

# Fedora 11

## Linux con Seguridad Mejorada

Guía del Usuario



**Murray McAllister**

**Daniel Walsh**

**Dominick Grift**

**Eric Paris**

**James Morris**

**Scott Radvan**

# Fedora 11 Linux con Seguridad Mejorada

## Guía del Usuario

### Edición 1.3

Autor	Murray McAllister	<a href="mailto:mmcallis@redhat.com">mmcallis@redhat.com</a>
Autor	Daniel Walsh	<a href="mailto:dwalsh@redhat.com">dwalsh@redhat.com</a>
Autor	Dominick Grift	<a href="mailto:domg472@gmail.com">domg472@gmail.com</a>
Autor	Eric Paris	<a href="mailto:eparis@parisplace.org">eparis@parisplace.org</a>
Autor	James Morris	<a href="mailto:jmorris@redhat.com">jmorris@redhat.com</a>
Autor	Scott Radvan	<a href="mailto:sradvan@redhat.com">sradvan@redhat.com</a>

Copyright © 2009 Red Hat, Inc.

Copyright © 2009 Red Hat, Inc. This material may only be distributed subject to the terms and conditions set forth in the Open Publication License, V1.0, (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Fedora and the Fedora Infinity Design logo are trademarks or registered trademarks of Red Hat, Inc., in the U.S. and other countries.

Red Hat and the Red Hat "Shadow Man" logo are registered trademarks of Red Hat Inc. in the United States and other countries.

All other trademarks and copyrights referred to are the property of their respective owners.

Documentation, as with software itself, may be subject to export control. Read about Fedora Project export controls at <http://fedoraproject.org/wiki/Legal/Export>.

Este libro es acerca de la administración y uso del Linux de Seguridad Mejorada.

---

---

<b>Prefacio</b>	<b>v</b>
1. Convenciones del Documento .....	v
1.1. Convenciones Tipográficas .....	v
1.2. Convenciones del documento .....	vii
1.3. Notas y Advertencias .....	viii
2. ¡Necesitamos sus comentarios! .....	viii
<b>1. Información de Marca Comercial</b>	<b>1</b>
1.1. Source Code .....	1
<b>2. Introducción</b>	<b>3</b>
2.1. Beneficios de usar SELinux .....	4
2.2. Ejemplos .....	5
2.3. Arquitectura de SELinux .....	6
2.4. SELinux en otros Sistemas Operativos .....	6
<b>3. Contextos de SELinux</b>	<b>7</b>
3.1. Transiciones de Dominios .....	8
3.2. Contextos de SELinux para los Procesos .....	9
3.3. Contextos de SELinux para los Usuarios .....	10
<b>4. Política Destinado</b>	<b>11</b>
4.1. Procesos Confinados .....	11
4.2. Procesos no Confinados .....	14
4.3. Usuarios Confinados y no Confinados .....	17
<b>5. Trabajando con SELinux</b>	<b>21</b>
5.1. Paquetes de SELinux .....	21
5.2. Qué Archivo Log se usa .....	22
5.3. Archivo de Configuración Principal .....	23
5.4. Habilitando y Deshabilitando SELinux .....	24
5.4.1. Habilitando SELinux .....	24
5.4.2. Deshabilitando SELinux .....	27
5.5. Modos de SELinux .....	28
5.6. Booleanos .....	28
5.6.1. Listando los Booleanos .....	28
5.6.2. Configurando los Booleanos .....	29
5.6.3. Booleanos para NFS y CIFS .....	30
5.7. Contextos de SELinux - Etiquetado de Archivos .....	31
5.7.1. Cambios Temporales: chcon .....	31
5.7.2. Cambios Persistentes: semanage fcontext .....	33
5.8. Los tipos file_t y default_t .....	38
5.9. Montaje de Sistemas de Archivos .....	38
5.9.1. Montajes de Contexto .....	39
5.9.2. Cambio del Contexto Predeterminado .....	39
5.9.3. Montando un Sistema de Archivos NFS .....	40
5.9.4. Montajes NFS Múltiples .....	40
5.9.5. Haciendo Persistente los Contextos de Montajes .....	41
5.10. Mantenición de las Etiquetas de SELinux .....	42
5.10.1. Copia de Directorios y Archivos .....	42
5.10.2. Movimiento de Archivos y Directorios .....	44
5.10.3. Chequeando el Contexto SELinux Predeterminado .....	45
5.10.4. Archivando archivos con tar .....	46
5.10.5. Archivando archivos con tar .....	47

<b>6. Confinando a los Usuarios</b>	<b>51</b>
6.1. Linux y los Mapeos de Usuarios de SELinux .....	51
6.2. Confinando Usuarios Nuevos de Linux: useradd .....	51
6.3. Confinando Usuarios Linux Existentes: semanage login .....	52
6.4. Cambiando el Mapeo Predeterminado .....	54
6.5. xguest: Modo Kiosk .....	55
6.6. Booleanos para que los Usuarios Ejecuten Aplicaciones .....	55
<b>7. Solución a Problemas</b>	<b>57</b>
7.1. Qué pasa cuando el Acceso es Denegado .....	57
7.2. Tres Principales Causas de Problemas .....	58
7.2.1. Problemas de Etiquetados .....	58
7.2.2. ¿Cómo se Ejecutan los Servicios Confinados? .....	59
7.2.3. Evolucionando las Reglas y las Aplicaciones Rotas .....	61
7.3. Corrección de Problemas .....	61
7.3.1. Permisos de Linux .....	61
7.3.2. Posibles Causas de las Negaciones Silenciosas .....	62
7.3.3. Páginas de Manual para Servicios .....	62
7.3.4. Dominios Permisivos .....	63
7.3.5. Búsqueda y Revisión de Negaciones .....	65
7.3.6. Mensajes Crudos de Auditoría .....	67
7.3.7. Mensajes sealert .....	68
7.3.8. Permitiendo el Acceso: audit2allow .....	71
<b>8. Información Adicional</b>	<b>75</b>
8.1. Contributors .....	75
8.2. Other Resources .....	75
<b>A. Revision History</b>	<b>77</b>

---

# Prefacio

La Guía del Usuario de SELinux de Fedora 11 es para gente sin o con mínima experiencia con SELinux. Aunque la experiencia de administración de sistema no es necesario, el contenido de esta guía se escribe para tareas de administración del sistema. Esta guía provee una introducción a los conceptos fundamentales y aplicaciones prácticas de SELinux. Después de leer esta guía debe tener un entendimiento intermedio de SELinux.

Gracias a todos los que nos alentaron, ofrecieron ayuda y lo probaron - la ayuda es muy apreciada. Agradecimientos muy especiales a:

- Dominick Grift, Stephen Smalley y Russell Coker por sus contribuciones, ayuda y paciencia.
- Karsten Wade por su ayuda, al agregar un componente para esta guía en el *Bugzilla de Red Hat*<sup>1</sup> y resolver el sitio web en <http://docs.fedoraproject.org/>.
- Al *Equipo de Infraestructura de Fedora*<sup>2</sup> por proveer el alojamiento.
- Jens-Ulrik Petersen por asegurar que la oficina de Brisbane de Red Hat tenga espejos de Fedora actualizados.

## 1. Convenciones del Documento

Este manual utiliza varias convenciones para resaltar algunas palabras y frases y llamar la atención sobre ciertas partes específicas de información.

En ediciones PDF y de papel, este manual utiliza tipos de letra procedentes de *Liberation Fonts*<sup>3</sup>. Liberation Fonts también se utilizan en ediciones de HTML si están instalados en su sistema. Si no, se muestran tipografías alternativas pero equivalentes. Nota: Red Hat Enterprise Linux 5 y siguientes incluyen Liberation Fonts predeterminadas.

### 1.1. Convenciones Tipográficas

Se utilizan cuatro convenciones tipográficas para llamar la atención sobre palabras o frases específicas. Dichas convenciones y las circunstancias en que se aplican son las siguientes:

#### **Negrita monoespaciado**

Utilizada para resaltar la entrada del sistema, incluyendo comandos de shell, nombres de archivo y rutas. También se utiliza para resaltar teclas claves y combinaciones de teclas.

Para ver el contenido del archivo **my\_next\_bestselling\_novel** en su directorio actual de trabajo, escriba el comando **cat my\_next\_bestselling\_novel** en el intérprete de comandos de shell y pulse **Enter** para ejecutar el comando.

Lo anterior incluye un nombre de archivo, un comando de shell y una tecla clave, todo en negrita-monoespaciado y distinguible gracias al contexto.

Las combinaciones de teclas se pueden distinguir de las teclas claves mediante el guión que conecta cada parte de una combinación de tecla. Por ejemplo:

Pulse **Enter** para ejecutar el comando.

---

<sup>3</sup> <https://fedorahosted.org/liberation-fonts/>

Pulse **Control+Alt+F1** para cambiar a la primera terminal virtual. Pulse **Control+Alt+F7** para volver a su sesión de Ventanas-X.

La primera oración resalta la tecla clave determinada que se debe pulsar. La segunda resalta dos conjuntos de tres teclas claves, cada conjunto presionado simultáneamente.

Si el código fuente es discutido, nombres de clase, métodos, funciones, nombres de variables, y valores de retorno mencionados dentro de un párrafo serán presentados como lo anterior, en **Negrita-monoespaciado**. Por ejemplo:

Las clases de archivo relacionadas incluyen **filename** para sistema de archivos, **file** para archivos y **dir** para directorios. Cada clase tiene su propio conjunto asociado de permisos.

### Negrita proporcional

Esta denota palabras o frases encontradas en un sistema, incluyendo nombres de aplicación; texto de cuadro de diálogo, botones etiquetados, etiquetas de cajilla de verificación y botón de radio; títulos de menú y títulos del sub-menú. Por ejemplo:

Seleccionar **Sistema > Preferencias > Ratón** desde la barra del menú principal para lanzar **Preferencias de Ratón**. En la pestaña de **Botones**, haga clic en la cajilla **ratón de mano izquierda** y luego haga clic en **Cerrar** para cambiar el botón principal del ratón de la izquierda a la derecha (adecuando el ratón para la mano izquierda).

Para insertar un caracter especial en un archivo **gedit**, seleccione desde la barra del menú principal **Aplicaciones > Accesorios > Mapa de caracteres**. Luego, desde la barra del menú elija **Búsqueda > Hallar... mapa de caracteres**, teclee el nombre del caracter en el campo de **Búsqueda** y haga clic en **Siguiente**. El caracter buscado se resaltará en la **Tabla de caracteres**. Haga doble clic en este caracter resaltado para colocarlo en el campo de **Texto para copiar** y luego haga clic en el botón de **Copiar**. Ahora regrese a su documento y elija **Editar > Pegar** desde la barra de menú de **gedit**.

El texto anterior incluye nombres de aplicación; nombres de menú de todo el sistema y elementos; nombres de menú de aplicaciones específicas y botones y texto hallados dentro de una interfaz GUI, todos presentados en negrita proporcional y distinguibles por contexto.

Observe la **>** abreviatura utilizada para indicar recorrido a través de un menú y sus sub-menús. Esto es para evitar la dificultad de seguir el método 'Seleccionar **Ratón** desde el sub-menú **Preferencias** en el menú de **Sistema** de la barra de menú principal'.

### ***Itálicas-negrita monoespaciado o Itálicas-negrita proporcional***

Negrita monoespaciado o Negrita proporcional, la adición de itálicas indica texto reemplazable o variable. Las itálicas denotan texto que usted no escribe literalmente o texto mostrado que cambia dependiendo de la circunstancia. Por ejemplo:

Para conectar a una máquina remota utilizando ssh, teclee **ssh nombredeusuario@dominio.nombre** en un intérprete de comandos de shell. Si la máquina remota es **example.com** y su nombre de usuario en esa máquina es john, teclee **ssh john@example.com**.

El comando **mount -o remount file-system** remonta el sistema de archivo llamado. Por ejemplo, para volver a montar el sistema de archivo **/home**, el comando es **mount -o remount /home**.

Para ver la versión de un paquete actualmente instalado, utilice el comando `rpm -q paquete`. Éste entregará el resultado siguiente: ***paquete-versión-lanzamiento***.

Observe las palabras en itálicas- negrita sobre — nombre de usuario, domain.name, sistema de archivo, paquete, versión y lanzamiento. Cada palabra es un marcador de posición, tanto para el texto que usted escriba al ejecutar un comando como para el texto mostrado por el sistema.

Aparte del uso estándar para presentar el título de un trabajo, las itálicas denotan el primer uso de un término nuevo e importante. Por ejemplo:

cuando el servidor Apache HTTP acepta solicitudes, envía procesos hijos o hilos para manejarlos. Este grupo de procesos hijos o hilos se conoce como un *server-pool*. Bajo el servidor HTTP de Apache 2.0, la responsabilidad para crear o mantener estos server-pool se resume en un grupo de módulos llamado *Módulos de multi-procesamiento (MPMs)*. A diferencia de otros módulos, sólo un módulo del grupo MPM puede ser cargado por el servidor HTTP de Apache.

## 1.2. Convenciones del documento

Dos, usualmente de varias líneas, los tipos de datos se distinguen visualmente del texto circundante.

Salida enviada a una terminal está establecida en tipo romano monoespaciado y presentada así:

```
libros      Escritorio  documentación  borradores  mss      fotos      cosas
  svn
libros_pruebas  Escritorio1  descargas      imágenes  notas  scripts  svgs
```

Los listados de código fuente también se establecen en romano monoespaciado, pero se presentan y resaltan de la siguiente manera:

```
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. Notas y Advertencias

Finalmente, utilizamos tres estilos visuales para llamar la atención sobre la información que de otro modo se podría pasar por alto.



### Nota

Una nota es una sugerencia, atajo o enfoque alternativo que se tiene a mano para la tarea. Ignorar una nota no debería tener consecuencias negativas, pero podría perderse de algunos trucos que pueden facilitarle las cosas.



### Importante

Los cuadros de importante dan detalles de cosas que se pueden pasar por alto fácilmente: cambios de configuración únicamente aplicables a la sesión actual, o servicios que necesitan reiniciarse antes de que se aplique una actualización. Ignorar estos cuadros de importante no ocasionará pérdida de datos, pero puede causar enfado y frustración.



### Advertencia

Las advertencias no deben ignorarse. Ignorarlas muy probablemente ocasionará pérdida de datos.

## 2. ¡Necesitamos sus comentarios!

Si encuentra un error tipográfico en este manual o si sabe de alguna manera de mejorarlo, nos gustaría escuchar sus sugerencias. Por favor complete un reporte en Bugzilla: <http://bugzilla.redhat.com/bugzilla/> usando el producto **Fedora Documentation**.

Cuando envíe un reporte de error no olvide mencionar el identificador del manual: *selinux-user-guide*

Si tiene una sugerencia para mejorar la documentación, intente ser tan específico como sea posible cuando describa su sugerencia. Si ha encontrado un error, por favor incluya el número de sección y parte del texto que rodea el error para que podamos encontrarlo más fácilmente.



---

# Información de Marca Comercial

Linux® es una marca comercial registrada de Linus Torvalds en los EEUU y en otros países.

UNIX es una marca comercial registrada de El Grupo Abierto.

Type Enforcement (Obligación de Tipos) es una marca comercial de Secure Computing, LLC, una subsidiaria de McAfee, Inc., registrada en los EEUU y en otros países. Ni McAfee ni Secure Computing, LLC, ha consentido el uso o referencia de esta marca comercial para el autor fuera de esta guía.

Apache es una marca comercial de La Fundación de Software Apache.

MySQL es una marca comercial o marca comercial registrada de MySQL AB en los EEUU y en otros países.

## 1.1. Source Code

The XML source for this guide is available at <http://svn.fedorahosted.org/svn/selinuxguide/>

---

---

# Introducción

Los Archivos, tales como directorios y dispositivos, se llaman objetos. Los procesos, tal como un comando que ejecuta el usuario o la aplicación Mozilla® Firefox®, se llaman sujetos. La mayoría de los sistemas operativos usan un sistema de Control de Acceso Discrecional (DAC), que controla cómo interactúan los sujetos con los objetos, y cómo los sujetos interactúan entre sí. En sistemas operativos que usan DAC, los usuarios controlan los permisos de archivos (objetos) de los que son dueños. Por ejemplo, en sistemas operativos Linux®, los usuarios pueden hacer sus directorios legibles para el resto del mundo, dando a los usuarios y procesos (sujetos) acceso a información potencialmente sensible.

Los mecanismos DAC son fundamentalmente inadecuados para una fuerte seguridad del sistema. Las decisiones de acceso DAC son sólo basadas en la identidad del usuario y su propiedad, ignorando información de seguridad relevante tal como el rol del usuario, la función y la confiabilidad del programa, y la sensibilidad e integridad de los datos. Cada usuario tiene completa discreción sobre sus archivos, haciendo imposible aplicar una política de seguridad a nivel de sistema. Más aún, cada programa que ejecuta un usuario hereda todos los permisos garantizados al usuario y es libre de cambiar el acceso de archivos del usuario, por lo que no se provee una protección contra software malicioso. Muchos servicios del sistema y programas privilegiados deben ejecutarse con privilegios más allá de lo que realmente necesitan, por lo que una brecha en cualquiera de estos programas se puede explotar para obtener acceso completo al sistema.<sup>1</sup>

El siguiente es un ejemplo de permisos usados en sistemas operativos Linux que no corren el Linux de Seguridad Mejorada (SELinux). Los permisos en estos ejemplos pueden diferir de su sistema. Use el comando `ls -l` para ver los permisos de archivos:

