M** crea un file di Type Enforcement (**.te**) di nome specificato con l'opzione **-M** nella directory corrente:

```
# audit2allow -a -M mycertwatch

****************** IMPORTANT ***********************
To make this policy package active, execute:

semodule -i mycertwatch.pp

# ls
mycertwatch.pp  mycertwatch.te
```

Inoltre, **audit2allow** compila la regola di Type Enforcement in un pacchetto di policy (**.pp**). Per installare il modulo, eseguire il comando **/usr/sbin/semodule -i *mycertwatch.pp***, come root.

Importante

I moduli creati con **audit2allow** potrebbero concedere più accessi di quanto realmente richiesto. Si raccomanda di inviare la policy creata con **audit2allow** ad

una mailing list su SELinux, come *fedora-selinux-list*[11], per una revisione. Se si ritiene che ci sia un bug nella policy, creare un nuovo bug su *Red Hat Bugzilla*[12], descrivendo il problema.

Se si hanno divieti multipli da diversi processi, e si vuole creare una policy personalizzata per un singolo processo, usare **grep** per restringere l'input di **audit2allow**, (in pipe col precedente).
Di seguito si mostra l'uso di **grep** per selezionare solo i divieti relativi a **certwatch** da inviare ad **audit2allow**:

```
# grep certwatch /var/log/audit/audit.log | audit2allow -M mycertwatch2
******************** IMPORTANT ***********************
To make this policy package active, execute:

# /usr/sbin/semodule -i mycertwatch2.pp
```

Refer to Dan Walsh's *"Using audit2allow to build policy modules. Revisited."*[13] blog entry for further information about using **audit2allow** to build policy modules.

[13] http://danwalsh.livejournal.com/24750.html

Ulteriori informazioni

8.1. Contributori

- *Geert Warrink*[1] (traduzione - Olandese)
- *Domingo Becker*[2] (traduzione - Spagnolo)
- *Daniel Cabrera*[3] (traduzione - Spagnolo)

8.2. Altre risorse

La National Security Agency (NSA)

Dalla pagina dell'NSA *Contributori a SELinux*[4]:

Researchers in NSA's National Information Assurance Research Laboratory (NIARL) designed and implemented flexible mandatory access controls in the major subsystems of the Linux kernel and implemented the new operating system components provided by the Flask architecture, namely the security server and the access vector cache. The NSA researchers reworked the LSM-based SELinux for inclusion in Linux 2.6. NSA has also led the development of similar controls for the X Window System (XACE/XSELinux) and for Xen (XSM/Flask).

- Sito web principale di SELinux: *http://www.nsa.gov/research/selinux/index.shtml*.
- Documentazione su SELinux: *http://www.nsa.gov/research/selinux/docs.shtml*.
- Background su SELinux: *http://www.nsa.gov/research/selinux/background.shtml*.

Tresys Technology

Alla *Tresys Technology*[5] mantengono:

- *Librerie e strumenti SELinux nello spazio utente*[6].
- *Policy di Riferimento SELinux*[7].

Notizie SELinux

- Notizie: *http://selinuxnews.org/wp/*.
- Pianeta SELinux (blogs): *http://selinuxnews.org/planet/*.

SELinux Project Wiki

- Pagina principale: *http://selinuxproject.org/page/Main_Page*.
- Risorse per gli utenti, compresi i link alla documentazione, le mailing list, i siti web, e gli strumenti: *http://selinuxproject.org/page/User_Resources*.

[4] http://www.nsa.gov/research/selinux/contrib.shtml
[5] http://www.tresys.com/

Red Hat Enterprise Linux

- La guida *Red Hat Enterprise Linux Deployment Guide*[8] contiene una sezione con *Riferimenti*[9] a SELinux, in cui è possibile trovare collegamenti a tutorial, informazioni generali, ed informazioni sulla tecnologia di SELinux.
- *Red Hat Enterprise Linux 4 SELinux Guide*[10].

Fedora

- Pagina principale: *http://fedoraproject.org/wiki/SELinux*.
- Risoluzione dei problemi: *http://fedoraproject.org/wiki/SELinux/Troubleshooting*.
- Fedora SELinux FAQ: *http://docs.fedoraproject.org/selinux-faq/*.
- SELinux Managing Confined Services Guide: *http://docs.fedoraproject.org/selinux-managing-confined-services-guide/*

The UnOfficial SELinux FAQ

http://www.crypt.gen.nz/selinux/faq.html

IRC

Su *Freenode*[11]:

- #selinux
- #fedora-selinux
- #security

[11] http://freenode.net/

Appendice A. Storia delle revisioni

Revisione 1.5 Mon May 10 2010 Scott Radvan *sradvan@redhat.com*

Update and verification for Fedora 13

Revisione 1.4 Mon Aug 31 2009 Scott Radvan *sradvan@redhat.com*

Aggiornamento e verifiche per Fedora 12

Revisione 1.3 Tue May 12 2009 Scott Radvan *sradvan@redhat.com*

Aggiornamento e verifiche per Fedora 11

Revisione 1.2 Mon Jan 19 2009 Murray McAllister *mmcallis@redhat.com*

Aggiornamento dei collegamenti ai siti dell'NSA

Revisione 1.1 Sat Dec 6 2008 Murray McAllister *mmcallis@redhat.com*

Resolving *Red Hat Bugzilla #472986, "httpd does not write to /etc/httpd/logs/"*[1]
Added new section, "6.6. Booleans for Users Executing Applications"
Revisioni di minore importanza

Revisione 1.0 Tue Nov 25 2008 Murray McAllister *mmcallis@redhat.com*

Contenuti della versione iniziale su *http://docs.fedoraproject.org/*