```
$ ls -l archivo1
-rwxrw-r-- 1 usuario1 grupo1 0 2009-04-30 15:42 archivo1
```

Los primeros tres bits de permisos, `rwx`, controlan el acceso que el usuario Linux `usuario1` (en este caso, el dueño) tiene para el `archivo1`. Los siguientes tres bits de permisos, `rw-`, controlan el acceso que el grupo Linux `grupo1` tiene para el `archivo1`. Los últimos tres bits de permisos, `r--`, controlan el acceso que todo el mundo tiene para el `archivo1`, que incluyen a todos los usuarios y procesos.

Linux de Seguridad Mejorada (SELinux por Security-Enhanced Linux) agrega Control de Acceso Obligatorio (MAC en inglés) al kernel de Linux, y se habilita por defecto en Fedora. Una arquitectura MAC de propósito general necesita la habilidad de aplicar una política de seguridad puesta administrativamente sobre todos los procesos y archivos del sistema, basando decisiones en las etiquetas que contienen información variada relevante para la seguridad. Cuando se implementa apropiadamente, hace que un sistema se autodefienda adecuadamente y ofrece soporte crítico para la seguridad de aplicaciones protegiéndolas de la manipulación y sobrepaso. MAC provee una separación fuerte de aplicaciones que permite la ejecución a salvo de aplicaciones no confiables. Su habilidad para limitar los privilegios asociados con la ejecución de procesos limita el ámbito del potencial daño que puede resultar de la explotación de vulnerabilidades en aplicaciones y servicios del sistema. MAC hace que la información esté protegida de los usuarios legítimos

---

<sup>1</sup> "Integración del Soporte Flexible para Políticas de Seguridad en el Sistema Operativo Linux", de Peter Loscocco y Stephen Smalley. Este paper fue preparado originalmente por la Agencia de Seguridad Nacional y está, consecuentemente, en el dominio público. Vaya al [paper original](http://www.nsa.gov/research/_files/selinux/papers/freenix01/index.shtml) [http://www.nsa.gov/research/\_files/selinux/papers/freenix01/index.shtml] para mas detalles y el documento tal como fue lanzado la primera vez. Cualquier edición y cambio fue hecho por Murray McAllister.

con autorización limitada, así como de usuarios autorizados que involuntariamente ejecutaron aplicaciones maliciosas.<sup>2</sup>

El siguiente es un ejemplo de las etiquetas que contienen información de seguridad relevante que se usa en los procesos, usuarios Linux y archivos, en sistemas operativos Linux que corren SELinux. Esta información se llama contexto de SELinux y se visualiza usando el comando `ls -Z`:

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

En este ejemplo, SELinux provee un usuario (`unconfined_u`), un rol (`object_r`), un tipo (`user_home_t`), y un nivel (`s0`). Esta información se usa para tomar decisiones sobre el control de acceso. Con DAC, el acceso se controla basado sólo en los IDs de usuarios y grupos de Linux. Las reglas de políticas de SELinux se chequean después de las reglas DAC. Las reglas de políticas de SELinux no se usan si las reglas DAC niegan el acceso al principio.

### Linux y los Usuarios SELinux

En sistemas operativos Linux que corren SELinux, hay usuarios Linux así como usuarios SELinux. Los usuarios SELinux son parte de la política de SELinux. Los usuarios Linux se mapean a usuarios SELinux. Para evitar confusión, esta guía usa "usuarios Linux" y "usuarios SELinux" para diferenciar entre ambos.

## 2.1. Beneficios de usar SELinux

- Todos los procesos y archivos se etiquetan con un tipo. Un tipo define un dominio para los procesos y un tipo para los archivos. Los procesos se separan entre sí corriéndolos en sus propios dominios, y las reglas de políticas de SELinux define cómo interactúan los procesos con los archivos, así como la forma en que interactúan entre sí. El acceso sólo se permite si existe una regla de política de SELinux que específicamente lo permita.
- Control de acceso más fino. Yendo un paso más allá de los permisos tradicionales de UNIX® que se controlan a discreción del usuario y se basa en los IDs de usuario y de grupos de Linux, las decisiones de accesos de SELinux se basan en toda la información disponible, tales como un usuario SELinux, el rol, el tipo y, opcionalmente, un nivel.
- La política de SELinux se define administrativamente, obligando a todo el sistema, y no se pone a discreción del usuario.
- Vulnerabilidad reducida para ataques de escalamiento de privilegios. Un ejemplo: dado que los procesos corren en dominios, y son por lo tanto separados entresi, y las reglas de la política de SELinux definen cómo los procesos acceden a los archivos y a otros procesos, si un proceso queda comprometido, el atacante sólo tiene acceso a las funciones normales de ese proceso, y a archivos a los que el proceso tiene configurado que puede acceder. Por ejemplo, si el Servidor HTTP Apache es comprometido, un atacante no puede usar el proceso para leer archivos en los directorios home de los usuarios, a menos que una regla específica de la política de SELinux haya sido agregada o configurada para permitir tal acceso.

---

<sup>2</sup> "Consiguiendo Objetivos de Seguridad Críticos con Linux de Seguridad Mejorada", por Peter Loscocco y Stephen Smalley. Este paper fue preparado originalmente por la Agencia de Seguridad Nacional y está, consecuentemente, en el dominio público. Vaya al [paper original](http://www.nsa.gov/research/_files/selinux/papers/ottawa01/index.shtml) [http://www.nsa.gov/research/\_files/selinux/papers/ottawa01/index.shtml] para detalles y por el documento tal como fue lanzado la primera vez. Cualquier edición y cambio fue hecho por Murray McAllister.

- Se linux se puede usar para asegurar la confidencialidad e integridad de los datos, así como proteger los procesos de entradas no confiables.

SELinux no es:

- software antivirus.
- un reemplazo para las contraseñas, cortafuegos y otros sistemas de seguridad.
- una solución todo en uno.

SELinux está diseñado para mejorar las soluciones de seguridad existentes, no reemplazarlas. Aún cuando corra SELinux, siga las buenas prácticas de seguridad, tales como mantener el software actualizado, usar contraseñas difíciles de adivinar, cortafuegos y demás.

## 2.2. Ejemplos

Los siguientes ejemplos demuestran cómo SELinux aumenta la seguridad:

- la acción por defecto es denegar. Si una regla de la política de SELinux no existe para permitir el acceso, como para que un proceso pueda abrir un archivo, el acceso es denegado.
- SELinux puede confinar a usuarios Linux. Existe un número de usuarios SELinux confinados. Los usuarios Linux se pueden mapear a usuarios SELinux para sacar ventaja de los usuarios SELinux confinados. Por ejemplo, mapear un usuario Linux al usuario SELinux `user_u`, resulta en un usuario Linux que no puede ejecutar (a menos que se configure de otra forma) aplicaciones con el ID del usuario (`setuid`), tales como **sudo** y **su**, así como previene la ejecución de archivos y aplicaciones en sus directorios de inicio (`home`)- si se configura, esto evita que los usuarios ejecuten archivos maliciosos desde sus directorios de inicio.
- separación de procesos. Los procesos que corren en sus propios dominios, previenen que estos accedan a archivos usados por otros procesos, así como el acceso a otros procesos. Por ejemplo, cuando se ejecuta SELinux, a menos que se configure de otra forma, un atacante no puede comprometer un servidor Samba, y luego usar ese servidor Samba para leer y escribir archivos usados por otros procesos, tales como las bases de datos usadas por MySQL®.
- ayuda a limitar los daños hechos por errores en la configuración. Los servidores [Sistema de Nombres de Dominio \(DNS\)](#)<sup>3</sup> pueden replicar información entre si. Esto se conoce como transferencia de zona. Los atacantes pueden usar las transferencias de zonas para actualizar los servidores de DNS con información falsa. Cuando se ejecuta el servidor de DNS [Berkeley Internet Name Domain \(BIND\)](#)<sup>4</sup> en Fedora 11, aún si un administrador se olvida de limitar qué servidores pueden realizar las transferencias de zona, la política predeterminada de SELinux evita que los archivos de zonas<sup>5</sup> sean actualizados por transferencias, por el demonio BIND `named` y por otros procesos.
- vaya al artículo de [Red Hat® Magazine](#)<sup>6</sup>, [Informe de riesgo: Tres años del Linux para Empresas de Red Hat 4](#)<sup>78</sup>, para explotaciones que fueron restringidas gracias a la política destinada de SELinux en el Linux® para Empresas de Red Hat® 4.
- vaya al artículo en [LinuxWorld.com](#)<sup>9</sup> article, [Un cinturón de seguridad para software servidor: SELinux bloquea explotaciones del mundo real](#)<sup>1011</sup>, para un poco de información sobre SELinux, y sobre varias explotaciones que SELinux evita.

- vaya al blog de James Morris [SELinux mitiga la vulnerabilidad de root en OpenPegasus](#)<sup>12</sup>, para información acerca de una brecha en [OpenPegasus](#)<sup>13</sup> que fue mitigada por SELinux tal como fue distribuido con el Linux para Empresas de Red Hat 4 y 5.

El sitio web de [Tresys Technology](#)<sup>14</sup> tiene una sección de [Noticias de Migración a SELinux](#)<sup>15</sup> (en la parte derecha), que lista los ataques recientes que fueron mitigados o prevenidos por SELinux.

### 2.3. Arquitectura de SELinux

SELinux es un módulo de seguridad de Linux que se construye dentro del kernel de Linux. SELinux se maneja por reglas de políticas cargables. Cuando un acceso de seguridad relevante se lleva a cabo, tal como un proceso que trata de abrir un archivo, la operación es interceptada por SELinux en el kernel. Si una regla de política de SELinux permite la operación, continúa, sino, la operación se bloquea y el proceso recibe un error.

Las decisiones de SELinux, tales como permitir o negar accesos, son cacheadas. Este caché se conoce como Caché Vector de Acceso (AVC). Las decisiones de cacheado disminuye la necesidad de que las reglas de políticas de SELinux sean chequeadas muy a menudo, lo que mejora la performance. Las reglas de políticas de SELinux no tienen efecto si las reglas DAC niegan el acceso primero.

### 2.4. SELinux en otros Sistemas Operativos

Vaya a la siguiente información sobre cómo correr SELinux en sistemas operativos:

- Gentoo endurecido: <http://www.gentoo.org/proj/en/hardened/selinux/selinux-handbook.xml>.
- Debian: <http://wiki.debian.org/SELinux>.
- Ubuntu: <https://wiki.ubuntu.com/SELinux> and <https://help.ubuntu.com/community/SELinux>.
- Linux para Empresas de Red Hat: [Guía de Despliegue del Linux para Empresas de Red Hat](#)<sup>16</sup> y la [Guía de SELinux para el Linux para Empresas de Red Hat](#)<sup>17</sup>.
- Fedora: <http://fedoraproject.org/wiki/SELinux> y el [FAQ de SELinux de Fedora Core 5](#)<sup>18</sup>.

---

<sup>14</sup> <http://www.tresys.com/>

<sup>15</sup> <http://www.tresys.com/innovation.php>

---

# Contextos de SELinux

Los procesos y archivos se etiquetan con un contexto SELinux que contiene información adicional, tal como un usuario SELinux, rol, tipo y, opcionalmente, un nivel. Cuando se ejecuta SELinux, toda esta información se usa para tomar decisiones de control de acceso. En Fedora 11, SELinux provee una combinación de Control de Acceso Basado en Roles (RBAC en inglés), Obligación de Tipos® (TE en inglés), y, opcionalmente, Seguridad Multi-nivel (MLS en inglés).

El siguiente es un ejemplo de contexto de SELinux. Los contextos de SELinux se usan en procesos, usuarios de Linux y archivos, en sistemas operativos Linux que corren SELinux. Use el comando **ls -Z** para ver el contexto SELinux de archivos y directorios:

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

Los contextos de SELinux siguen la *SELinux nombre-de-usuario:role:type:level* sintaxis:

## Usuario SELinux

La identidad de usuario SELinux es una identidad conocida para la política a la que se le autoriza un conjunto específico de roles, y un rango específico de MLS. Cada usuario Linux se mapea a un usuario SELinux vía la política de SELinux. Esto permite a los usuarios de Linux heredar las restricciones de los usuarios SELinux. La identidad del usuario SELinux mapeado se usa en el contexto de SELinux para procesos en esa sesión, para agrupar a qué roles y niveles pueden ingresar. Ejecute el comando **semanage login -l** como usuario root de Linux para ver una lista de mapeos entre cuentas de usuarios Linux y SELinux:

```
# /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

La salida puede ser distinta de sistema a sistema. La columna Nombre de Ingreso lista los usuarios Linux, y la columna Usuario SELinux lista qué a usuario SELinux es mapeado un usuario Linux. Para procesos, el usuario SELinux limita qué roles y niveles son accesibles. La última columna, Rango MLS/MCS, es el nivel usado por la Seguridad Multi Nivel (MLS) y por la Seguridad Multi Categoría (MCS). Los niveles se describen brevemente más adelante.

## rol

Parte de SELinux es el modelo de seguridad de Control de Acceso Basado en Roles (RBAC). El rol es un atributo de RBAC. Los usuarios de SELinux son autorizados para ciertos roles y los roles son autorizados para ciertos dominios. Los roles sirven como un intermediario entre dominios y usuarios SELinux. Los roles en los que se puede ingresar determinan los dominios a los que se ingresan - al final, esto controla los tipos de objetos que se pueden acceder. Esto ayuda a reducir la vulnerabilidad de ataques de escalada de privilegios.

## tipo

El tipo es un atributo de la Obligación de Tipos. El tipo define un dominio para procesos y un tipo para los archivos. Las reglas de políticas de SELinux define cómo se acceden los tipos entre si,

ya sea un dominio que accede a un tipo o un dominio que accede a otro dominio. Los acceso sólo se permiten si existe una regla de política de SELinux que lo permita.

### nivel

El nivel es un atributo de MLS y la Seguridad Multi Categoría (MCS). Un rango MLS es un par de niveles, escrito como *bajonivel-altonivel* si los niveles son distintos, o *bajonivel* si los niveles son idénticos ( $s_0 - s_0$  es lo mismo que  $s_0$ ). Cada nivel es un par sensible a categorías, donde las categorías son opcionales. Si no hay categorías, el nivel se escribe como *sensibilidad:conjunto-de-categoría*. Si no hay categorías, se escribe como *sensibilidad*.

Si el conjunto de categoría es una serie contigua, se puede abreviar. Por ejemplo,  $c_0 . c_3$  es lo mismo que  $c_0, c_1, c_2, c_3$ . El archivo `/etc/selinux/targeted/setrans.conf` mapea los niveles ( $s_0 : c_0$ ) a una forma legible al humano (`CompanyConfidential`). No edite `setrans.conf` con un editor de texto: use `semanage` para hacer cambios. Vaya a la página de manual de `semanage(8)` para más información. En Fedora 11, la política destinada obglica MCS, y en MCS hay sólo una sensibilidad, la  $s_0$ . MCS en Fedora 11 soporta 1024 categorías distintas:  $c_0$  a  $c_{1023}$ .  $s_0 - s_0 : c_0 . c_{1023}$  es la sensibilidad  $s_0$  y autorizado para todas las categorías.

MLS aplica el [Modelo de Acceso Obligatorio Bell-LaPadula](#)<sup>1</sup>, y se usa en entornos de Perfil de Protección de Seguridad Etiquetada (LSPP en inglés). Para usar las restricciones MLS, instale el paquete `selinux-policy-mls`, y configure a MLS como la política de SELinux predeterminada. La política MLS puesta en Fedora mite muchos dominios de programa que no eran parte de la configuración evaluada, y por lo tanto, MLS es inútil en un escritorio (no hay soporte para el Sistema de Ventanas X); sin embargo, una política MLS se puede incluir desde los [desarrolladores de la Política de Referencia de SELinux](#)<sup>2</sup> que incluya todos los dominios de programa.

## 3.1. Transiciones de Dominios

Un proceso transiciona de un dominio a otro ejecutando una aplicación que tiene el tipo `entrypoint` en el nuevo dominio. Los permisos `entrypoint` se usan en las políticas de SELinux, y controlan qué aplicaciones pueden usarse para ingresar a un dominio. El siguiente ejemplo muestra una transición de dominio.

1. Un usuario desea cambiar su contraseña. Para cambiar sus contraseñas, corren la aplicación `passwd`. El ejecutable `/usr/bin/passwd` está etiquetado con el 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
```

La aplicación `passwd` accede `/etc/shadow`, que está etiquetado con el tipo `shadow_t`:

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

2. Una regla de política de SELinux dice que los procesos que se ejecutan en el dominio `passwd_t` no pueden leer y escribir archivos etiquetados con el tipo `shadow_t`. El tipo `shadow_t` sólo se aplica a archivos que necesitan un cambio de contraseñas. Esto incluye a `/etc/gshadow`, `/etc/shadow`, y sus archivos de respaldo.

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Bell-LaPadula\\_model](http://en.wikipedia.org/wiki/Bell-LaPadula_model)

<sup>2</sup> <http://oss.tresys.com/projects/refpolicy>



3. Una regla de política de SELinux fija que el dominio `passwd_t` tiene permiso de `entrypoint` al tipo `passwd_exec_t`.
4. Cuando un usuario ejecuta la aplicación `/usr/bin/passwd`, el proceso shell del usuario transiciona al dominio `passwd_t`. Con SELinux, dado que la acción por defecto es negar, y existe una regla que permite (entre otras cosas) a aplicaciones que corran en el dominio `passwd_t` acceder a archivos etiquetados con el tipo `shadow_t`, se le permite a `passwd` acceder a `/etc/shadow`, y actualizar la contraseña del usuario.

Este ejemplo no es exhaustivo, y se usa como un ejemplo básico para explicar la transición de dominio. Aunque hay una regla actual que permite a sujetos corriendo en el dominio `passwd_t` acceder objetos etiquetados con el tipo de archivo `shadow_t`, otras reglas de política de SELinux se deben cumplir para que el sujeto pueda transicionar a un nuevo dominio. En este ejemplo, la Obligación de Tipo asegura:

- el dominio `passwd_t` sólo se puede ingresar ejecutando una aplicación con la etiqueta del tipo `passwd_exec_t`; sólo pueden ejecutar desde bibliotecas compartidas autorizadas, tales como las del tipo `lib_t`; y no pueden ejecutar ninguna otra aplicación.
- sólo los dominios autorizados, tales como `passwd_t`, pueden escribir en archivos con la etiqueta del tipo `shadow_t`. Aún si otros procesos corren con privilegios de superusuario, esos procesos no podrán escribir archivos etiquetados con el tipo `shadow_t`, porque no están corriendo en el dominio `passwd_t`.
- sólo los dominios autorizados pueden transicionar al dominio `passwd_t`. Por ejemplo, el proceso `sendmail` corriendo en el dominio `sendmail_t` no tiene una razón legítima para ejecutar `passwd`; por lo tanto, no puede transicionar nunca al dominio `passwd_t`.
- los procesos que se ejecutan en el dominio `passwd_t` sólo pueden leer y escribir a tipos autorizados, tales como archivos etiquetados con los tipos `etc_t` o `shadow_t`. Esto impide a la aplicación `passwd` de ser modificada para leer o escribir en archivos arbitrarios.

## 3.2. Contextos de SELinux para los Procesos

Use el comando `ps -eZ` para ver los contextos de SELinux para los procesos. Por ejemplo:

1. Abra una terminal, como la de **Aplicaciones** → **Herramientas del Sistema** → **Terminal**.
2. Ejecute el comando `/usr/bin/passwd`. No ingrese una nueva contraseña.
3. Abra una nueva pestaña, u otra terminal, y ejecute el comando `ps -eZ | grep passwd`. La salida es similar a la siguiente:

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

4. En la primer pestaña, presione **Ctrl+C** para cancelar la aplicación `passwd`.

En este ejemplo, cuando la aplicación `/usr/bin/passwd` (etiquetada con el tipo `passwd_exec_t`) se ejecuta, el proceso shell del usuario transiciona al dominio `passwd_t`. Recuerde: el tipo define un dominio para procesos y un tipo para archivos.

Use el comando **ps -eZ** para ver los contextos SELinux de los procesos en ejecución. El siguiente es un ejemplo limitado de la salida, y puede cambiar en su sistema:

```
system_u:system_r:setroubleshootd_t:s0 1866 ? 00:00:08 setroubleshootd
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
```

El rol `system_r` se usa para procesos de sistema, como los demonios. El tipo obligatorio los separa luego en dominios.

### 3.3. Contextos de SELinux para los Usuarios

Use el comando **id -Z** para ver el contexto SELinux asociado con su usuario Linux:

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

En Fedora 11, los usuarios Linux corren no confinados por defecto. Este contexto de SELinux muestra que el usuario Linux se mapea al usuario SELinux `unconfined_u`, corriendo con el rol `unconfined_r` y en el dominio `unconfined_t`. `s0-s0` es un rango MLS, que en este caso, es el mismo que `s0`. Las categorías a las que el usuario tiene acceso se definen por `c0.c1023`, que son todas las categorías (`c0` a `c1023`).

---

# Política Destinado

La política destinada es la política predeterminada de SELinux usada en Fedora 11. Cuando la política destinada está en uso, los procesos son destinados a correr en dominios confinados, y los procesos que no son destinados a correr en un dominio corren en un dominio no confinado. Por ejemplo, por defecto, los usuarios que ingresaron corren en el dominio `unconfined_t`, y los procesos del sistema iniciado por `init` corren en el dominio `initrc_t` - ambos dominios están no confinados.

Unconfined domains (as well as confined domains) are subject to executable and writeable memory checks. By default, subjects running in an unconfined domain can not allocate writeable memory and execute it. This reduces vulnerability to [buffer overflow attacks](#)<sup>1</sup>. These memory checks are disabled by setting Booleans, which allow the SELinux policy to be modified at runtime. Boolean configuration is discussed later.

## 4.1. Procesos Confinados

Casi todos los servicios que escuchan en la red se confinan en Fedora 11. También, la mayoría de los procesos que corren como usuario `root` de Linux y realizan tareas para los usuarios, tales como la aplicación `passwd`, están confinadas. Cuando un proceso se confina, corre en su propio dominio, tal como el proceso `httpd` ejecutándose en el dominio `httpd_t`. Si un proceso confinado es comprometido por un atacante, dependiendo de la configuración de política de SELinux, el acceso del atacante a recursos y el posible daño que puedan causar es limitado.

El siguiente ejemplo muestra cómo previene SELinux la lectura por parte del Servidor HTTP Apache (`httpd`) de archivos que no se etiquetaron correctamente, tales como archivos a usar con Samba. Este es un ejemplo, y no se debe usar en producción. Asume que los paquetes `httpd`, `wget`, `setroubleshoot-server` y `audit` están instalados, que se usa la política destinada de SELinux y que SELinux está corriendo en modo obligatorio:

1. Ejecute el comando `sestatus` para confirmar que SELinux está activado, se ejecuta en modo obligatorio y que la política destinada se está usando:

```
$ /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 is returned when SELinux is enabled. Current mode: enforcing is returned when SELinux is running in enforcing mode. Policy from config file: targeted is returned when the SELinux targeted policy is used.

2. Como usuario `root` de Linux, ejecute el comando `touch /var/www/html/prueba` para crear un archivo.
3. Ejecute el comando `ls -Z /var/www/html/prueba` para ver el contexto SELinux:

---

<sup>1</sup> [http://en.wikipedia.org/wiki/Buffer\\_overflow](http://en.wikipedia.org/wiki/Buffer_overflow)

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

Por defecto, los usuarios Linux corren no confinados en Fedora 11, razón por la cual el archivo **prueba** se etiqueta con el usuario SELinux `unconfined_u`. RBAC se usa para procesos, y no para archivos. Los roles no tienen significado para archivos - el rol `object_r` es un rol genérico usado para archivos (en sistemas de archivos persistentes y de red). Bajo el directorio `/proc/`, los archivos relacionados con procesos pueden usar el rol `system_r`.<sup>2</sup> El tipo `httpd_sys_content_t` permite al proceso `httpd` acceder a este archivo.

4. Como usuario `root` de Linux, ejecute el comando **`service httpd start`** para iniciar el proceso `httpd`. La salida es como sigue si `httpd` inicia con éxito:

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

5. Cambie al directorio donde su usuario Linux tenga acceso de escritura y ejecute el comando **`wget http://localhost/prueba`**. A menos que hubieran cambios en la configuración predeterminada, este comando tiene éxito:

```
--2009-05-06 23:00:01-- http://localhost/prueba
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: `prueba'

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

2009-05-06 23:00:01 (0.00 B/s) - `prueba' saved [0/0]
```

6. El comando **`chcon`** reetiqueta archivos; sin embargo, tales cambios de etiquetas no sobreviven cuando el sistema se reetiqueta. Para que los cambios sobrevivan un reetiquetado del sistema, use el comando **`semanage`**, que se discute más adelante. Como usuario `root` de Linux, corra el siguiente comando para cambiar el tipo a un tipo usado por Samba:

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

Ejecute el comando **`ls -Z /var/www/html/prueba`** para ver los cambios:

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

7. Nota: los permisos de DAC actuales permiten al proceso `httpd` acceder al **prueba**. Cambie al directorio donde el usuario Linux tenga permiso de escritura y ejecute el comando **`wget http://localhost/prueba`**. A menos que hayan cambios en la configuración predeterminada, este comando fallará:

```
--2009-05-06 23:00:54-- http://localhost/prueba
```

```
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 403 Forbidden
2009-05-06 23:00:54 ERROR 403: Forbidden.
```

8. Como usuario root de Linux, corra el comando **rm -i /var/www/html/prueba** para borrar **prueba**.
9. Si no necesita que corra httpd, como usuario root de Linux corra el comando **service httpd stop** para detener a httpd:

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

Este ejemplo muestra la seguridad adicional agregada por SELinux. Aunque las reglas de DAC permitieron al proceso httpd acceder a **prueba** en el paso 7, dado que estaba etiquetado con un tipo al que el proceso httpd no tenía acceso, SELinux negó el acceso. Después del paso 7, un error similar al siguiente se guarda en **/var/log/messages**:

```
May 6 23:00:54 localhost setroubleshoot: SELinux is preventing httpd
(httpd_t) "getattr"
to /var/www/html/prueba (samba_share_t). For complete SELinux messages.
run sealert -l c05911d3-e680-4e42-8e36-fe2ab9f8e654
```

Archivos log previos pueden usar el formato **/var/log/messages.YYYYMMDD**. Cuando se ejecuta **syslog-ng**, los archivos log previos pueden usar el formato **/var/log/messages.X**. Si los procesos setroubleshootd y auditd están ejecutándose, errores similares a los siguientes se registran en **/var/log/audit/audit.log**:

```
type=AVC msg=audit(1220706212.937:70): avc: denied { getattr }
for pid=1904 comm="httpd" path="/var/www/html/prueba"
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)
```

También, un error similar al siguiente se registra en **/var/log/httpd/error\_log**:

```
[Wed May 06 23:00:54 2009] [error] [client 127.0.0.1] (13)Permission
denied: access to /prueba denied
```



### Nota

En Fedora 11, los paquetes *setroubleshoot-server* y *audit* se instalan por defecto. Estos paquetes incluyen los demonios *setroubleshootd* y *auditd* respectivamente. Estos demonios se ejecutan por defecto. Detener alguno de ellos provoca un cambio de en

dónde se graban las negaciones de SELinux. Vaya a *Sección 5.2, “Qué Archivo Log se usa”* para más información.

### 4.2. Procesos no Confinados

Los procesos no confinados corren en dominios no confinados, por ejemplo, los programas `init` (del arranque) corren en el dominio no confinado `initrc_t`, los procesos no confinados del kernel corren en el dominio `kernel_t` y los usuarios no confinados de Linux corren en el dominio `unconfined_t`. Para procesos no confinados, las reglas de la política de SELinux son aplicadas, pero hay reglas de la política que permiten que los procesos se ejecuten en dominios no confinados tengan casi todos los accesos. Los procesos que corren en dominios no confinados terminan usando exclusivamente las reglas DAC. Si un proceso no confinado es comprometido, SELinux no impide que un atacante gane acceso a los recursos del sistema y a los datos, pero, por supuesto, las reglas DAC todavía se usan. SELinux es una mejora de seguridad sobre las reglas DAC - no las reemplaza.

El siguiente ejemplo muestra cómo el Servidor HTTP Apache (`httpd`) puede acceder datos que se suponen son para ser usados por Samba, cuando se corre no confinado. Nota: en Fedora 11, el proceso `httpd` corre en el dominio confinado `httpd_t` por defecto. Este es un ejemplo, y no debe usarse en producción. Se asume que los paquetes `httpd`, `wget`, `setroubleshoot-server`, y `audit` están instalados, que se usa la política destinada de SELinux y que SELinux corre en modo obligatorio:

1. Ejecute el comando `sestatus` para confirmar que SELinux está activado, se ejecuta en modo obligatorio y que la política destinada se está usando:

```
$ /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 is returned when SELinux is enabled. Current mode: enforcing is returned when SELinux is running in enforcing mode. Policy from config file: targeted is returned when the SELinux targeted policy is used.

2. Como usuario `root` de Linux, corra el comando `touch /var/www/html/prueba2` para crear un archivo.
3. Ejecute el comando `ls -Z /var/www/html/prueba2` para ver el contexto SELinux:

```
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/
www/html/prueba2
```

Por defecto, los usuarios Linux corren no confinados en Fedora 11, razón por la cual el archivo `prueba2` está etiquetado con el usuario SELinux `unconfined_u`. RBAC se usa para procesos, no para archivos. Los roles no tienen un significado para archivos - el rol `object_r` es el rol genérico que se usa para archivos (en almacenamiento persistente y sistemas de archivos de red). Bajo el directorio `/proc/`, los archivos relacionados con procesos pueden usar el rol `system_r`.<sup>3</sup> El tipo `httpd_sys_content_t` permite al proceso `httpd` acceder a este archivo.

4. El comando **chcon** reetiqueta archivos; sin embargo, tales cambios de etiquetas no sobreviven cuando el sistema se reetiqueta. Para que los cambios sobrevivan un reetiquetado del sistema, use el comando **semanage**, que se discute más adelante. Como usuario root de Linux, corra el siguiente comando para cambiar el tipo a un tipo usado por Samba:

```
chcon -t samba_share_t /var/www/html/prueba2
```

Ejecute el comando **ls -Z /var/www/html/prueba2** para ver los cambios:

```
-rw-r--r-- root root unconfined_u:object_r:samba_share_t:s0 /var/www/html/prueba2
```

5. Corra el comando **service httpd status** para confirmar que el proceso `httpd` no se está ejecutando:

```
$ /sbin/service httpd status
httpd está detenido
```

Si la salida difiere, ejecute **service httpd stop** como usuario root de Linux para detener el proceso `httpd`:

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

6. Para hacer que el proceso `httpd` corra no confinado, ejecute el siguiente comando como usuario root de Linux para cambiar el tipo de `/usr/sbin/httpd`, a un tipo que no transicione a un dominio confinado:

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

7. Ejecute el comando **ls -Z /usr/sbin/httpd** para confirmar que `/usr/sbin/httpd` está etiquetado con el tipo `unconfined_exec_t`:

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

8. Como usuario root de Linux, ejecute el comando **service httpd start** para iniciar el proceso `httpd`. La salida es como sigue si `httpd` inicia con éxito:

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

9. Ejecute el comando **ps -eZ | grep httpd** para ver si `httpd` está corriendo en el 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. Cambie al directorio donde su usuario Linux tenga permiso de escritura y ejecute el comando **wget http://localhost/prueba2**. A menos que hayan cambios en la configuración predeterminada, este comando debería tener éxito:

```
--2009-05-07 01:41:10-- http://localhost/prueba2
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: `prueba2.1'

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

2009-05-07 01:41:10 (0.00 B/s) - `prueba2.1' saved [0/0]
```

Aunque el proceso httpd no tiene acceso a archivos etiquetados con el tipo samba\_share\_t, httpd se ejecuta en el dominio unconfined\_t, y termina usando las reglas DAC, como tal, el comando **wget** tiene éxito. Teniendo a httpd ejecutándose en el dominio httpd\_t, el comando **wget** habría fallado.

11. El comando **restorecon** restaura el contexto SELinux predeterminado de los archivos. Como usuario root de Linux, ejecute el comando **restorecon -v /usr/sbin/httpd** para restaurar el contexto SELinux de **/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
```

Ejecute el comando **ls -Z /usr/sbin/httpd** para confirmar que **/usr/sbin/httpd** está etiquetado con el 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. Como usuario root de Linux, corra el comando **/sbin/service httpd restart** para reiniciar httpd. Después de reiniciar, ejecute **ps -eZ | grep httpd** para confirmar que httpd se está ejecutando en el dominio confinado 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. Como usuario root de Linux, corra el comando **rm -i /var/www/html/prueba2** para eliminar **prueba2**.
14. Si no necesita que corra httpd, como usuario root de Linux corra el comando **service httpd stop** para detener a httpd:

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

Los ejemplos en estas secciones muestran cómo proteger los datos desde un proceso confinado comprometido (protegido por SELinux), así como cuánto más accesible son los datos para un atacante si el proceso comprometido estaba no confinado (no protegido por SELinux).

### 4.3. Usuarios Confinados y no Confinados

Cada usuario Linux se mapea a un usuario SELinux vía la política de SELinux. Esto permite a los usuarios Linux heredar las restricciones sobre los usuarios SELinux. Este mapeo de usuarios Linux se ve ejecutando el comando **semanage login -l** como usuario root de 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

En Fedora 11, los usuarios de Linux se mapean por defecto al ingreso **\_\_default\_\_** de SELinux (el cual se mapea al usuario SELinux **unconfined\_u**). Lo siguiente define el mapeo por defecto:

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

El siguiente ejemplo muestra el agregado de un usuario de Linux nuevo y el mapeo de ese usuario al usuario SELinux **unconfined\_u**. Asume que el usuario root corre no confinado, como es por defecto en Fedora 11:

1. Como usuario root de Linux, ejecute el comando **/usr/sbin/useradd usuarioNuevo** para crear un nuevo usuario Linux con nombre **usuarioNuevo**.
2. As the Linux root user, run the **passwd newuser** command to assign a password to the Linux **newuser** user:

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

- Salga de su sesión actual e ingrese como el usuario Linux usarionuevo. Cuando ingrese, pam\_selinux mapea el usuario Linux a un usuario SELinux (en este caso, unconfined\_u), y configura el contexto SELinux resultante. El shell del usuario Linux es entonces lanzado en este contexto. Corra el comando **id -Z** para ver el contexto de un usuario Linux:

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

- Salga de la sesión del usarionuevo de Linux e ingrese con su cuenta. Si no quiere el usuario usarionuevo, ejecute el comando **/usr/sbin/userdel -r usarionuevo** como usuario root de Linux para eliminarlo, junto con su directorio de inicio.

Los usuarios Linux confinados y no confinados son sujeto a chequeo de memoria escribible y ejecutable, y también son restringidos por MCS (y MLS, si la política MLS está siendo usada). Si los usuarios Linux no confinados ejecutan una aplicación que la política de SELinux define que puede transicionar del dominio unconfined\_t a su propio dominio confinado, los usuarios Linux no confinados están sujetos a restricciones de ese dominio confinado. El beneficio de seguridad de esto es que, aunque un usuario Linux corra no confinado, la aplicación permanece confinada, y por lo tanto, la explotación de una brecha en la aplicación está limitada por la política. Nota: esto no protege al sistema del usuario. En su defecto, el usuario y el sistema están siendo protegidos de posibles daños causados por una brecha en la aplicación.

Los siguientes usuarios SELinux confinados están disponibles en Fedora 11:

Usuario	Dominio	Sistema de Ventanas X	su y sudo	Ejecute en el directorio de inicio y /tmp/	Red
guest_u	guest_t	no	no	opcional	no
xguest_u	xguest_t	si	no	opcional	sólo <b>Firefox</b>
user_u	user_t	si	no	opcional	si
staff_u	staff_t	si	sólo <b>sudo</b>	opcional	si

Tabla 4.1. Capacidades del Usuario SELinux

- Los usuarios Linux en los dominios guest\_t, xguest\_t y user\_t sólo pueden ejecutar aplicaciones con ID de usuario (setuid) si la política de SELinux lo permite (tal como **passwd**). No podrán ejecutar **su** y **/usr/bin/sudo**
- Los usuarios Linux en el dominio guest\_t no tienen acceso a la red, y sólo pueden ingresar vía una terminal (incluyendo ssh; pueden ingresar por ssh, pero no se pueden ssh conectar a otro sistema).
- El único acceso de red que tienen los usuarios en el dominio xguest\_t es con **Firefox** conectándose a páginas web.

- Los usuarios Linux en los dominios `xguest_t`, `user_t` y `staff_t` pueden ingresar vía el Sistema de Ventanas X o por una terminal.
- Por defecto, los usuarios Linux en el dominio `staff_t` no tienen permisos para ejecutar aplicaciones con `/usr/bin/sudo`. Estos permisos deben ser configurados por un administrador.

Por defecto, los usuarios de Linux en los dominios `guest_t` y `xguest_t` no pueden ejecutar aplicaciones en sus directorios de inicio o en `/tmp/`, previniéndolos de ejecutar aplicaciones (que hereden los permisos de los usuarios) en directorios a los que tienen acceso de escritura. Esto ayuda a prevenir que aplicaciones maliciosas modifiquen archivos de los usuarios.

Por defecto, los usuarios Linux en los dominios `user_t` y `staff_t` pueden ejecutar aplicaciones en sus directorios de inicio y en `/tmp/`. Vaya a [Sección 6.6, “Booleanos para que los Usuarios Ejecuten Aplicaciones”](#) para información sobre permitir y evitar que los usuarios ejecuten aplicaciones en sus directorios de inicio y en `/tmp/`.

---

---

# Trabajando con SELinux

Las siguientes secciones dan un breve repaso de los paquetes de SELinux principales en Fedora 11; instalación y actualización de paquetes; qué archivos de registro se usan; el archivo de configuración principal de SELinux; habilitación y deshabilitación de SELinux; modos de SELinux; configuración de Booleanos; cambios temporales y permanentes de etiquetas de archivos y directorios; superposición de las etiquetas de sistemas de archivos con el comando **mount**; montaje de sistemas de archivos NFS; y cómo preservar contextos de SELinux cuando se copia y compacta archivos y directorios.

## 5.1. Paquetes de SELinux

En Fedora 11, los paquetes SELinux se instalan por defecto, a menos que hayan sido manualmente excluidos durante la instalación. Por defecto, la política destinada de SELinux se usa y SELinux corre en modo Obediente. La siguiente es una descripción breve de los paquetes SELinux principales:

*polycoreutils*: provee utilitarios, tales como **semanage**, **restorecon**, **audit2allow**, **semodule**, **load\_policy** y **setsebool**, para la operación y administración de SELinux.

*polycoreutils-gui*: provee **system-config-selinux**, una herramienta gráfica para la administración de SELinux.

*selinux-policy*: provee una Política de Referencia de SELinux. La Política de Referencia de SELinux en una política de SELinux completa, y se usa como base para otras políticas, tales como la política destinada de SELinux. Vaya a la página [Política de Referencia de SELinux](#)<sup>1</sup> de Tresys Technology para más información. El paquete *selinux-policy-devel* provee herramientas de desarrollo, tales como `/usr/share/selinux/devel/policygentool` y `/usr/share/selinux/devel/policyhelp`, así como archivos de política ejemplos. Este paquete fue mezclado con el paquete *selinux-policy*.

*selinux-policy-policy*: provee las políticas de SELinux. Para la política destinada, instale *selinux-policy-targeted*. Para MLS, instale *selinux-policy-mls*. En Fedora 8, la política estricta fue mezclada con la política destinada, permitiendo a los usuarios confinados y no confinados coexistir en el mismo sistema.

*setroubleshoot-server*: traduce mensajes de negaciones, producidos cuando el acceso es denegado por SELinux, en descripciones detalladas que se ven con **sealert** (que se provee en este paquete).

*setools*, *setools-gui* y *setools-console*: estos paquetes proveen la [Distribución de SEHerramientas de Tresys Technology](#)<sup>2</sup>, un número de herramientas y bibliotecas para analizar y consultar la política, monitorear e informes de auditoría, y administración de contexto de archivos<sup>3</sup>. El paquete *setools* es un metapaquete para las SEHerramientas. El paquete *setools-gui* provee las herramientas **apol**, **seaudit** y **sediffx**. El paquete *setools-console* provee las herramientas de línea de comando **seaudit-report**, **sechecker**, **sediff**, **seinfo**, **sesearch**, **findcon**, **replcon** y **indexcon**. Vaya a la página de las [SEHerramientas de Tresys Technology](#)<sup>4</sup> para información sobre estas herramientas.

*libselinux-utils*: provee las herramientas **avcstat**, **getenforce**, **getsebool**, **matchpathcon**, **selinuxconlist**, **selinuxdefcon**, **selinuxenabled**, **setenforce**, **togglesebool**.

---

<sup>1</sup> <http://oss.tresys.com/projects/refpolicy>

<sup>2</sup> <http://oss.tresys.com/projects/setools>

<sup>3</sup> Brindle, Joshua. "Re: blurb for fedora setools packages" Email para Murray McAllister. 1 Noviembre 2008. Cualquier edición o cambio en esta versión fue hecha por Murray McAllister.

<sup>4</sup> <http://oss.tresys.com/projects/setools>

*mcstrans*: traduce niveles, tales como `s0-s0:c0.c1023`, a una forma legible como `SystemLow-SystemHigh`. Este paquete no se instala por defecto.

Para instalar paquetes en Fedora 11, como usuario `root` de Linux ejecute el comando **yum install nombre-de-paquete**. Por ejemplo, para instalar el paquete *mcstrans*, ejecute el comando **yum install mcstrans**. Para actualizar todos los paquetes instalados en Fedora 11, ejecute el comando **yum update**.

Vaya a [Administración de Software con yum](#)<sup>5</sup> para más información sobre el uso de **yum** para administrar paquetes.



### Nota

En versiones anteriores de Fedora, el paquete *selinux-policy-devel* es necesario cuando se crea un módulo de política local con **audit2allow -M**.

## 5.2. Qué Archivo Log se usa

En Fedora 11, los paquetes *setroubleshoot-server* y *audit* se instalan si los paquetes no se eliminan de la selección de paquetes predeterminada. Estos paquetes incluyen a los demonios *setroubleshootd* y *auditd* respectivamente. Estos demonios se ejecutan por defecto.

Los mensajes de negación de SELinux, tales como el siguiente, se escriben por defecto en **/var/log/audit/audit.log**:

```
type=AVC msg=audit(1223024155.684:49): avc: denied { getattr }
for pid=2000 comm="httpd" path="/var/www/html/archivo1"
dev=dm-0 ino=399185 scontext=unconfined_u:system_r:httpd_t:s0
tcontext=system_u:object_r:samba_share_t:s0 tclass=file
```

También, si *setroubleshootd* está ejecutándose, lo que es por defecto, los mensajes de negación desde **/var/log/audit/audit.log** se traducen a una forma más fácil de leer en **/var/log/messages**:

```
May 7 18:55:56 localhost setroubleshoot: SELinux is preventing
httpd (httpd_t) "getattr" to /var/www/html/archivo1 (samba_share_t).
For complete SELinux messages. run sealert -l de7e30d6-5488-466d-
a606-92c9f40d316d
```

Los mensajes de negación se envían a una ubicación distinta, dependiendo de cuáles demonios se están ejecutando:

Daemon	Log Location
<code>auditd</code> on	<b>/var/log/audit/audit.log</b>
<code>auditd</code> off; <code>rsyslogd</code> on	<b>/var/log/messages</b>
<code>setroubleshootd</code> , <code>rsyslogd</code> , and <code>auditd</code> on	<b>/var/log/audit/audit.log</b> . Easier-to-read denial messages also sent to <b>/var/log/messages</b>

<sup>5</sup> <http://docs.fedoraproject.org/yum/en/>

<sup>6</sup> Administración de Software con yum, escrito por Stuart Ellis, editado por Paul W. Fields, Rodrigo Menezes y Hugo Cisneiros.

### Iniciando Demonios Automáticamente

Para configurar los demonios `auditd`, `rsyslogd`, y `setroubleshootd` para que inicien automáticamente al arrancar, corra los siguientes comandos como usuario `root` de Linux:

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

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

```
/sbin/chkconfig --levels 345 setroubleshoot on
```

Use el comando `service nombre-de-servicio status` para chequear si estos servicios se están ejecutando, por ejemplo:

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

Si los servicios de arriba no se están ejecutando (*nombre-de-servicio* está detenido), use el comando `service nombre-de-servicio start` como usuario `root` de Linux para iniciarlos. Por ejemplo:

```
# /sbin/service setroubleshoot start
Starting setroubleshootd: [ OK ]
```

## 5.3. Archivo de Configuración Principal

El archivo `/etc/selinux/config` es el archivo de configuración principal de SELinux. Controla el modo de SELinux y la política de SELinux a usar:

```
# 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`

La opción `SELINUX` pone el modo en el que corre SELinux. SELinux tiene tres modos: obediente, permisivo y deshabilitado. Cuando se usa modo obediente, la política de SELinux es aplicada y SELinux niega el acceso basándose en las reglas de políticas de SELinux. Los mensajes de negación se guardan. Cuando se usa modo permisivo, la política de SELinux no es obediente. Los mensajes son guardados. SELinux no niega el acceso, pero se guardan las negaciones de acciones que hubieran sido negadas si SELinux estaba en modo obediente. Cuando se usa el modo deshabilitado, SELinux está deshabilitado (el módulo de SELinux no se registra con el kernel de Linux), y sólo se usan las reglas DAC.

SELINUXTYPE=targeted

La opción SELINUXTYPE pone la política SELinux a usar. La política Destinada es la predeterminada. Sólo cambie esta opción si quiere usar la política MLS. Para usar la política MLS, instale el paquete *selinux-policy-mls*; configure SELINUXTYPE=mls en **/etc/selinux/config**; y reinicie su sistema.



### Importante

Cuando los sistemas corren con SELinux en modo permisivo o deshabilitado, los usuarios tiene permiso para etiquetar los archivos incorrectamente. También, los archivos creados con SELinux deshabilitado no son etiquetados. Esto causa problemas cuando se cambia a modo obediente. Para prevenir el etiquetado incorrecto o la falta de etiquetado, los sistemas de archivos son automáticamente reetiquetados cuando se cambia desde el modo deshabilitado al modo permisivo u obediente.

## 5.4. Habilitando y Deshabilitando SELinux

Use los comandos **/usr/sbin/getenforce** o **/usr/sbin/sectatus** para chequear el estado de SELinux. El comando **getenforce** devuelve Obediente, Permisivo, o Deshabilitado. El comando **getenforce** devuelve Obediente cuando SELinux está habilitado (las reglas de la política de SELinux son aplicadas):

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

El comando **getenforce** devuelve Permissive cuando SELinux está activado, pero las reglas de políticas de SELinux no están en obligatorio, y sólo se usan las reglas DAC. El comando **getenforce** devuelve Disabled si SELinux está deshabilitado.

El comando **sectatus** devuelve el estado de SELinux y la política de SELinux que se está usando:

```
$ /usr/sbin/sectatus
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 is returned when SELinux is enabled. Current mode: enforcing is returned when SELinux is running in enforcing mode. Policy from config file: targeted is returned when the SELinux targeted policy is used.

### 5.4.1. Habilitando SELinux

En sistemas con SELinux deshabilitado, la opción SELINUX=disabled se configura en **/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
```

También, el comando **getenforce** devuelve Disabled:

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

Para habilitar SELinux:

1. Use los comandos **rpm -qa | grep selinux**, **rpm -q policycoreutils** y **rpm -qa | grep setroubleshoot** para confirmar que los paquetes de SELinux están instalados. esta guía asume que los siguientes paquetes están instalados: *selinux-policy-targeted*, *selinux-policy*, *libselinux*, *libselinux-python*, *libselinux-utils*, *policycoreutils*, *setroubleshoot*, *setroubleshoot-server*, *setroubleshoot-plugins*. Si estos paquetes no están instalados, como usuario root de Linux, debe instalarlos con el comando **yum install nombre-de-paquete**. Los siguientes paquetes son opcionales: *policycoreutils-gui*, *setroubleshoot*, *selinux-policy-devel* y *mcstrans*.

Después de instalar el paquete *setroubleshoot-server*, use el comando **/sbin/chkconfig --list setroubleshoot** para confirmar que *setroubleshootd* se inicia cuando el sistema corre en el nivel de ejecución <sup>7</sup> 3, 4, y 5:

```
$ /sbin/chkconfig --list setroubleshoot
setroubleshoot 0:off 1:off 2:off 3:on 4:on 5:on 6:off
```

Si la salida difiere, como usuario root de Linux ejecute el comando **/sbin/chkconfig --levels 345 setroubleshoot on**. Esto hace que *setroubleshootd* se inicie automáticamente cuando el sistema esté en los niveles de ejecución 3, 4 y 5.

2. Antes de activar SELinux, cada archivo en el sistema de archivo debe ser etiquetado con un contexto de SELinux. Antes que esto ocurra, los dominios confinados pueden tener el acceso denegado, impidiendo de que su sistema se inicie correctamente. Para prevenir esto, configure **SELINUX=permissive** en **/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. Como usuario root de Linux, corra el comando **reboot** para reiniciar el sistema. Durante el siguiente arranque, los sistemas de archivo son etiquetados. El proceso etiqueta todos los archivos con un contexto de 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.
****
```

Cada carácter \* en la línea de abajo representa 1000 archivos que han sido etiquetados. En el ejemplo de arriba, cuatro caracteres \* representan 4000 archivos etiquetados. El tiempo que toma reetiquetar todos los archivos depende del número de archivos del sistema, y la velocidad de los discos rígidos. En sistemas modernos, este proceso puede tomar 10 minutos.

4. En modo permisivo, la política de SELinux no es aplicada, pero las negaciones se guardan para las acciones que hubieran sido negadas si estaba corriendo en modo obediente. Antes de cambiar a modo obediente, como usuario root de Linux, ejecute el comando **grep "SELinux está negando" /var/log/messages** como usuario root de Linux para confirmar que SELinux no negó acciones durante la última reiniciada. Si SELinux no negó acciones durante el último arranque, este comando no devuelve nada. Vaya a [Capítulo 7, Solución a Problemas](#) para información para resolver problemas si SELinux negó el acceso durante el arranque.
5. Si no hay mensajes de negación en **/var/log/messages**, configure SELINUX=enforcing en **/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. Reinicie su sistema. Después de reiniciar, confirme que **getenforce** devuelve Enforcing:

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

7. Como usuario root de Linux, corra el comando **/usr/sbin/semanage login -l** para ver el mapeo entre usuarios de SELinux y de Linux. La salida debe ser como la siguiente:

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

Si éste no es el caso, ejecute los siguientes comandos como usuario root de Linux para corregir los mapeos de usuario. Es seguro ignorar los mensajes El usuario SELinux *nombre-de-usuario* ya está definido si es que aparecen, donde *nombre-de-usuario* puede ser *unconfined\_u*, *guest\_u*, o *xguest\_u*:

1. 

```
/usr/sbin/semange usuario -a -S targeted -P usuario -R "unconfined_r system_r" -r s0-s0:c0.c1023 unconfined_u
```
2. 

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

```
/usr/sbin/semange login -m -S targeted -s "unconfined_u" -r s0-s0:c0.c1023 root
```
4. 

```
/usr/sbin/semange usuario -a -S targeted -P usuario -R guest_r guest_u
```
5. 

```
/usr/sbin/semange usuario -a -S targeted -P usuario -R xguest_r xguest_u
```



### Importante

Cuando los sistemas corren con SELinux en modo permisivo o deshabilitado, los usuarios tiene permiso para etiquetar los archivos incorrectamente. También, los archivos creados con SELinux deshabilitado no son etiquetados. Esto causa problemas cuando se cambia a modo obediente. Para prevenir el etiquetado incorrecto o la falta de etiquetado, los sistemas de archivos son automáticamente reetiquetados cuando se cambie desde el modo deshabilitado al modo permisivo u obediente.

## 5.4.2. Deshabilitando SELinux

Para deshabilitar SELinux, configure `SELINUX=disabled` en `/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
```

Reinicie su sistema. Después de reiniciar, confirme que **getenforce** devuelve Disabled:

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

Disabled

## 5.5. Modos de SELinux

SELinux tiene tres modos:

- Obligatorio: la política de SELinux es obligatoria. SELinux niega el acceso basado en las reglas de políticas de SELinux.
- Permisivo: la política de SELinux no es obligatoria. SELinux no niega el acceso, pero se guardan las negaciones para acciones que hubieran sido negadas si el modo obligatorio estaba activado.
- Deshabilitado: SELinux está deshabilitado. Sólo se usan las reglas DAC.

Use el comando `/usr/sbin/setenforce` para cambiar entre los modos obediente y permisivo. Los cambios hechos con `/usr/sbin/setenforce` no sobreviven a una reiniciada. Para cambiar a modo obediente, como usuario root de Linux, ejecute el comando `/usr/sbin/setenforce 1`. Para cambiar a modo permisivo, ejecute el comando `/usr/sbin/setenforce 0`. Use el comando `/usr/sbin/getenforce` para ver el modo de SELinux actual.

Los cambios de modo persistentes se cubren en [Sección 5.4, “Habilitando y Deshabilitando SELinux”](#).

## 5.6. Booleanos

Los booleanos permiten cambiar partes de la política de SELinux en tiempo de ejecución, sin ningún conocimiento sobre la escritura de políticas de SELinux. Esto permite cambios, como permitir el acceso de servicios a sistemas de archivo NFS, sin recargar o recompilar la política de SELinux.

### 5.6.1. Listando los Booleanos

Para una lista de los Booleanos, una explicación de lo que son y de si están activos o inactivos, ejecute el comando `semanage boolean -l` como usuario root de Linux. El siguiente ejemplo no lista todos los Booleanos:

```
# /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
xgquest_connect_network -> on    Allow xgquest to configure Network
  Manager
```

La columna SELinux boolean lista los nombres de Booleanos. La columna Description lista si el booleano está activo (on) o inactivo (off) y lo que hacen.

En el siguiente ejemplo, el Booleano ftp\_home\_dir está apagado, impidiendo al demonio FTP (vsftpd) la lectura y escritura de archivos en los directorios de inicio de los usuarios:

```
ftp_home_dir             -> off   Permite a ftp leer y escribir al
  directorio de inicio de los usuarios
```

El comando **getsebool -a** lista los Booleanos, ya sea que estén activos o inactivos, pero no da una descripción de cada uno. El siguiente ejemplo no lista todos los booleanos:

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

Ejecute el comando **getsebool nombre-de-booleano** para listar solamente el estado del booleano *nombre-de-booleano*:

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

Una lista separada por espacio para listar los Booleanos múltiples:

```
$ 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. Configurando los Booleanos

El comando **setsebool nombre-de-booleano x** activa o desactiva Booleanos, donde *nombre-de-booleano* es un nombre de Booleano, y *x* es **on** para activar, u **off** para desactivar.

El siguiente ejemplo muestra la configuración de Booleano `httpd_can_network_connect_db`:

1. Por defecto, el booleano `httpd_can_network_connect_db` está apagado, impidiendo a los scripts y módulos del Servidor HTTP Apache conectarse a servidores de bases de datos:

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

2. Para permitir temporalmente a los scripts y módulos del Servidor HTTP Apache conectarse a servidores de bases de datos, ejecute el comando **setsebool httpd\_can\_network\_connect\_db on** como usuario `root` de Linux.
3. Use el comando **getsebool httpd\_can\_network\_connect\_db** para verificar que el Booleano está activado:

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

Esto permite a los scripts y módulos del Servidor HTTP Apache conectarse a servidores de bases de datos.

4. Este cambio no es persistente entre reinicios. Para hacer los cambios persistentes, corra el comando **setsebool -P boolean-name on** como usuario `root` de Linux:

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

5. Para revertir temporalmente el comportamiento por defecto, como usuario root de Linux, corra el comando **setsebool httpd\_can\_network\_connect\_db off**. Para que los cambios sean persistentes entre reiniciadas, ejecute el comando **setsebool -P httpd\_can\_network\_connect\_db off**.

### 5.6.3. Booleanos para NFS y CIFS

Por defecto, los montajes NFS en el lado del cliente se etiquetan con el contexto predeterminado definido por la política para sistemas de archivos NFS. En políticas comunes, este contexto predeterminado usa el tipo `nfs_t`. También, por defecto, los compartidos de Samba en el lado del cliente se etiquetan con el contexto predeterminado definido por la política. En políticas comunes, este contexto predeterminado usa el tipo `cifs_t`.

Dependiendo en la configuración de la política, los servicios pueden tener bloqueado la lectura a archivos con la etiqueta de los tipos `nfs_t` o `cifs_t`. Esto puede prevenir que los sistemas de archivo etiquetados con estas etiquetas se monten y sean leídos o exportados por otros servicios. Hay Booleanos que se pueden poner en 1 o 0 para controlar qué servicios pueden acceder los tipos `nfs_t` y `cifs_t`.

Los comandos **setsebool** y **semanage** se deben ejecutar como usuario root de Linux. El comando **setsebool -P** hace persistentes a los cambios. No use la opción `-P` si no quiere que los cambios persistan entre reiniciadas:

#### Servidor HTTP Apache

Para permitir el acceso a sistemas de archivo NFS (archivos etiquetados con el tipo `nfs_t`):

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

Para permitir el acceso a sistemas de archivos SAMBA (archivos etiquetados con el tipo `cifs_t`):

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

#### Samba

Para exportar sistemas de archivo NFS:

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

#### FTP (vsftpd)

Para permitir el acceso a sistemas de archivo NFS:

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

Para permitir el acceso a sistemas de archivo Samba:

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

#### Otros Servicios

Para una lista de los Booleanos relacionados con NFS para otros servicios:

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

Para una lista de los Booleanos relacionados con SAMBA para otros servicios:

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



### Nota

Estos Booleanos existen en la política de SELinux tal cual fueron puestos en Fedora 11. No pueden existir en la política puesta en otras versiones de Fedora o de otros sistemas operativos.

## 5.7. Contextos de SELinux - Etiquetado de Archivos

En sistemas que corren SELinux, todos los procesos y archivos son etiquetados con una etiqueta que contiene información de seguridad relevante. Esta información se llama contexto de SELinux. Para archivos, esto se ve usando el comando `ls -Z`:

```
$ ls -Z archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 archivo1
```

En este ejemplo, SELinux provee un usuario (`unconfined_u`), un rol (`object_r`), un tipo (`user_home_t`) y un nivel (`s0`). Esta información se usa para tomar decisiones sobre el control de acceso. En sistemas DAC, el control de acceso se basa en los IDs de usuario y grupo de Linux. Las reglas de la política de SELinux se chequean después de las reglas DAC. Las reglas de la política de SELinux no se usan si las reglas DAC niegan el acceso antes.

Hay muchos comandos para la administración del contexto de archivos de SELinux, como por ejemplo `chcon`, `semanage fcontext`, y `restorecon`.

### 5.7.1. Cambios Temporales: `chcon`

El comando `chcon` cambia el contexto SELinux de los archivos. Estos cambios no sobreviven un reetiquetado del sistema de archivo, o el comando `/sbin/restorecon`. La política de SELinux controla si los usuarios pueden modificar el contexto SELinux de algún archivo. Cuando se usa `chcon`, los usuarios proveen toda o parte del contexto SELinux a cambiar. Un tipo de archivo incorrecto es una causa común de negación de acceso de SELinux.

#### Referencia Rápida

- Ejecute el comando `chcon -t tipo nombre-de-archivo` para cambiar el tipo de archivo, donde *tipo* es el tipo, por ejemplo `httpd_sys_content_t`, y *nombre-de-archivo* es un nombre de archivo o de directorio.
- Ejecute el comando `chcon -R -t tipo nombre-de-directorio` para cambiar el tipo de un directorio y su contenido, donde *tipo* es el tipo, por ejemplo `httpd_sys_content_t`, y *nombre-de-directorio* es un nombre de directorio.

#### Cambiando el Tipo de un Archivo o de un Directorio

El siguiente ejemplo muestra el cambio de tipo solamente en el contexto de SELinux:

1. Ejecute el comando **cd** sin argumentos para cambiar a su directorio de inicio.
2. Ejecute el comando **touch archivo1** para crear un archivo nuevo. Use el comando **ls -Z archivo1** para ver el contexto de SELinux del **archivo1**:

```
$ ls -Z archivo1
-rw-rw-r--  usuario1 grupo1 unconfined_u:object_r:user_home_t:s0
archivo1
```

En este ejemplo, el contexto SELinux del **archivo1** incluye el usuario **unconfined\_u**, el rol **object\_r**, el tipo **user\_home\_t**, y el nivel **s0**. Para una descripción de cada parte del contexto SELinux, vaya a [Capítulo 3, Contextos de SELinux](#).

3. Ejecute el comando **chcon -t samba\_share\_t archivo1** para cambiar el tipo a **samba\_share\_t**. La opción **-t** sólo cambia el tipo. Vea el cambio con **ls -Z archivo1**:

```
$ ls -Z archivo1
-rw-rw-r--  usuario1 grupo1 unconfined_u:object_r:samba_share_t:s0
archivo1
```

4. Use el comando **/sbin/restorecon -v archivo1** para restaurar el contexto de SELinux del **archivo1**. Use la opción **-v** para ver qué cambia:

```
$ /sbin/restorecon -v archivo1
restorecon reset archivo1 context
unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:user_home_t:s0
```

En este ejemplo, el tipo previo **samba\_share\_t**, se restaura al tipo correcto **user\_home\_t**. Cuando se usa la política destinada (la política SELinux predeterminada en Fedora 11), el comando **/sbin/restorecon** lee los archivos en el directorio **/etc/selinux/targeted/contexts/files/** para ver qué contexto de SELinux deben tener los archivos.

El ejemplo en esta sección funciona igual para directorios, por ejemplo, si **archivo1** fuera un directorio.

### Cambio de un Directorio y sus Tipos de Contenidos

El siguiente ejemplo muestra la creación de un directorio nuevo y el cambio del tipo de archivo del mismo (junto con su contenido) a un tipo usado por el Servidor HTTP Apache. La configuración en este ejemplo se usa si quiere que el Servidor HTTP Apache use una raíz de documento distinta (en vez de **/var/www/html/**):

1. Como usuario **root** de Linux, ejecute el comando **mkdir /web** para crear un directorio nuevo, y luego el comando **touch /web/archivo{1,2,3}** para crear 3 archivos vacíos (**archivo1**, **archivo2** y **archivo3**). El directorio **/web/** y los archivos en él son etiquetados con el 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 archivo1
-rw-r--r-- root root unconfined_u:object_r:default_t:s0 archivo2
-rw-r--r-- root root unconfined_u:object_r:default_t:s0 archivo3
```

2. Como usuario root de Linux, corra el comando **chcon -R -t httpd\_sys\_content\_t /web/** para cambiar el tipo del directorio **/web/** (y su contenido) a **httpd\_sys\_content\_t**:

```
# 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
  archivo1
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
  archivo2
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
  archivo3
```

3. Como usuario root de Linux, corra el comando **/sbin/restorecon -R -v /web/** para restaurar los contextos de SELinux predeterminados:

```
# /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/archivo2 context
  unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
restorecon reset /web/archivo3 context
  unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
restorecon reset /web/archivo1 context
  unconfined_u:object_r:httpd_sys_content_t:s0-
>system_u:object_r:default_t:s0
```

Vaya a la página del manual de **chcon**(1) para más información sobre **chcon**.



### Nota

La Obligación de Tipo es el control de permisos principal usado en la política destinada de SELinux. Para la mayor parte, los usuarios y roles de SELinux se pueden ignorar.

## 5.7.2. Cambios Persistentes: semanage fcontext

El comando **/usr/sbin/semanage fcontext** cambia el contexto SELinux de los archivos. Cuando se usa la política destinada, los cambios hechos con este comando se agregan al archivo **/etc/selinux/targeted/contexts/files/file\_contexts** si los cambios son para archivos que están en **file\_contexts**, se agregan a **file\_contexts.local** para archivos nuevos y directorios, como sería al crear un directorio **/web/** nuevo. **setfiles**, que se usa cuando el

sistema de archivo es reetiquetado, y **/sbin/restorecon**, que restaura los contextos de SELinux predeterminados, leen estos archivos. Lo que significa que los cambios hechos por **/usr/sbin/semange fcontext** son persistentes, aún si el sistema de archivo es reetiquetado. La política de SELinux controla si los usuarios pueden modificar el contexto de SELinux para cualquier archivo dado.

### Referencia Rápida

Para hacer que los cambios de contexto de SELinux sobrevivan un reetiquetado del sistema de archivo:

1. Ejecute el comando **/usr/sbin/semange fcontext -a opciones nombre-de-archivo|nombre-de-directorio**, recuerde usar la dirección completa del archivo o del directorio.
2. Ejecute el comando **/sbin/restorecon -v nombre-de-archivo|nombre-de-directorio** para aplicar los cambios de contexto.

### Cambiando un Tipo de archivo

El siguiente ejemplo muestra el cambio de tipo de un archivo, sin tocar otros atributos del contexto de SELinux:

1. Como usuario root de Linux, ejecute el comando **touch /etc/archivo1** para crear un archivo nuevo. Por defecto, los archivos recién creados en el directorio **/etc/** se etiquetan con el tipo **etc\_t**:

```
# ls -Z /etc/archivo1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/archivo1
```

2. Como usuario root de Linux, ejecute el comando **/usr/sbin/semange fcontext -a -t samba\_share\_t /etc/archivo1** para cambiar el tipo del **archivo1** a **samba\_share\_t**. La opción **-a** agrega un registro nuevo, y la opción **-t** define un tipo (**samba\_share\_t**). Nota: al ejecutar este comando no se cambia directamente el tipo - el **archivo1** todavía es del tipo **etc\_t**:

```
# /usr/sbin/semange fcontext -a -t samba_share_t /etc/archivo1
# ls -Z /etc/archivo1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/archivo1
```

El comando **/usr/sbin/semange fcontext -a -t samba\_share\_t /etc/archivo1** agrega la siguiente entrada a **/etc/selinux/targeted/contexts/files/file\_contexts.local**:

```
/etc/archivo1 unconfined_u:object_r:samba_share_t:s0
```

3. Como usuario root de Linux, ejecute el comando **/sbin/restorecon -v /etc/archivo1** para cambiar el tipo. Dado que el comando **semange** agregó una entrada a **file\_contexts.local** para **/etc/archivo1**, el comando **/sbin/restorecon** cambia el tipo a **samba\_share\_t**:

```
# /sbin/restorecon -v /etc/archivo1
restorecon reset /etc/archivo1 context unconfined_u:object_r:etc_t:s0-
>system_u:object_r:samba_share_t:s0
```

4. Como usuario root de Linux, ejecute el comando **rm -i /etc/archivo1** para borrar el **archivo1**.
5. Como usuario root de Linux, ejecute el comando **/usr/sbin/semanage fcontext -d /etc/archivo1** para eliminar el contexto agregado para **/etc/archivo1**. Cuando el contexto se elimina, ejecutando **restorecon** cambia el tipo a **etc\_t**, en vez de **samba\_share\_t**.

### Cambiando un Tipo de Directorio

El siguiente ejemplo muestra la creación de un directorio nuevo y el cambio del tipo de archivo del directorio a un tipo usado por el Servidor HTTP Apache:

1. Como usuario root de Linux, ejecute el comando **mkdir /web** para crear un directorio nuevo. Este directorio se etiqueta con el tipo **default\_t**:

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

La opción **-d** de **ls** hace que **ls** liste la información de un directorio, en vez de su contenido, y la opción **-Z** hace que **ls** muestre el contexto de SELinux (en este ejemplo, **unconfined\_u:object\_r:default\_t:s0**).

2. Como usuario root de Linux, ejecute el comando **/usr/sbin/semanage fcontext -a -t httpd\_sys\_content\_t /web** para cambiar el tipo de **/web/** a **httpd\_sys\_content\_t**. La opción **-a** agrega un nuevo registro, y la opción **-t** define un tipo (**httpd\_sys\_content\_t**). Nota: la ejecución de este comando no cambia el tipo directamente - **/web/** todavía tiene la etiqueta de 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
```

El comando **/usr/sbin/semanage fcontext -a -t httpd\_sys\_content\_t /web** agrega la siguiente entrada a **/etc/selinux/targeted/contexts/files/file\_contexts.local**:

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

3. Como usuario root de Linux, ejecute el comando **/sbin/restorecon -v /web** para cambiar el tipo. Como el comando **semanage** agregó una entrada a **file\_contexts.local** para **/web**, el comando **/sbin/restorecon** cambia el tipo a **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
```

Por defecto, los archivos y directorios recién creados heredan el tipo SELinux de sus carpetas padres. Cuando se usó este ejemplo, y antes de eliminar el contexto SELinux agregado para **/web/**, los archivos y directorios creados en el directorio **/web/** fueron etiquetados con el tipo `httpd_sys_content_t`.

4. Como usuario root de Linux, ejecute el comando `/usr/sbin/semanage fcontext -d /web` para borrar el contexto agregado para **/web/**.
5. Como usuario root de Linux, ejecute el comando `/sbin/restorecon -v /web` para restaurar el contexto predeterminado de SELinux.

### Cambio de un Directorio y sus Tipos de Contenidos

El siguiente ejemplo muestra la creación de un directorio nuevo y el cambio del tipo de archivo del mismo (junto con su contenido) a un tipo usado por el Servidor HTTP Apache. La configuración en este ejemplo se usa si quiere que el Servidor HTTP Apache use una raíz de documento distinta (en vez de `/var/www/html/`):

1. Como usuario root de Linux, ejecute el comando `mkdir /web` para crear un directorio nuevo, y luego el comando `touch /web/archivo{1,2,3}` para crear 3 archivos vacíos (**archivo1**, **archivo2** y **archivo3**). El directorio **/web/** y los archivos en él son etiquetados con el 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 archivo1
-rw-r--r-- root root unconfined_u:object_r:default_t:s0 archivo2
-rw-r--r-- root root unconfined_u:object_r:default_t:s0 archivo3
```

2. Como usuario root de Linux, ejecute el comando `/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"` para cambiar el tipo del directorio **/web/** junto con los archivos dentro de él, a `httpd_sys_content_t`. La opción `-a` agrega un registro nuevo, y la opción `-t` define un tipo (`httpd_sys_content_t`). La expresión regular `"/web(/.*)?"` hace que el comando **semanage** aplique los cambios al directorio **/web/**, así como a los archivos dentro de él. Nota: la ejecución de este comando no cambia el tipo directamente - **/web/** y los archivos dentro de él todavía tienen la etiqueta del 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 archivo1
-rw-r--r-- root root unconfined_u:object_r:default_t:s0 archivo2
-rw-r--r-- root root unconfined_u:object_r:default_t:s0 archivo3
```

El comando `/usr/sbin/semanage fcontext -a -t httpd_sys_content_t "/web(/.*)?"` agrega una entrada a `/etc/selinux/targeted/contexts/files/file_contexts.local`:

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

- Como usuario root de Linux, ejecute el comando **/sbin/restorecon -R -v /web** para cambiar el tipo del directorio **/web/**, junto con los archivos dentro de él. La opción **-R** significa recursivo, es decir, todos los archivos y directorios dentro del directorio **/web/** se etiquetarán con el tipo **httpd\_sys\_content\_t**. Dado que el comando **semanage** agregó una entrada en **file\_contexts.local** para **/web(/. \*)?**, el comando **/sbin/restorecon** los tipos a **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/archivo2 context
  unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/archivo3 context
  unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
restorecon reset /web/archivo1 context
  unconfined_u:object_r:default_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

Por defecto, los archivos y directorios recién creados heredan el tipo SELinux de sus padres. En este ejemplo, los archivos y directorios creados en el directorio **/web/** se etiquetarán con el tipo **httpd\_sys\_content\_t**.

- Como usuario root de Linux, ejecute el comando **/usr/sbin/semanage fcontext -d "/web(/. \*)?"** para eliminar el contexto agregado para **"/web(/. \*)?"**.
- Como usuario root de Linux, ejecute el comando **/sbin/restorecon -R -v /web** para restaurar el contexto predeterminado de SELinux.

### Borrado de un Contexto agregado

El siguiente ejemplo muestra el agregado y su eliminación del contexto de SELinux:

- Como usuario root de Linux, ejecute el comando **/usr/sbin/semanage fcontext -a -t httpd\_sys\_content\_t /prueba**. El directorio **/prueba/** no tiene que existir. Este comando agrega el siguiente contexto a **/etc/selinux/targeted/contexts/files/file\_contexts.local**:

```
/prueba    system_u:object_r:httpd_sys_content_t:s0
```

- Para eliminar el contexto, como usuario root de Linux, ejecute el comando **/usr/sbin/semanage fcontext -d nombre-de-archivo|nombre-de-directorio**, donde **nombre-de-archivo|nombre-de-directorio** es la primera parte en **file\_contexts.local**. El siguiente es un ejemplo de un contexto en **file\_contexts.local**:

```
/prueba    system_u:object_r:httpd_sys_content_t:s0
```

Siendo la primera parte **/prueba**. Para evitar que el directorio **/prueba/** se etiquete con **httpd\_sys\_content\_t** después de ejecutar **/sbin/restorecon**, o después de un

reetiquetado del sistema, ejecute el siguiente comando como usuario root de Linux para eliminar el contexto de `file_contexts.local`:

```
/usr/sbin/semanage fcontext -d /prueba
```

Si el contexto es parte de una expresión regular, por ejemplo `/web(/. *)?`, use las comillas para encerrar la expresión regular:

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

Vaya a la página de manual de `semanage(8)` para información adicional sobre `/usr/sbin/semanage`.



### Importante

Cuando se cambia el contexto de SELinux con `/usr/sbin/semanage fcontext -a`, use la dirección completa del archivo o directorio para evitar etiquetar mal los archivos después de un reetiquetado del sistema de archivo, o después ejecutar el comando `/sbin/restorecon`.

## 5.8. Los tipos `file_t` y `default_t`

En sistemas de archivo que soportan atributos extendidos, cuando se accede a un archivo en disco al que le falta el contexto de SELinux, se trata como si tuviera un contexto predeterminado tal como fue definido en la política de SELinux. En políticas comunes, este contexto por defecto usa el tipo `file_t`. Este debería ser el único uso de este tipo, por lo que los archivos sin un contexto en el disco se podrían distinguir en la política, y generalmente mantenerse inaccesibles a dominios confinados. El tipo `file_t` no debería existir en sistemas de archivo correctamente etiquetados, porque todos los archivos dado que todos los archivos en un sistema corriendo SELinux deberían tener un contexto de SELinux, y el tipo `file_t` no debería usarse en la configuración de contexto de archivos<sup>8</sup>.

El tipo `default_t` se usa en archivos que no coinciden con ningún otro patrón en la configuración de contexto de archivo, por lo que tales archivos se pueden distinguir de aquellos que no tienen un contexto en el disco, y generalmente mantenerse inaccesibles a los dominios confinados. Si crea un directorio de alto nivel, tal como `/midir/`, este directorio puede ser etiquetado con el tipo `default_t`. Si los servicios necesitan acceder a tal directorio, actualice la configuración de contextos de archivo para esta ubicación. Vaya a [Sección 5.7.2, "Cambios Persistentes: `semanage fcontext`"](#) para más detalles sobre agregado de un contexto para la configuración de contexto de archivos.

## 5.9. Montaje de Sistemas de Archivos

Por defecto, cuando un sistema de archivo que soporta atributos extendidos se monta, el contexto de seguridad para cada archivo se obtiene de atributo extendido `security.selinux` del archivo. A los archivos en sistemas de archivo que no dan soporte a atributos extendidos se les asigna un único contexto de seguridad predeterminado desde la configuración de la política, basada en el tipo de sistema de archivo.

Use el comando `mount -o context` para superponer los atributos extendidos actuales, o para especificar uno distinto y por defecto para sistemas de archivo que no dan soporte a atributos

---

<sup>8</sup> Los archivos en `/etc/selinux/targeted/contexts/files/` definen los contextos de los archivos y directorios. Los archivos en este directorio son accedidos por `restorecon` y por `setfiles` para restaurar al valor predeterminado los contextos de los archivos y directorios.

extendidos. Esto es útil si no confía en que un sistema de archivo provea los atributos correctos, por ejemplo, medios removibles en sistemas múltiples. El comando `mount -o context` también se puede usar para dar soporte al etiquetado de sistemas de archivos que no soportan atributos extendidos, tales como la Tabla de Ubicación de Archivos (FAT) o los sistemas de archivo NFS. El contexto especificado con la opción `context` no se escribe al disco: los contextos originales son preservados, y se ven cuando se lo monta sin la opción `context` (si el sistema de archivo ya tenía soporte para atributos extendidos).

Para más información sobre el etiquetado de sistemas de archivo, vaya al artículo de James Morris "Etiquetado del Sistema de Archivo en SELinux": <http://www.linuxjournal.com/article/7426>.

### 5.9.1. Montajes de Contexto

Para montar un sistema de archivo con el contexto especificado, superponiendo los contextos existentes si existieran, o para especificar uno predeterminado distinto para un sistema de archivo que no da soporte para atributos extendidos, como usuario root de Linux, use el comando `mount -o context=SELinux_user:role:type:level` cuando monte el sistema de archivo deseado. Los cambios de contexto no se graban en el disco. Por defecto, los montajes NFS en el lado del cliente se etiquetan con un contexto distinto definido por una política para sistemas de archivo NFS. En políticas comunes, este contexto predeterminado usa el tipo `nfs_t`. Sin las opciones de montaje adicionales, esto podría evitar el que sistemas de archivo NFS sean compartidos vía otros servicios, como el Servidor HTTP Apache. El siguiente ejemplo monta un sistema de archivo NFS para que se pueda acceder a través del Servidor HTTP Apache:

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

Los archivos y directorios recién creados en este sistema de archivo parecen tener un contexto SELinux especificado con `-o contexto`; sin embargo, dado que los cambios del contexto no se escriben en el disco en estas situaciones, el contexto especificado por la opción `-o contexto` sólo se mantiene si se usa la misma opción en la siguiente montada, y si además se especifica el mismo contexto.

La Obligación de Tipo es el control de permiso principal en la política destinada de SELinux. Para la mayor parte, los usuarios y roles de SELinux se pueden ignorar, por lo que, cuando se superponga el contexto de SELinux con `-o context`, use el usuario SELinux `system_u` y el rol `object_r`, y concéntrese en el tipo. Si no está usando la política MLS o seguridad multi-categoría, use el nivel `s0`.



#### Nota

Cuando se monta un sistema de archivo con la opción `context`, los cambios de contexto (por usuarios y procesos) son prohibidos. Por ejemplo, ejecutando `chcon` en un sistema de archivo montado con la opción `context` resulta en un error de `Operación no soportada`.

### 5.9.2. Cambio del Contexto Predeterminado

Como se mencionó en [Sección 5.8, "Los tipos `file\_t` y `default\_t`"](#), en sistemas de archivo que soportan atributos extendidos, cuando se accede a un archivo en disco que le falta el contexto SELinux, se trata como si hubiera tenido el contexto predeterminado tal como se define en la política SELinux. En

políticas comunes, este contexto predeterminados usa el tipo `file_t`. Si se desea usar un contexto predeterminado diferente, monte el sistema de archivo con la opción `defcontext`.

El siguiente ejemplo monta un sistema de archivo recién creado (en `/dev/sda2`) en el directorio recién creado `/prueba/`. Asume que no hay reglas en `/etc/selinux/targeted/contexts/files/` que definan el contexto del directorio `/prueba/`:

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

En este ejemplo:

- la opción `defcontext` define que `system_u:object_r:samba_share_t:s0` es "el contexto de seguridad predeterminado para archivos no etiquetados"<sup>9</sup>.
- cuando sea montado, el directorio raíz (`/prueba/`) del sistema de archivo se trata como si estuviera etiquetado con el contexto especificado por `defcontext` (esta etiqueta no se guarda en el disco). Esto afecta el etiquetado de archivos creados en `/prueba/`: los archivos nuevos heredan el tipo `samba_share_t`, y estas etiquetas se guardan en el disco.
- los archivos creados bajo `/prueba/` mientras el sistema de archivo estaba montado con la opción `defcontext` retendrán sus etiquetas.

### 5.9.3. Montando un Sistema de Archivos NFS

Por defecto, los montajes NFS en el lado del cliente son etiquetados con un contexto predeterminado por la política para los sistemas de archivo NFS. En políticas comunes, este contexto predeterminado usa el tipo `nfs_t`. Dependiendo de la configuración de la política, los servicios, como el Servidor HTTP Apache y MySQL, pueden no poder leer archivos etiquetados con el tipo `nfs_t`. Esto puede prevenir que los sistemas de archivos etiquetados con este tipo se monten y sean leídos o exportados por otros servicios.

Si desea montar un sistema de archivo NFS y leer o exportar ese sistema de archivo con otro servicio, use la opción `contexto` cuando monte para anular el tipo `nfs_t`. Use la siguiente opción de contexto para montar sistemas de archivo NFS para que puedan compartirse vía el Servidor HTTP Apache:

```
mount servidor:/export /local/punto/de/montaje -o\
context="system_u:object_r:httpd_sys_content_t:s0"
```

Dado que los cambios de contexto no se escriben al disco para estas situaciones, el contexto especificado con la opción `context` sólo se retiene si la opción `context` se usa en el siguiente montaje, y si el mismo contexto se especifica.

Como una alternativa a montar sistemas de archivo con la opción `contexto`, los Booleanos se pueden activar para permitir a los servicios acceder sistemas de archivos etiquetados con el tipo `nfs_t`. Vaya a [Sección 5.6.3, "Booleanos para NFS y CIFS"](#) para instrucciones sobre configuración de Booleanos para permitir a servicios acceder al tipo `nfs_t`.

### 5.9.4. Montajes NFS Múltiples

Cuando se monten múltiples montajes desde el mismo NFS exportado, el intento de sobrescribir el contexto de SELinux e cada montaje con un contexto diferente, resulta en fallos de los comandos de



montaje subsecuentes. En el siguiente ejemplo, el servidor NFS tiene un exportado único, **/export**, que tiene dos subdirectorios, **web/** and **database/**. El siguiente comando intenta dos montajes desde un único export NFS e intenta sobrescribir el contexto para cada uno:

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

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

El segundo comando mount falla, y se graba lo siguiente en **/var/log/messages**:

```
kernel: SELinux: montaje inválido. Similar superblock, configuración de
seguridad diferente para (dev 0:15, tipo nfs)
```

Para montar montajes múltiples de un exportado NFS único, con cada montaje teniendo un contexto diferente, use las opciones **-o nosharecache, context**. El siguiente ejemplo monta montajes múltiples de un único export de NSF, con un contexto diferente para cada montaje (permitiendo un único acceso de servicio a cada uno):

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

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

En este ejemplo, **server:/export/web** se monta localmente en **/local/web/**, con todos los archivos etiquetados con el tipo **httpd\_sys\_content\_t**, lo que permite el acceso al Servidor HTTP Apache. **server:/export/database** está montado localmente en **/local/database**, con los archivos etiquetados con el tipo **mysql\_db\_t**, lo que permite a MySQL el acceso. Estos cambios de tipo no se escriben en el disco.



### Importante

Las opciones **nosharecache** le permiten montar el mismo subdirectorio de un exportado varias veces con distintos contextos (por ejemplo, montar **/export/web** varias veces). No monte el mismo directorio de un exportado varias veces con distintos contextos, dado que esto crea un montado solapado, donde los archivos se pueden acceder con dos contextos diferentes.

## 5.9.5. Haciendo Persistente los Contextos de Montajes

Para hacer que los contextos de montajes persistentes entre remontadas y reiniciadas, agregue las entradas de los sistemas de archivos en **/etc/fstab** o un mapa de automontador, y use el contexto deseado como una opción de montaje. El siguiente ejemplo agrega una entrada en **/etc/fstab** para un montaje de contexto NFS:

```
servidor:/export /local/montaje/ nfs
context="system_u:object_r:httpd_sys_content_t:s0" 0 0
```

Vaya a la [Guía de Despliegue del Linux para Empresas de Red Hat 5, Sección 19.2. "Configuración de Cliente NFS"](#)<sup>10</sup> para información adicional sobre montaje de sistemas de archivo NFS.

### 5.10. Mantenimiento de las Etiquetas de SELinux

Estas secciones describen qué les pasa a los contextos SELinux cuando se copia, mueve y compacta archivos y directorios. También explica cómo preservar los contextos cuando se copia o se compacta.

#### 5.10.1. Copia de Directorios y Archivos

Cuando se copia un archivo o directorio, se crea uno si no existía. El contexto de ese archivo o directorio nuevo está basado en las reglas de etiquetados predeterminada, y no el contexto original del archivo o directorio (a menos que se usen opciones para preservar el contexto original). Por ejemplo, los archivos creados en los directorios home de los usuarios se etiquetan con el tipo `user_home_t`:

```
$ touch archivo1
$ ls -Z archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 archivo1
```

Si un archivo se copia a otro directorio, tal como `/etc/`, el archivo nuevo se crea de acuerdo a las reglas de etiquetado predeterminado del directorio `/etc/`. El copiado de un archivo (sin opciones adicionales) puede no preservar el contexto original:

```
$ ls -Z archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 archivo1
# cp archivo1 /etc/
$ ls -Z /etc/archivo1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/archivo1
```

Cuando el **archivo1** se copia a `/etc/`, si `/etc/archivo1` no existe, `/etc/archivo1` se crea como un archivo nuevo. Como se muestra en el ejemplo de arriba, `/etc/archivo1` se etiqueta con el tipo `etc_t`, de acuerdo con las reglas de etiquetado predeterminadas.

Cuando un archivo se copia sobre otro existente, el contexto del archivo existente se preserva, a menos que el usuario especifique opciones de `cp` para preservar el contexto del archivo original, tal como `--preserve=context`. La política de SELinux puede prevenir que se preserven los contextos al copiar.

#### Copia sin Preservar los Contextos de SELinux

Cuando se copia un archivo con el comando `cp`, si no se dan opciones, el tipo se hereda desde el directorio padre destino:

```
$ touch archivo1
$ ls -Z archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 archivo1
$ ls -dZ /var/www/html/
drwxr-xr-x root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

---

<sup>10</sup> [http://www.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5.2/html/Deployment\\_Guide/s1-nfs-client-config.html](http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/s1-nfs-client-config.html)

```
# cp archivo1 /var/www/html/
$ ls -Z /var/www/html/archivo1
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/
www/html/archivo1
```

En este ejemplo, **archivo1** se crea en el directorio de inicio del usuario, y se etiqueta con el tipo `user_home_t`. El directorio `/var/www/html/` está etiquetado con el tipo `httpd_sys_content_t`, como se muestra con el comando `ls -dZ /var/www/html/`. Cuando el **archivo1** se copia a `/var/www/html/`, hereda el tipo `httpd_sys_content_t`, como se muestra con el comando `ls -Z /var/www/html/archivo1`.

### Preservación de los Contextos de SELinux cuando se copia

Use el comando `cp --preserve=context` para preservar los contextos cuando se copia:

```
$ touch archivo1
$ ls -Z archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 archivo1
$ 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 archivo1 /var/www/html/
$ ls -Z /var/www/html/archivo1
-rw-r--r-- root root unconfined_u:object_r:user_home_t:s0 /var/www/html/
archivo1
```

En este ejemplo, el **archivo1** se crea en el directorio de inicio del usuario, y se etiqueta con el tipo `user_home_t`. El directorio `/var/www/html/` está etiquetado con el tipo `httpd_sys_content_t`, como se muestra con el comando `ls -dZ /var/www/html/`. Usando la opción `--preserve=context` se mantienen los contextos de SELinux durante operaciones de copia. Como se muestra con el comando `ls -Z /var/www/html/archivo1`, el tipo `user_home_t` del **archivo1** fue preservado cuando el archivo se copió a `/var/www/html/`.

### Copiado y Cambio del Contexto

Use el comando `cp -Z` para cambiar el contexto destino de copia. El siguiente ejemplo se realizó en el directorio de inicio del usuario:

```
$ touch archivo1
$ cp -Z system_u:object_r:samba_share_t:s0 archivo1 archivo2
$ ls -Z archivo1 archivo2
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 archivo1
-rw-rw-r-- usuario1 grupo1 system_u:object_r:samba_share_t:s0 archivo2
$ rm archivo1 archivo2
```

En este ejemplo, el contexto se define en la opción `-Z`. Sin la opción `-Z`, **archivo2** se etiquetaría con el contexto `unconfined_u:object_r:user_home_t`.

### Copia de un Archivos sobre un otro existente

Cuando un archivo se copia sobre otro existente, el contexto del archivo existente se preserva (a menos que se use una opción para preservar los contextos). Por ejemplo:

```
# touch /etc/archivo1
# ls -Z /etc/archivo1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/archivo1
# touch /tmp/archivo2
# ls -Z /tmp/archivo2
-rw-r--r-- root root unconfined_u:object_r:user_tmp_t:s0 /tmp/archivo2
# cp /tmp/archivo2 /etc/archivo1
# ls -Z /etc/archivo1
-rw-r--r-- root root unconfined_u:object_r:etc_t:s0 /etc/archivo1
```

En este ejemplo, se crean dos archivos: **/etc/archivo1**, etiquetado con el tipo `etc_t`, y **/tmp/archivo2**, etiquetado con el tipo `user_tmp_t`. El comando `cp /tmp/archivo2 /etc/archivo1` sobrescribe **archivo1** con **archivo2**. Después de copiar, el comando `ls -Z /etc/archivo1` muestra a **archivo1** etiquetado con el tipo `etc_t`, en vez del `user_tmp_t` de **/tmp/archivo2** que reemplazó a **/etc/archivo1**.



### Importante

Copie archivos y directorios, en vez de moverlos. Esto ayuda a asegurar que se etiquetan con los contextos de SELinux correctos. Los contextos SELinux incorrectos pueden hacer que los procesos no puedan acceder a esos archivos y directorios.

## 5.10.2. Movimiento de Archivos y Directorios

Los archivos y directorios mantienen su contexto SELinux actual cuando se mueven. En muchos casos, esto es incorrecto para la ubicación nueva a donde se los mueve. El siguiente ejemplo muestra la movida de un archivo desde el directorio de inicio del usuario a **/var/www/html/**, que es usado por el Servidor HTTP Apache. Dado que el archivo es movido, no hereda el contexto SELinux correcto:

1. Ejecute el comando `cd` sin ningún argumento para cambiar a su directorio de inicio. Una vez ahí, ejecute el comando `touch archivo1` para crear un archivo. Este archivo se etiqueta con el tipo `user_home_t`:

```
$ ls -Z archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0
archivo1
```

2. Ejecute el comando `ls -dZ /var/www/html/` para ver el contexto de SELinux del directorio **/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/
```

Por defecto, el directorio **/var/www/html/** se etiqueta con el tipo `httpd_sys_content_t`. Los archivos y directorios creados bajo el directorio **/var/www/html/** heredan este tipo, y como tal, son etiquetados con este tipo.

- Como usuario root de Linux, ejecute el comando `mv archivo1 /var/www/html/` para mover el **archivo1** al directorio `/var/www/html/`. Dado que el archivo es movido, mantiene su tipo `user_home_t` actual:

```
# mv archivo1 /var/www/html/
# ls -Z /var/www/html/archivo1
-rw-rw-r-- usuario1 grupo1 unconfined_u:object_r:user_home_t:s0 /var/
www/html/archivo1
```

Por defecto, el Servidor HTTP Apache no puede leer archivos etiquetados con el tipo `user_home_t`. Si todos los archivos de una página web se etiquetaron con `user_home_t`, u otro tipo al que el Servidor HTTP Apache no puede leer, el permiso es negado cuando intente accederlo vía Firefox o algún otro navegador web basado en texto.



### Importante

Mover archivos y directorios con el comando `mv` puede resultar en el contexto SELinux incorrecto, evitando que los procesos tales como el Servidor HTTP Apache y Samba puedan acceder a tales archivos y directorios.

### 5.10.3. Chequeando el Contexto SELinux Predeterminado

Use el comando `/usr/sbin/matchpathcon` para chequear si los archivos y directorios tienen el contexto SELinux correcto. De la página de manual de `matchpathcon(8)`: "**matchpathcon** consulta la política del sistema y muestra el contexto de seguridad predeterminado asociado con una dirección de archivo."<sup>11</sup>. El siguiente ejemplo muestra el uso del comando `/usr/sbin/matchpathcon` para verificar que los archivos dentro del directorio `/var/www/html/` están etiquetados correctamente:

- Como usuario root de Linux, ejecute el comando `touch /var/www/html/archivo{1,2,3}` para crear tres archivos (**archivo1**, **archivo2** y **archivo3**). Estos heredan el tipo `httpd_sys_content_t` del directorio `/var/www/html/`:

```
# touch /var/www/html/archivo{1,2,3}
# ls -Z /var/www/html/
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
archivo1
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
archivo2
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
archivo3
```

- Como usuario root de Linux, ejecute el comando `chcon -t samba_share_t /var/www/html/archivo1` para cambiar el tipo del **archivo1** a `samba_share_t`. Nota: El Servidor HTTP Apache no puede leer archivos o directorios etiquetados con el tipo `samba_share_t`.
- La opción `/usr/sbin/matchpathcon -V` compara el contexto SELinux actual con el contexto predeterminado correcto dado por la política de SELinux. Ejecute el comando `/usr/sbin/`

<sup>11</sup> La página de manual `matchpathcon(8)`, tal como se incluyó en el paquete `libselinux-utils` en Fedora, fue escrita por Daniel Walsh. Cualquier edición o cambio en esta versión fue hecha por Murray McAllister.

**matchpathcon -V /var/www/html/\*** para chequear todos los archivos del directorio **/var/www/html/**:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/archivo1 tiene el contexto
unconfined_u:object_r:samba_share_t:s0, debería ser
system_u:object_r:httpd_sys_content_t:s0
/var/www/html/archivo2 verified.
/var/www/html/archivo3 verified.
```

La siguiente salida del comando **/usr/sbin/matchpathcon** explica que el **archivo1** está etiquetado con el tipo **samba\_share\_t**, pero debería estar etiquetado con el tipo **httpd\_sys\_content\_t**:

```
/var/www/html/archivo1 tiene el contexto
unconfined_u:object_r:samba_share_t:s0, debería tener
system_u:object_r:httpd_sys_content_t:s0
```

Para resolver el problema de etiqueta y permitir al Servidor HTTP Apache acceder a **archivo1**, como usuario root de Linux corra el comando **/sbin/restorecon -v /var/www/html/archivo1**:

```
# /sbin/restorecon -v /var/www/html/archivo1
restorecon reset /var/www/html/archivo1 context
unconfined_u:object_r:samba_share_t:s0-
>system_u:object_r:httpd_sys_content_t:s0
```

### 5.10.4. Archivando archivos con tar

**tar** no retiene los atributos extendidos por defecto. Dado que los contextos SELinux se almacenan en los atributos extendidos, los contextos se pueden perder cuando se compactan archivos. Use **tar --selinux** para crear archivos que retengan los contextos. Si un archivo Tar contiene archivos sin los atributos extendidos, o si quiere que los atributos extendidos coincidan con los predeterminados del sistema, ejecute el archivado a través de **/sbin/restorecon**:

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

Nota: dependiendo del directorio, puede necesitar ser el usuario root de Linux para ejecutar el comando **/sbin/restorecon**.

El siguiente ejemplo muestra la creación de un archivo Tar que mantiene sus contextos SELinux:

1. Como usuario root de Linux, ejecute el comando **touch /var/www/html/archivo{1,2,3}** para crear tres archivos (**archivo1**, **archivo2** y **archivo3**). Estos heredan el tipo **httpd\_sys\_content\_t** del directorio **/var/www/html/**:

```
# touch /var/www/html/archivo{1,2,3}
# ls -Z /var/www/html/
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
archivo1
```

```
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
archivo2
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
archivo3
```

- Ejecute el comando `cd /var/www/html/` para cambiar al directorio `/var/www/html/`. Una vez en este directorio, como usuario root de Linux ejecute el comando `tar --selinux -cf prueba.tar archivo{1,2,3}` para crear un archivo Tar con nombre `prueba.tar`.
- Como usuario root de Linux, corra el comando `mkdir /prueba` para crear un directorio nuevo, y luego ejecute el comando `chmod 777 /prueba/` para permitir a los usuarios acceso total al directorio `/prueba/`.
- Ejecute el comando `cp /var/www/html/prueba.tar /prueba/` para copiar el archivo `prueba.tar` en el directorio `/prueba/`.
- Ejecute el comando `cd /prueba/` para cambiar al directorio `/test/`. Una vez ahí, ejecute el comando `tar -xvf prueba.tar` para extraer el archivo Tar.
- Ejecute el comando `ls -lZ /prueba/` para ver los contextos SELinux. El tipo `httpd_sys_content_t` fue retenido, en vez de haberse cambiado al `default_t`, lo que hubiera pasado si la opción `--selinux` no se hubiera usado:

```
$ ls -lZ /prueba/
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:httpd_sys_content_t:s0
archivo1
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:httpd_sys_content_t:s0
archivo2
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:httpd_sys_content_t:s0
archivo3
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:default_t:s0
prueba.tar
```

- Si el directorio `/prueba/` no se necesita más, como usuario root de Linux ejecute el comando `rm -ri /prueba/` para eliminarlo, así como todos los archivos en él.

Vaya a la página man de `tar(1)` para información adicional de `tar`, tal como la opción `--xattrs` para retener todos los atributos extendidos.

### 5.10.5. Archivando archivos con tar

`star` no retiene los atributos extendidos por defecto. Dado que los contextos SELinux se almacenan en los atributos extendidos, los contextos se pueden perder cuando se crean esos archivos. Use `star -xattr -H=exustar` para crear archivos que retengan los contextos. El paquete `star` no se instala por defecto. Para instalar `star`, ejecute el comando `yum install star` como usuario root de Linux.

El siguiente ejemplo muestra la creación de un archivo Star que retiene los contextos SELinux:

- Como usuario root de Linux, ejecute el comando `touch /var/www/html/archivo{1,2,3}` para crear tres archivos (`archivo1`, `archivo2` y `archivo3`). Estos heredan el tipo `httpd_sys_content_t` del directorio `/var/www/html/`:

```
# touch /var/www/html/archivo{1,2,3}
# ls -Z /var/www/html/
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
  archivo1
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
  archivo2
-rw-r--r-- root root unconfined_u:object_r:httpd_sys_content_t:s0
  archivo3
```

2. Ejecute el comando **cd /var/www/html/** para cambiar al directorio **/var/www/html/**. Una vez en este directorio, como usuario root de Linux ejecute el comando **star -xattr -H=exustar -c -f=prueba.star archivo{1,2,3}** para crear un archivo Star llamado **prueba.star**:

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

3. Como usuario root de Linux, corra el comando **mkdir /prueba** para crear un directorio nuevo, y luego ejecute el comando **chmod 777 /prueba/** para permitir a los usuarios acceso total al directorio **/prueba/**.
4. Ejecute el comando **cp /var/www/html/prueba.star /prueba/** para copiar el archivo **prueba.star** al directorio **/prueba/**.
5. Ejecute el comando **cd /prueba/** para cambiar al directorio **/prueba/**. Una vez ahí, ejecute el comando **star -x -f=prueba.star** para extraer el archivo Star:

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

6. Ejecute el comando **ls -lZ /prueba/** para ver los contextos SELinux. El tipo **httpd\_sys\_content\_t** fue retenido, en vez de haberse cambiado al **default\_t**, lo que hubiera pasado si la opción **--selinux** no se hubiera usado:

```
$ ls -lZ /prueba/
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:httpd_sys_content_t:s0
  archivo1
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:httpd_sys_content_t:s0
  archivo2
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:httpd_sys_content_t:s0
  archivo3
-rw-r--r-- usuario1 grupo1 unconfined_u:object_r:default_t:s0
  prueba.star
```

7. Si el directorio **/prueba/** no se necesita más, como usuario root de Linux ejecute el comando **rm -ri /prueba/** para eliminarlo, así como todos los archivos en él.
8. Si **star** ya no se necesita, como usuario root de Linux, ejecute el comando **yum remove star** para eliminar el paquete.



Vaya a la página man de star(1) para más información acerca de **star**.



---

# Confinando a los Usuarios

Un número de usuarios confinados SELinux están disponibles en Fedora 11. Cada usuario Linux se mapea aun usuario SELinux vía la política de SELinux, permitiendo a los usuarios Linux heredar las restricciones sobre los usuarios SELinux, por ejemplo (dependiendo del usuario), no poder: ejecutar el sistema de ventanas X; usar la red; ejecutar aplicaciones setuid (a menos que la política de SELinux lo permita); o ejecutar comandos **su** y **sudo** para volverse el usuario root de Linux. Esto ayuda a proteger el sistema del usuario. Vaya a [Sección 4.3, “Usuarios Confinados y no Confinados”](#) para información adicional sobre usuarios confinados en Fedora 11.

## 6.1. Linux y los Mapeos de Usuarios de SELinux

Como usuario root de Linux, corra el comando **semanage login -l** para ver el mapeo entre los usuarios de Linux y los usuarios de SELinux:

```
# /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

En Fedora 11, los usuarios Linux se mapean al ingreso SELinux `__default__` por defecto (que se mapea al usuario SELinux `unconfined_u`). Cuando se crea un usuario Linux con el comando **useradd**, si no se especifica ninguna opcion, son mapeados al usuario SELinux `unconfined_u`. Lo siguiente define el mapeo por defecto:

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

## 6.2. Confinando Usuarios Nuevos de Linux: useradd

Los usuarios Linux mapeados al usuario SELinux `unconfined_u` corren en el dominio `unconfined_t`. Esto se ve ejecutando el comando **id -Z** luego de haber ingresado como el usuario Linux que se mapea a `unconfined_u`:

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

Cuando los usuarios linux ejecuten en el dominio `unconfined_t`, se aplican las reglas de la política de SELinux, pero las reglas de políticas que existen para usuarios Linux que corren en el dominio `unconfined_t` permiten casi todos los accesos. Si los usuarios Linux no confinados ejecutan una aplicación que la política de SELinux define pueden transicionar desde el dominio `unconfined_t` a su propio dominio confinado, los usuarios Linux no confinados todavía pueden ser sujetos a restricciones del dominio confinado. El beneficio de seguridad de esto es que, aunque el usuario Linux corre en un dominio confinado, la aplicación permanece confinada, y por lo tanto, la explotación de una brecha en la aplicación se puede limitar por la política. Nota: esto no protege al sistema del usuario. En su lugar, el usuario y el sistema están siendo protegido de posibles daños causados en alguna debilidad en la aplicación.

Cuando se crean usuarios Linux con **useradd**, use la opción **-Z** para especificar a qué usuario SELinux se debe mapear. El siguiente ejemplo crea un usuario Linux nuevo, **useruser**, y mapea ese usuario al usuario SELinux **user\_u**. Los usuarios Linux mapeados al usuario SELinux **user\_u** corren en el dominio **user\_t**. En este dominio, los usuarios Linux no pueden correr aplicaciones **setuid** a menos que la política de SELinux lo permita (tal como **passwd**), y tampoco pueden correr **su** o **sudo**, lo que evita que se puedan volver usuarios root de Linux con estos comandos.

1. Como usuario root de Linux, corra el comando **/usr/sbin/useradd -Z user\_u useruser** para crear el usuario Linux nuevo (**useruser**) que se mapeará al usuario SELinux **user\_u**.
2. Como usuario root de Linux, corra el comando **semanage login -l** para ver el mapeo entre el usuario Linux **useruser** y **user\_u**:

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

Login Name	SELinux User	MLS/MCS Range
<b>__default__</b>	<b>unconfined_u</b>	<b>s0-s0:c0.c1023</b>
<b>root</b>	<b>unconfined_u</b>	<b>s0-s0:c0.c1023</b>
<b>system_u</b>	<b>system_u</b>	<b>s0-s0:c0.c1023</b>
<b>useruser</b>	<b>user_u</b>	<b>s0</b>

3. Como usuario root de Linux, corra el comando **passwd useruser** para asignar una contraseña para el usuario **useruser** de Linux:

```
# passwd useruser
```

Cambiando la contraseña del usuario **useruser**.

Nueva contraseña de UNIX: *Ingrese una contraseña*

Reingrese la nueva contraseña de UNIX: *Ingrese la misma contraseña de nuevo*

**passwd**: todos los tokens actualizados exitosamente.

4. Salga de su sesión actual, e ingrese como el usuario **useruser** de Linux. Cuando ingrese, **pam\_selinux** mapea el usuario Linux a un usuario SELinux (en este caso, **user\_u**), y configura el contexto SELinux resultante. El shell del usuario Linux es luego lanzado con este contexto. Corra el comando **id -Z** para ver el contexto de un usuario Linux:

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

5. Salga de la sesión **useruser** de Linux, y vuelva a ingresar en su cuenta. Si no quiere el usuario **useruser**, ejecute el comando **/usr/sbin/userdel -r useruser** como usuario root de Linux para borrarlo junto con su directorio de inicio.

### 6.3. Confinando Usuarios Linux Existentes: **semanage login**

Si un usuario Linux se mapea al usuario **unconfined\_u** (el comportamiento predeterminado), y desea cambiar le usuario SELinux al que se mapea, use el comando **semanage login**. El siguiente

ejemplo crea un usuario de Linux nuevo llamado `usuarionuevo`, luego lo mapea al usuario SELinux `user_u`:

1. Como usuario root de Linux, ejecute el comando `/usr/sbin/useradd usuarionuevo` para crear un nuevo usuario (`usuarionuevo`). Dado que este usuario usa el mapeo por defecto, no aparece en la salida de `/usr/sbin/semanage login -l`:

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

Login Name	SELinux User	MLS/MCS Range
<code>__default__</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>root</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>system_u</code>	<code>system_u</code>	<code>s0-s0:c0.c1023</code>

2. Para mapear un usuario `usuarionuevo` de Linux al usuario `user_u` de SELinux, corra el siguiente comando como usuario root de Linux:

```
/usr/sbin/semanage login -a -s user_u usuarionuevo
```

La opción `-a` agrega un registro nuevo y la opción `-s` especifica el usuario SELinux al que mapea el usuario Linux. El último argumento `usuarionuevo`, es el usuario Linux al que quiere que se mapee el usuario SELinux especificado.

3. Para ver el mapeo entre el usuario `usuarionuevo` de Linux y `user_u`, corra el comando `semanage login -l` como usuario root de Linux:

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

Login Name	SELinux User	MLS/MCS Range
<code>__default__</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>usuarionuevo</code>	<code>user_u</code>	<code>s0</code>
<code>root</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>system_u</code>	<code>system_u</code>	<code>s0-s0:c0.c1023</code>

4. As the Linux root user, run the `passwd newuser` command to assign a password to the Linux `newuser` user:

```
# passwd usuarionuevo
Cambiando la contraseña del usuario usuarionuevo.
Nueva contraseña de UNIX: Ingrese una contraseña
Reingrese la nueva contraseña de UNIX: Ingrese la misma contraseña de nuevo
passwd: todos los tokens de autenticación se actualizaron exitosamente.
```

5. Salga de su sesión actual e ingrese como el usuario de Linux `usuarionuevo`. Corra el comando `id -Z` para ver el contexto SELinux de `usuarionuevo`:

```
[usuarionuevo@localhost ~]$ id -Z
```

```
user_u:user_r:user_t:s0
```

- Salga de la sesión de Linux del usuario `usuario_nuevo` y vuelva a ingresar en su cuenta. Si no quiere el usuario `usuario_nuevo` de Linux, corra el comando `userdel -r usuario_nuevo` como usuario root de Linux, junto con su directorio de inicio. También, el mapeo del usuario Linux `usuario_nuevo` y `user_u` se elimina:

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

Login Name	SELinux User	MLS/MCS Range
<code>__default__</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>root</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>system_u</code>	<code>system_u</code>	<code>s0-s0:c0.c1023</code>

### 6.4. Cambiando el Mapeo Predeterminado

En Fedora 11, los usuarios Linux se mapean al ingreso SELinux `__default__` por defecto (que se mapea al usuario SELinux `unconfined_u`). Si quiere que los usuarios nuevos de Linux y los usuarios Linux no mapeados específicamente a un usuario SELinux sean confinados por defecto, cambie el mapeo predeterminado con el comando `semanage login`.

Por ejemplo, corra el siguiente comando como usuario root de Linux para cambiar el mapeo predeterminado de `unconfined_u` a `user_u`:

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

Corra el comando `semanage login -l` como usuario root de Linux para verificar que el ingreso `__default__` se mapea a `user_u`:

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

Login Name	SELinux User	MLS/MCS Range
<code>__default__</code>	<code>user_u</code>	<code>s0</code>
<code>root</code>	<code>unconfined_u</code>	<code>s0-s0:c0.c1023</code>
<code>system_u</code>	<code>system_u</code>	<code>s0-s0:c0.c1023</code>

Si un usuario Linux nuevo se crea y el usuario SELinux no se especifica, o si un usuario Linux existente ingresa y no coincide una entrada específica de la salida de `semanage login -l`, se mapean a `user_u`, según el ingreso `__default__`.

Para volver al comportamiento predeterminado, corra el siguiente comando como usuario root de Linux para mapear el ingreso `__default__` al usuario SELinux `unconfined_u`:

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

## 6.5. xguest: Modo Kiosk

El paquete *xguest* provee una cuenta de usuario kiosk. Esta cuenta se usa para asegurar máquinas a las que ingresan personas y las usan, como las de las bibliotecas, bancos, aeropuertos, quioscos de información y cyber cafés. La cuenta de usuario kiosk está muy bloqueada: esencialmente, sólo permite a los usuarios ingresar y usar **Firefox** para navegar sitios de Internet. Cualquier cambio hecho mientras se ingresó con esa cuenta, tal como la creación y cambio de la configuración, se pierde cuando se sale.

Para configurar la cuenta kiosk:

1. Como usuario root de Linux, corra el comando **yum install xguest** para instalar el paquete *xguest*. Instale las dependencias requeridas.
2. Para permitir a la cuenta kiosk usarse para una variedad de personas, la cuenta no se protege con contraseña, y como tal, la cuenta sólo se puede proteger si SELinux está funcionando en modo obediente. Antes de ingresar con esta cuenta, use el comando **getenforce** para confirmar que SELinux está funcionando en modo obediente:

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

Si no es éste el caso, vaya a [Sección 5.5, “Modos de SELinux”](#) para información acerca del cambio de modo de obligatorio. No es posible ingresar con esta cuenta si SELinux está en modo permisivo o deshabilitado.

3. Solamente puede ingresar a esta cuenta a través de la Administración de Pantalla de GNOME (GDM). Una vez que el paquete *xguest* se instala, se agrega una cuenta Invitado a GDM. Para ingresar, haga clic en la cuenta Invitado:



## 6.6. Booleanos para que los Usuarios Ejecuten Aplicaciones

No permitir que los usuarios Linux ejecuten aplicaciones (que heredan los permisos del usuario) en sus directorios de inicio (home) y **/tmp/**, a los que tienen acceso de escritura, lo que ayuda a evitar

que aplicaciones con brechas o maliciosas puedan modificar archivos del usuario. En Fedora 11, por defecto, los usuarios Linux en los dominios `guest_t` y `xguest_t` no pueden ejecutar aplicaciones en los directorios de inicio o en `/tmp/`; however, por defecto, los usuarios Linux en los dominios `user_t` and `staff_t` si pueden.

Hay booleanos disponibles para cambiar este comportamiento, y se configuran con el comando **setsebool**. El comando **setsebool** se debe usar con el usuario root de Linux. El comando **setsebool -P** hace los cambios persistentes. No use la opción `-P` si no quiere que los cambios persistan entre reiniciadas:

### `guest_t`

Para *permitir* a los usuarios Linux en el dominio `guest_t` que ejecuten aplicaciones en sus directorios de inicio y en `/tmp/`:

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

### `xguest_t`

Para *permitir* a los usuarios Linux en el dominio `xguest_t` ejecutar aplicaciones en sus directorios inicios y `/tmp/`:

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

### `user_t`

Para *impedir* que los usuarios Linux en el dominio `user_t` ejecuten aplicaciones en sus directorios de inicio y `/tmp/`:

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

### `staff_t`

Para *impedir* que los usuarios Linux en el dominio `staff_t` ejecuten aplicaciones en sus directorios de inicio y en `/tmp/`:

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



---

# Solución a Problemas

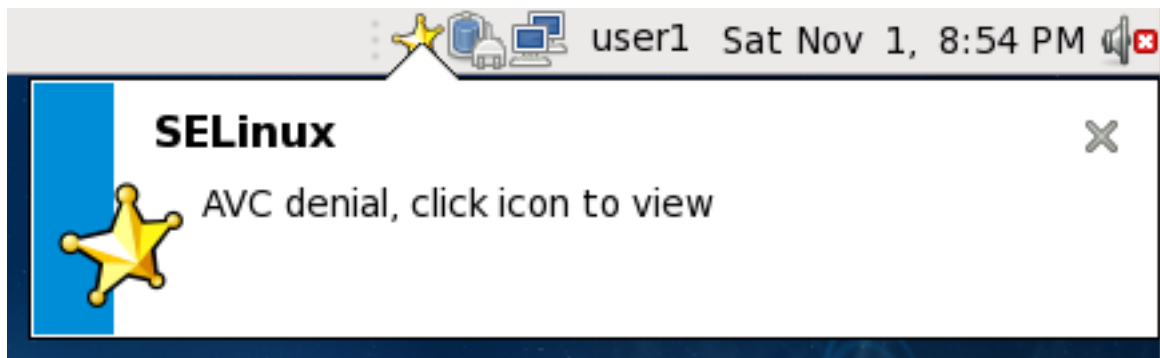
El siguiente capítulo describe qué pasa cuando SELinux niega el acceso; las principales tres causas de problemas; dónde encontrar información acerca del correcto etiquetado; análisis de las negaciones de SELinux; y creación de módulos de políticas personalizados con **audit2allow**.

## 7.1. Qué pasa cuando el Acceso es Denegado

Las decisiones de SELinux, tales como permitir o negar el acceso, son cacheadas. Este cache se conoce como Caché Vector de Acceso (AVC). Los mensajes de negación se registran cuando SELinux niega el acceso. Estas negaciones se conocen también como "negaciones AVC", y se guardan en una ubicación distinta, dependiendo de los demonios que se ejecutan:

Daemon	Log Location
auditd on	<b>/var/log/audit/audit.log</b>
auditd off; rsyslogd on	<b>/var/log/messages</b>
setroubleshootd, rsyslogd, and auditd on	<b>/var/log/audit/audit.log</b> . Easier-to-read denial messages also sent to <b>/var/log/messages</b>

Si está corriendo el Sistema de Ventanas X, ya tienen los paquetes *setroubleshoot* y *setroubleshoot-server* instalados, y los demonios *setroubleshootd* y *auditd* están ejecutándose, se muestra una estrella amarilla y una advertencia cuando SELinux deniega el acceso:



Haciendo clic en las estrellas se presenta un análisis detallado de por qué SELinux negó el acceso, y una posible solución para permitir el acceso. Si no está corriendo el Sistema de Ventanas X, es menos obvio cuando un acceso es negado por SELinux. Por ejemplo, los usuarios navegando su sitio web pueden recibir un error similar al siguiente:

Prohibido

No tiene permiso para acceder el *archivo xx* en este servidor

Para estas situaciones, si las reglas DAC (los permisos estándares de Linux) permiten el acceso, busque en **/var/log/messages** y **/var/log/audit/audit.log** la cadena de los errores "SELinux está negando" y "negado" respectivamente. Esto se puede hacer con los siguientes comandos como usuario root de Linux:

```
grep "SELinux está previniendo" /var/log/messages
```

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

### 7.2. Tres Principales Causas de Problemas

Las siguientes secciones describen las tres principales causas de problemas: problemas de etiquetados, configuración de Booleanos y puertos para servicios, y la evolución de las reglas SELinux.

#### 7.2.1. Problemas de Etiquetados

En sistemas que corren SELinux, todos los procesos y archivos se etiquetan con una etiqueta que contiene información de seguridad relevante. Esta información se llama contexto de SELinux. Si estas etiquetas están mal, el acceso puede ser negado. Si una aplicación se etiqueta incorrectamente, el proceso al que transiciona puede no tener la etiqueta correcta, causando negaciones de acceso de SELinux, y los procesos pueden crear archivo con las etiquetas incorrectas.

Una causa común de problemas de etiquetados es cuando un directorio no estandar se usa para un servicio. Por ejemplo, en vez de usar `/var/www/html/` para un sitio web, un administrador prefiere usar `/srv/myweb/`. En Fedora 11, el directorio `/srv/` se etiqueta con el tipo `var_t`. Los archivos y directorios creados en `/srv/` heredan este tipo. También, los directorios de alto nivel recién creados (como `/miservidor/`) puede ser etiquetado con el tipo `default_t`. SELinux impide al Servidor HTTP Apache (`httpd`) el acceso a estos dos tipos. Para permitirle el acceso, SELinux debe saber qué archivos en `/srv/myweb/` son accesibles a `httpd`:

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

Este comando **semanage** agrega el contexto para el directorio `/srv/myweb/` (y todos los archivos dentro de él) a la configuración de contexto de archivos de SELinux <sup>1</sup>. El comando **semanage** no cambia el contexto. Como usuario root de Linux, ejecute el comando **restorecon** para aplicar los cambios:

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

Vaya a [Sección 5.7.2, “Cambios Persistentes: \*semanage fcontext\*”](#) para más información sobre agregado de contextos a la configuración de contexto de archivo.

##### 7.2.1.1. ¿Cuál es el contexto correcto?

El comando **matchpathcon** chequea el contexto de un nombre completo de archivo y lo compara con la etiqueta por defecto para esa dirección. El siguiente ejemplo muestra el uso de **matchpathcon** en un directorio con archivos etiquetados incorrectamente:

```
$ /usr/sbin/matchpathcon -V /var/www/html/*
/var/www/html/index.html tiene el contexto
unconfined_u:object_r:user_home_t:s0, debería ser
system_u:object_r:httpd_sys_content_t:s0
/var/www/html/page1.html tiene el contexto
unconfined_u:object_r:user_home_t:s0, debería ser
system_u:object_r:httpd_sys_content_t:s0
```

---

<sup>1</sup> Los archivos en `/etc/selinux/targeted/contexts/files/` definen los contextos de archivos y directorios. Los archivos en este directorio son leídos por **restorecon** y **setfiles** para restaurar archivos y directorios a sus contextos predeterminados.

En este ejemplo, los archivos **index.html** and **pagina1.html** se etiquetan con el tipo `user_home_t`. Este tipo se usa para archivos en los directorios de inicio de los usuarios. Usando el comando `mv` para mover archivos puede resultar en archivos etiquetados con el tipo `user_home_t`. Este tipo no debería existir fuera de los directorios `home`. Use el comando **restorecon** para restaurar tales archivos a su tipo correcto:

```
# /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
```

Para restaurar el contexto de todos los archivos bajo un directorio, use la opción `-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
```

Vaya a [Sección 5.10.3, “Chequeando el Contexto SELinux Predeterminado”](#) para un ejemplo más detallado de **matchpathcon**.

## 7.2.2. ¿Cómo se Ejecutan los Servicios Confinados?

Los servicios se pueden ejecutar en una variedad de formas. Para cambiar esto, debe decirle a SELinux cómo correrá los servicios. Esto se puede conseguir vía los Booleanos que permiten que parte de las políticas de SELinux se cambien en tiempo de ejecución, sin ningún conocimiento sobre la escritura de políticas de SELinux. Esto permite cambios, tales como permitir a servicios que accedan a sistemas de archivo NFS, sin recargar o recompilar una política SELinux. También, correr servicios en números de puerto no predeterminados requiere que la configuración de la política se actualice vía el comando **semanage**.

Por ejemplo, para permitir al Servidor HTTP Apache comunicarse con MySQL, active el Booleano `httpd_can_network_connect_db`:

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

Si el acceso es denegado para un servicio particular, use los comandos **getsebool** y **grep** para ver si algún Booleano está disponible para permitir el acceso. Por ejemplo, use el comando **getsebool -a | grep ftp** para buscar un Booleano relacionado con 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
```

Para una lista de los Booleanos y de si están activos o inactivos, corra el comando `/usr/sbin/getsebool -a`. Para una lista de los Booleanos y una explicación de lo que son, y de si están activos o no, ejecute el comando `/usr/sbin/semanage boolean -l` como usuario root de Linux. vaya a [Sección 5.6, “Booleanos”](#) para información sobre listado y configuración de Booleanos.

### Números de Puertos

Dependiendo de la configuración de la política, los servicios pueden tener permitido correr sobre ciertos números de puerto. Intentar cambiar el puerto en el que corre un servicio sin cambiar la política puede resultar en un fallo al iniciar el servicio. Por ejemplo, ejecute el comando `semanage port -l | grep http` como usuario root de Linux para listar los puertos relacionados con `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
```

El tipo de puerto `http_port_t` define los puertos en los que el Servidor HTTP Apache puede escuchar, que en este caso son los puertos TCP 80, 443, 488, 8008, 8009, y 8443. Si un administrador configura `httpd.conf` para que `httpd` escuche en el puerto 9876 (`Listen 9876`), pero la política no fue actualizada para reflejar esto, el comando `service httpd start` falla:

```
# /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]
```

Una negación de SELinux es similar a la siguiente y se guarda en `/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
```

Para permitir a `httpd` escuchar en un puerto que no está listado en el tipo de puerto `http_port_t`, ejecute el comando `semanage port` para agregar un puerto a la configuración de la política <sup>2</sup>:

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

La opción `-a` agrega un nuevo registro; la opción `-t` define un tipo; y la opción `-p` define un protocolo. El último argumento es el número de puerto a agregar.

---

<sup>2</sup> El comando `semanage port -a` agrega una entrada al archivo `/etc/selinux/targeted/modules/active/ports.local`. Nota: por defecto, este archivo sólo puede ser visto por el usuario root de Linux.

### 7.2.3. Evolucionando las Reglas y las Aplicaciones Rotas

Las aplicaciones se pueden romper, provocando que SELinux niegue el acceso. También, las reglas de SELinux evolucionan - SELinux no se debe ver como una aplicación que se ejecuta en una cierta forma, haciendo que deniegue el acceso, aún cuando la aplicación está funcionando como se espera que lo haga. Por ejemplo, si una nueva versión de PostgreSQL se lanza, puede realizar acciones sobre la política actual que no han sido vistas antes, haciendo que el acceso sea denegado, aún cuando el acceso debería ser permitido.

Para estas situaciones, después de que se niegue el acceso, use **audit2allow** para crear un módulo de política personalizado para permitir el acceso. Vaya a [Sección 7.3.8, "Permitiendo el Acceso: audit2allow"](#) para más información sobre el uso de **audit2allow**.

## 7.3. Corrección de Problemas

Las siguientes secciones ayudan a resolver problemas. Cubren los temas: chequeo de los permisos de Linux, que se chequean antes que las reglas de SELinux; posibles causas de negaciones de acceso de SELinux, pero no negaciones que se estén guardando; páginas man de los servicios, que contienen información sobre etiquetado y Booleanos; dominios permisivos, para permitir a un proceso correr en modo permisivo, en vez de todo el sistema; cómo buscar y encontrar mensajes; análisis de negaciones; y creación de módulos de políticas personalizados con **audit2allow**.

### 7.3.1. Permisos de Linux

Cuando el acceso se niega, verifique los permisos estándares de Linux. Como se mencionó en [Capítulo 2, Introducción](#), la mayoría de los sistemas operativos usan un sistema de Control de Acceso Discrecional (DAC) para controlar el acceso, permitiendo a los usuarios controlar los permisos de sus propios archivos. Las reglas de políticas de SELinux se verifican después de estas reglas DAC. Las reglas de políticas de SELinux no se usan si las reglas DAC niegan el acceso al principio.

Si el acceso es denegado y no hay negaciones SELinux guardadas, use el comando **ls -l** para ver los permisos estándares de Linux:

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

En este ejemplo, **index.html** pertenece al usuario y al grupo root. El usuario root tiene permisos de lectura y escritura (-rw), y los miembros del grupo root tienen permisos de lectura (-r-). Cualquier otro no tiene acceso (- - -). Por defecto, tales permisos no permiten a **httpd** leer este archivo. Para resolver esto, use el comando **chown** el dueño y el grupo. Este comando se debe ejecutar como usuario root de Linux:

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

Esto asume la configuración predeterminada, en la que **httpd** corre como usuario **apache** de Linux. Si corre **httpd** con un usuario diferente, reemplace **apache:apache** con ese usuario.

Vaya al borrador [Proyecto de Documentación de Fedora - "Permisos"](#)<sup>3</sup> para información sobre el manejo de los permisos de Linux.

<sup>3</sup> <http://fedoraproject.org/wiki/Docs/Drafts/AdministrationGuide/Permissions>

### 7.3.2. Posibles Causas de las Negaciones Silenciosas

En ciertas situaciones, las negaciones AVC pueden no ser guardadas cuando SELinux niega el acceso. Las aplicaciones y las funciones de las bibliotecas del sistema a menudo prueban más accesos que los pedidos para realizar sus tareas. Para mantener el menor privilegio sin llenar los informes de auditoría con negaciones AVC para pruebas sin peligro de las aplicaciones, la política puede silenciar las negaciones AVC sin permitir el uso de reglas `dontaudit`. Estas reglas son comunes en la política estándar. La contraparte de `dontaudit` es que, aunque SELinux niega el acceso, los mensajes no se guardan, lo que dificulta resolver el problema.

Deshabilite temporalmente las reglas `dontaudit`, permitiendo que se guarden todas las negaciones, ejecute el siguiente comando como usuario `root` de Linux:

```
/usr/sbin/semodule -DB
```

La opción `-D` deshabilita las reglas `dontaudit`; la opción `-B` reconstruye la política. Después de ejecutar **`semodule -DB`**, pruebe ejercitar la aplicación que tuvo problemas de permisos, y vea si ahora se guardan negaciones de SELinux relacionadas con la aplicación. Tenga cuidado con la decisión de qué negaciones se deben permitir, dado que algunas se deben ignorar y manejarse vía reglas `dontaudit`. Si tiene duda, o busca alguna guía, contacte a otros usuarios y desarrolladores de SELinux en una lista de SELinux, tal como [fedora-selinux-list](http://www.redhat.com/mailman/listinfo/fedora-selinux-list)<sup>4</sup>.

Para reconstruir la política y habilitar las reglas `dontaudit`, ejecute el siguiente comando como usuario `root` de Linux:

```
/usr/sbin/semodule -B
```

Esto restaura la política a su estado original. Para una lista completa de las reglas `dontaudit`, corra el comando **`sesearch --dontaudit`**. Búsquedas más refinadas usando la opción `-s dominio` y el comando **`grep`**. Por ejemplo:

```
$ 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 ;
```

Vaya a [Sección 7.3.6, “Mensajes Crudos de Auditoría”](#) y a [Sección 7.3.7, “Mensajes sealert”](#) para información acerca del análisis de las negaciones.

### 7.3.3. Páginas de Manual para Servicios

Las páginas de manual para los servicios conteniendo información valiosa, tal como qué tipo de archivo usar para una situación dada, y los Booleanos para cambiar el acceso que un servicio tiene (tal como `httpd` para acceder sistemas de archivos NFS). Esta información puede estar en la página de manual estándar o una página de manual con `selinux` como prefijo o sufijo.

Por ejemplo, la página de manual `httpd_selinux(8)` tiene información sobre qué tipo de archivo usar para una situación dada, así como los Booleanos para permitir scripts, compartir archivos, acceder directorios dentro de los directorios `home`, y así sucesivamente. Otras páginas de manual con información de SELinux para servicios incluyen a:

---

<sup>4</sup> <http://www.redhat.com/mailman/listinfo/fedora-selinux-list>

- Samba: la página de manual `samba_selinux(8)` describe que los archivos y directorios exportar vía Samba deben ser etiquetados con el tipo `samba_share_t`, así como los Booleanos para permitir archivos etiquetados con otros tipos distintos a `samba_share_t` para exportarlos vía Samba.
- NFS: la página de manual `nfs_selinux(8)` describe que, por defecto, los sistemas de archivo no se pueden exportar vía NFS, y que no se permite exportar sistemas de archivos. Los Booleanos como `nfs_export_all_ro` o `nfs_export_all_rw` deben activarse.
- Berkeley Internet Name Domain (BIND): la página de manual de `named(8)` describe qué tipo de archivo usar para una situación dada (vea la sección Perfil de Seguridad de SELinux para BIND de Red Hat section). La página man de `named_selinux(8)` describe que, por defecto, `named` no puede escribir a archivos de zona maestros y que, para permitir ese acceso, el Booleano `named_write_master_zones` debe ser puesto en 1.

La información en las páginas del manual le ayudan a configurar los tipos de archivos correctos y los Booleanos, ayudándolo a prevenir las negaciones de acceso por parte de SELinux.

### 7.3.4. Dominios Permisivos

Cuando SELinux se ejecuta en modo permisivo, SELinux no niega el acceso, sino que las negaciones para las acciones se guardan como si fuera que corre en modo obediente. Previamente, no era posible hacer permisivo un único dominio (recuerde: los procesos corren en dominios). En ciertas situaciones, esto llevó a hacer el sistema permisivo para poder corregir los problemas.

Fedora 11 introduce los dominios permisivos, donde un administrador puede configurar un único proceso (dominio) para que corra permisivo, en vez de hacer todo el sistema permisivo. Los chequeos de SELinux se realizan igualmente para dominios permisivos; sin embargo, el kernel permite el acceso e informa la negación AVC para situaciones donde SELinux hubiera negado el acceso. Los dominios permisivos están también disponibles en Fedora 9 (con las últimas actualizaciones aplicadas).

En el Linux para Empresas de Red Hat 4 y 5, los Booleanos `dominio_disable_trans` están disponibles para prevenir que una aplicación transicione a un dominio confinado, y por lo tanto, el proceso se ejecute en un dominio no confinado, tal como `initrc_t`. Poniendo en 1 tales booleanos pueden causar problemas serios. Por ejemplo, si el Booleano `httpd_disable_trans` se pone en 1:

- `httpd` corre en el dominio no confinado `initrc_t`. Los archivos creados por los procesos en el dominio `initrc_t` puede no tener aplicadas las mismas reglas de etiquetados como los archivos creados por el proceso corriendo en el dominio `httpd_t`, permitiendo que los procesos puedan potencialmente crear archivos mal etiquetados. Esto causa problemas más adelante.
- dominios confinados que pueden comunicarse con `httpd_t` no pueden comunicarse con `initrc_t`, posiblemente causan fallas adicionales.

Los Booleanos `domain_disable_trans` fueron eliminados de Fedora 7, y no se pusieron reemplazos. Los dominios permisivos pueden resolver esos problemas: se aplican las reglas de transición y los archivos se crean con las etiquetas correctas.

Los dominios permisivos se pueden usar para:

- hacer que un único proceso (dominio) corra permisivo para solucionar alguna cuestión, en vez de poner todo el sistema en riesgo haciendo permisivo a todo el sistema.
- creación de políticas para nuevas aplicaciones. Previamente, era recomendado crear una política mínima, y luego poner la máquina completa en modo permisivo, para que la aplicación pudiera



funcionar, pero las negaciones de SELinux eran igualmente grabadas. **audit2allow** podría usarse luego para ayudar a escribir la política. Esto pone todo el sistema en riesgo. Con dominios permisivos, sólo el dominio en la nueva política puede marcarse como permisivo, sin poner en riesgo todo el sistema.

### 7.3.4.1. Creando un Dominio Permisivo

Para hacer un dominio permisivo, ejecute el comando **semanage permissive -a dominio**, donde *dominio* es el dominio que quiere hacer permisivo. Por ejemplo, ejecute el siguiente comando como usuario root de Linux para hacer permisivo el dominio `httpd_t` (el dominio en el que corre el Servidor HTTP Apache):

```
/usr/sbin/semanage permissive -a httpd_t
```

Para ver una lista de los dominios que hizo permisivos, corra el comando **semodule -l | grep permissive** como usuario root de Linux. Por ejemplo:

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

Si ya no quiere que un dominio sea permisivo, corra el comando **semanage permissive -d dominio** como usuario root de Linux. Por ejemplo:

```
/usr/sbin/semanage permissive -d httpd_t
```

### 7.3.4.2. Negaciones para Dominios Permisivos

El mensaje SYSCALL es diferente para dominios permisivos. El siguiente es un ejemplo de una negación de AVC (y la llamada a sistema asociada) desde el Servidor HTTP Apache:

```
type=AVC msg=audit(1226882736.442:86): avc: denied { getattr }
for pid=2427 comm="httpd" path="/var/www/html/archivo1"
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)
```

Por defecto, el dominio `httpd_t` es no permisivo, y como tal, la acción es negada, y el mensaje SYSCALL contiene `success=no`. El siguiente es un ejemplo de negación AVC para la misma situación, excepto que el comando **semanage permissive -a httpd\_t** se ejecutó para hacer el dominio `httpd_t` permisivo:

```
type=AVC msg=audit(1226882925.714:136): avc: denied
{ read } for pid=2512 comm="httpd" name="archivo1" 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)
```

En este caso, aunque la negación AVC fue grabada, el acceso no fue negado, como se muestra en el mensaje SYSCALL success=yes.

Vaya al blog de Dan Walsh "[Dominios Permisivos](#)"<sup>5</sup> para más información sobre dominios permisivos.

### 7.3.5. Búsqueda y Revisión de Negaciones

Esta sección asume que los paquetes *setroubleshoot*, *setroubleshoot-server*, y *audit* están instalados, y que los demonios *auditd*, *rsyslogd* y *setroubleshootd* están ejecutándose. Vaya a [Sección 5.2, "Qué Archivo Log se usa"](#) para información sobre cómo iniciar estos demonios. Hay disponibles un número de herramientas para ver las negaciones de SELinux, tales como **ausearch**, **aureport** y **sealert**.

#### ausearch

El paquete *audit* provee **ausearch**. De la página de manual de *ausearch(8)*: "**ausearch** es una herramienta que puede consultar los registros del demonio *audit* basados en distintos criterios de búsqueda"<sup>6</sup>. La herramienta **ausearch** accede a `/var/log/audit/audit.log`, y como tal, se debe ejecutar como usuario `root` de Linux:

Buscando	Comando
todas las negaciones	<code>/sbin/ausearch -m avc</code>
negaciones de hoy	<code>/sbin/ausearch -m avc -ts today</code>
negaciones desde los últimos 10 minutos	<code>/sbin/ausearch -m avc -ts recent</code>

Para buscar negaciones de SELinux para un servicio particular, use la opción `-c nombre-de-comando`, donde *nombre-de-comando* "es el nombre del ejecutable"<sup>7</sup>, por ejemplo, `httpd` para el Servidor HTTP Apache, y `smbd` para Samba:

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

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

Vaya a la página de manual de *ausearch(8)* para más opciones de **ausearch**.

#### aureport

El paquete *audit* provee **aureport**. De la página de manual de *aureport(8)*: "**aureport** es una herramienta que produce informes resumen de los registros de auditoría del sistema"<sup>8</sup>. La herramienta **aureport** accede a `/var/log/audit/audit.log`, y como tal, se debe ejecutar como

<sup>5</sup> <http://danwalsh.livejournal.com/24537.html>

<sup>6</sup> De la página man de *ausearch(8)*, como se incluyó en el paquete *audit* en Fedora 11.

<sup>7</sup> De la página de manual de *ausearch(8)*, como se incluyó en el paquete *audit* en Fedora 11.

<sup>8</sup> de la página de manual de *aureport(8)* como se incluyó en el paquete *audit* en Fedora 11.

usuario root de Linux. Para ver una lista de las negaciones de SELinux y cuán a menudo ocurren, ejecute el comando **aureport -a**. El siguiente es un ejemplo de la salida en donde se incluyen dos negaciones:

```
# /sbin/aureport -a

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
```

Vaya a la página de manual de [aureport\(8\)](#) para más opciones de **aureport**.

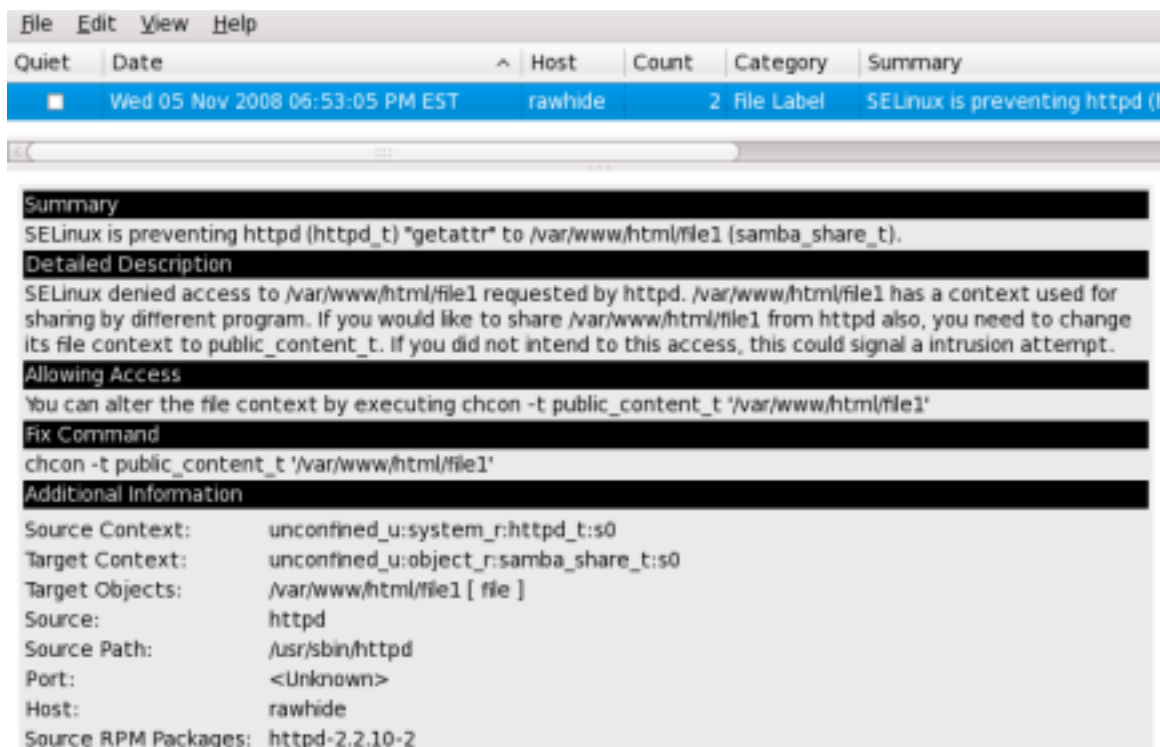
### sealert

El paquete *setroubleshoot-server* provee **sealert**, que lee los mensajes de negación traducidos por *setroubleshoot-server*. A las negaciones se le asignan IDs, como se ve en **/var/log/messages**. El siguiente es un ejemplo de negación en **messages**:

```
setroubleshoot: SELinux esta previniendo acerca de la intención de
  httpd (httpd_t) "getattr" to /var/www/html/archivo1 (samba_share_t).
  Para mensajes SELinux completos, ejecute sealert -l 84e0b04d-
  d0ad-4347-8317-22e74f6cd020
```

En este ejemplo, el ID de negación es 84e0b04d-d0ad-4347-8317-22e74f6cd020. La opción **-l** toma un ID como argumento. Ejecutando el comando **sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020** le presenta un análisis detallado de por qué SELinux negó el acceso, y una posible solución para permitir el acceso.

Si está corriendo el Sistema de Ventanas X, tenga los paquetes *setroubleshoot* y *setroubleshoot-server* instalados, y los demonios *setroubleshootd* y *auditd* ejecutándose, una estrella amarilla y una advertencia se muestran cuando un acceso es negado por SELinux. Con clic sobre la estrella se lanza **sealert** modo gráfico y se muestra la negación como HTML:



- Ejecute el comando **sealert -b** para lanzar la GUI de **sealert**.
- Ejecute el comando **sealert -l \\*** para ver un análisis detallado de todas las negaciones.
- Como usuario root de Linux, ejecute el comando **sealert -a /var/log/audit/audit.log -H > audit.html** para crear una versión HTML del análisis de **sealert**, como si se viera en la GUI de **sealert**.

Vaya a la página man de `sealert(8)` para más opciones de **sealert**.

### 7.3.6. Mensajes Crudos de Auditoría

Los mensajes crudos de auditoría se guardan en `/var/log/audit/audit.log`. El siguiente es un ejemplo de negación AVC (y su llamada a sistema asociado) que ocurrió cuando el Servidor HTTP Apache (corriendo en el dominio `httpd_t`) intentó acceder el `/var/www/html/archivo1` (etiquetado con el tipo `samba_share_t`):

```
type=AVC msg=audit(1226874073.147:96): avc: denied { getattr }
for pid=2465 comm="httpd" path="/var/www/html/archivo1"
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 }
```

El ítem entre llaves indica el permiso que fue negado. `getattr` indica el proceso fuente que intentó leer la información de estado del archivo destino. Esto ocurre antes de leer archivos. Esta acción es negada dado que el archivo está siendo accedido con la etiqueta equivocada. Los permisos vistos comúnmente incluyen a `getattr`, `read` y `write`.

```
comm="httpd"
```

El ejecutable que lanzó el proceso. La dirección completa del ejecutable se encuentra en la sección `exe=` del mensaje de la llamada al sistema (SYSCALL), que es en este caso `exe="/usr/sbin/httpd"`.

```
path="/var/www/html/archivo1"
```

La dirección al objeto (destino) al que quiere acceder el proceso.

```
scontext="unconfined_u:system_r:httpd_t:s0"
```

El contexto de SELinux del proceso que intentó la acción denegada. En este caso, es el contexto SELinux del Servidor HTTP Apache, que `core` en el dominio `httpd_t`.

```
tcontext="unconfined_u:object_r:samba_share_t:s0"
```

El contexto de SELinux del objeto (destino) al que intentó acceder el proceso. En este caso, es el contexto SELinux del **archivo1**. Nota: el tipo `samba_share_t` no es accesible para procesos que corren el dominio `httpd_t`.

En ciertas situaciones, el `tcontext` puede coincidir con `scontext`, por ejemplo, cuando un proceso intenta ejecutar un servicio del sistema que cambiará las características de ese proceso en ejecución, tales como el ID del usuario. También el `tcontext` puede coincidir con el `scontext` cuando un proceso intenta usar más recursos (como la memoria) más allá de los límites normales permitidos, lo que resulta en un chequeo de seguridad para ver si el proceso tiene permitido romper esos límites.

Desde el mensaje de llamado al sistema (SYSCALL) nos interesan dos ítems:

- `success=no`: indica si la negación (AVC) fue aplicada o no. `success=no` indica que la llamada al sistema no fue exitosa (SELinux negó el acceso). `success=yes` indica que la llamada al sistema fue exitosa - esto se puede ver en dominios permisivos o en dominios no confinados, tales como `initrc_t` y `kernel_t`.
- `exe="/usr/sbin/httpd"`: la dirección completa al ejecutable que inició el proceso, que es en este caso `exe="/usr/sbin/httpd"`.

Un tipo de archivo incorrecto es una causa común de que SELinux niegue el acceso. Para comenzar a resolver el problema, compare el contexto fuente (`scontext`) con el contexto destino (`tcontext`). ¿Debería el proceso (`scontext`) acceder a un objeto (`tcontext`)? Por ejemplo, el Servidor HTTP Apache (`httpd_t`) sólo debería acceder tipos especificados en la página `man httpd_selinux(8)`, tales como `httpd_sys_content_t`, `public_content_t`, y así sucesivamente, a menos que se configure de otra forma.

### 7.3.7. Mensajes `sealert`

Las negaciones tienen IDs asignados, como se ve en `/var/log/messages`. El siguiente es un ejemplo de negación AVC (guardado en `messages`) que ocurrió cuando el Servidor HTTP Apache (corriendo en el dominio `httpd_t` domain) intentó acceder el `/var/www/html/archivo1` (etiquetado con el tipo `samba_share_t`):

```
hostname setroubleshoot: SELinux está previniendo acerca de la intención de
  httpd (httpd_t) de "getattr" sobre /var/www/html/archivo1 (samba_share_t).
  Para los mensajes completos de SELinux, ejecute sealert -l 84e0b04d-
d0ad-4347-8317-22e74f6cd020
```

Como se sugirió, ejecute el comando **sealert -l 84e0b04d-d0ad-4347-8317-22e74f6cd020** para ver el mensaje completo. Este comando sólo funciona en la máquina local, y presenta la misma información que la interfase gráfica de **sealert**:

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

Summary:

SELinux is preventing httpd (httpd\_t) "getattr" to /var/www/html/archivo (samba\_share\_t).

Detailed Description:

SELinux denied access to /var/www/html/archivo1 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/archivo1'
```

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/archivo1 [ 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.fc11
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.fc11.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/archivo1"
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)
```

### Resumen

Un breve resumen de la acción negada. Esto es lo mismo que la negación en `/var/log/messages`. En este ejemplo, al proceso `httpd` se le negó el acceso al (**archivo1**), que está etiquetado con el tipo `samba_share_t`.

### Descripción Detallada

Una descripción más detallada. En este ejemplo **archivo1** está etiquetado con el tipo `samba_share_t`. Este tipo se usa para archivos y directorios que quiere exportar vía Samba. La descripción sugiere cambiar el tipo a un tipo que pueda ser accedido por Samba y por el Servidor HTTP Apache, si tal acceso es deseado.

### Permitiendo Acceso

Una sugerencia sobre cómo permitir el acceso. Esto puede hacerse reetiquetando archivos, poniendo en 1 un Booleano, o creando un módulo de política local. En este caso, la sugerencia es etiquetar el archivo con un tipo accesible por el Servidor HTTP Apache y por Samba.

### Comando para Corregir

Un comando sugerido para permitir el acceso y resolver la negación. En este ejemplo, se dá el comando para cambiar el tipo del **archivo1** a `public_content_t`, que es accesible por el Servidor HTTP Apache y por Samba.

### Información Adicional

Información que es útil en los informes de errores, como el nombre de paquete de política y la versión (`selinux-policy-3.5.13-11.fc11`), pero que no puede ser de ayuda para resolver por qué ocurrió la negación.

## Mensajes Crudos de Auditoría

Los mensajes crudos de auditoría de `/var/log/audit/audit.log` que son asociados con la negación. Vaya a [Sección 7.3.6, “Mensajes Crudos de Auditoría”](#) para información acerca de cada ítem en la negación AVC.

### 7.3.8. Permitiendo el Acceso: audit2allow

No use el ejemplo en esta sección en producción. Se usa sólo para mostrar el uso de **audit2allow**.

De la página de manual de audit2allow(1): “**audit2allow** - genera reglas para habilitar de la política de SELinux a partir de operaciones negadas”<sup>9</sup>. Después de analizar las negaciones como en [Sección 7.3.7, “Mensajes sealert”](#), y si ningún cambio de etiqueta o de Booleano permitió el acceso, use **audit2allow** para crear un módulo de política local. Después de que el acceso es negado por SELinux, ejecutando el comando **audit2allow** presenta reglas de Obligación de Tipo que permitirían el acceso previamente negado.

El siguiente ejemplo muestra el uso de **audit2allow** para crear un módulo de política:

1. Una negación y la llamada al sistema asociado se graban en `/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)
```

En este ejemplo, a **certwatch** (comm="certwatch") se le negó el acceso de escritura (`{ write }`) a un directorio etiquetado con el tipo `var_t` (tcontext=system\_u:object\_r:var\_t:s0). Analice la negación como en [Sección 7.3.7, “Mensajes sealert”](#). Si no hay cambios de etiquetas o Booleanos que permitan el acceso, use **audit2allow** para crear un módulo de política local.

2. Con una negación grabada, tal como la negación de certwatch en el paso 1, corra el comando **audit2allow -w -a** para producir una descripción legible al humano sobre por qué el acceso fue negado. La opción `-a` hace que se lean todos los registros de auditoría. La opción `-w` produce una descripción legible al humano. La herramienta **audit2allow** accede a `/var/log/audit/audit.log`, y como tal, debe ser ejecutada como usuario root de Linux:

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

<sup>9</sup> De la página man de audit2allow(1), tal como fue incluida en el paquete *policycoreutils* en Fedora 11.

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

Como se muestra, el acceso fue negado debido a que falta una regla de Obligación de Tipo.

3. Ejecute el comando `audit2allow -a` para ver la regla de Obligación de Tipo que permite el acceso negado:

```
# audit2allow -a

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



### Importante

La falta de reglas de Ejecución de Tipos son usualmente causados por errores en la política de SELinux, y deben ser informadas en el *Bugzilla de Red Hat*<sup>10</sup>. Para Fedora, crear informes sobre el producto Fedora, y seleccione el componente `selinux-policy`. Incluya la salida de los comandos `audit2allow -w -a` y `audit2allow -a` en el informe del error.

4. Para usar la regla mostrada por `audit2allow -a`, ejecute el comando `audit2allow -a -M mycertwatch` como usuario root de Linux. La opción `-M` crea un archivo de Obligación de Tipo (`.te`) con el nombre especificado en `-M`, en su directorio actual de trabajo:

```
# audit2allow -a -M mycertwatch

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

semodule -i mycertwatch.pp

# ls
mycertwatch.pp  mycertwatch.te
```

También, `audit2allow` compila la regla de Obediencia de Tipo en un paquete de política (`.pp`). Para instalar el módulo, ejecute el comando `/usr/sbin/semodule -i mycertwatch.pp` como usuario root de Linux.



### Importante

Los módulos creados con `audit2allow` pueden permitir más acceso que el requerido. Se recomienda que la política creada con `audit2allow` sea enviada a una lista de SELinux, tal como *fedora-selinux-list*<sup>11</sup>, para su revisión. Si cree que hay un error en la política, informe un error en *Bugzilla de Red Hat*<sup>12</sup>.



Si tiene múltiples negaciones de múltiples procesos, pero solo quiere crear una política personalizada para un proceso único, use el comando **grep** para una búsqueda más refinada de **audit2allow**. El siguiente ejemplo muestra el uso de **grep** para sólo enviar negaciones de **certwatch** a **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
```

Vaya a la entrada blog "[Uso de audit2allow para construir módulos de políticas. Revisado.](#)"<sup>13</sup> de Dan Walsh para información adicional sobre el uso de **audit2allow** para construir módulos de políticas.

---

<sup>13</sup> <http://danwalsh.livejournal.com/24750.html>



---

# Información Adicional

## 8.1. Contributors

- [Geert Warrink](#)<sup>1</sup> (translation - Dutch)
- [Domingo Becker](#)<sup>2</sup> (translation - Spanish)
- [Daniel Cabrera](#)<sup>3</sup> (translation - Spanish)

## 8.2. Other Resources

### La Agencia de Seguridad Nacional (NSA)

De la página [Contribuyentes de SELinux](#)<sup>4</sup>:

*Los investigadores del laboratorio de Investigación en Seguridad de la Información Nacional (NIARL) de la NSA diseñaron e implementaron controles de acceso obligatorios flexibles en los subsistemas principales del kernel de Linux e implementaron nuevos componentes del sistema operativo provistos por la arquitectura Flask, llamado servidor de seguridad y caché vector de acceso. Los investigadores de la NSA rediseñaron el SELinux basado en LSM para incluirlo en Linux 2.6. NSA también lideró el desarrollo de controles similares para el Sistema de Ventanas X (XACE/XSELinux) y para Xen (XSM/Flask).*

- El sitio web de SELinux principal: <http://www.nsa.gov/research/selinux/index.shtml>.
- Documentación de SELinux: <http://www.nsa.gov/research/selinux/docs.shtml>.
- Trasfondo de SELinux: <http://www.nsa.gov/research/selinux/background.shtml>.

### Tecnología de Tresys

[Tresys Technology](#)<sup>5</sup> son los desarrolladores de:

- [Herramientas y bibliotecas en el espacio del usuario para SELinux](#)<sup>6</sup>.
- [Política de Referencia de SELinux](#)<sup>7</sup>.

### Noticias de SELinux

- Noticias: <http://selinuxnews.org/wp/>.
- Planeta SELinux (blogs): <http://selinuxnews.org/planet/>.

### Wiki del Proyecto SELinux

- Página principal: [http://selinuxproject.org/page/Main\\_Page](http://selinuxproject.org/page/Main_Page).
- Recursos del usuario, incluyendo enlaces a documentación, listas de correo, sitios web y herramientas: [http://selinuxproject.org/page/User\\_Resources](http://selinuxproject.org/page/User_Resources).

---

<sup>4</sup> <http://www.nsa.gov/research/selinux/contrib.shtml>

<sup>5</sup> <http://www.tresys.com/>

### Linux para Empresas de Red Hat

- La *Guía de Despliegue del Linux para Empresas de Red Hat*<sup>8</sup> contiene una sección de *Referencias*<sup>9</sup> SELinux, que tiene enlaces a tutoriales de SELinux, información general y la tecnología detrás de SELinux.
- La *Guía de SELinux del Linux para Empresas de Red Hat 4*<sup>10</sup>.

### Fedora

- Página principal: <http://fedoraproject.org/wiki/SELinux>.
- Resolución de problemas: <http://fedoraproject.org/wiki/SELinux/Troubleshooting>.
- FAQ de SELinux para Fedora Core 5: <http://docs.fedoraproject.org/selinux-faq-fc5/>.

### Las Preguntas Frecuentes No Oficiales de SELinux

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

### IRC

En *Freenode*<sup>11</sup>:

- #selinux
- #fedora-selinux

---

<sup>11</sup> <http://freenode.net/>

---

# Apéndice A. Revision History

Revisión 1.3 Tue May 12 2009

Scott Radvan [sradvan@redhat.com](mailto:sradvan@redhat.com)

Revisión para Fedora 11

Revisión 1.2 Mon Jan 19 2009

Murray McAllister [mmcallis@redhat.com](mailto:mmcallis@redhat.com)

Actualización de los enlaces a sitios web de la NSA

Revisión 1.1 Sat Dec 6 2008

Murray McAllister [mmcallis@redhat.com](mailto:mmcallis@redhat.com)

Resolución del [Bugzilla de Red Hat #472986](#), "[httpd no escribe en /etc/httpd/logs](#)"<sup>1</sup>

Se agregó la sección nueva, "6.6. Booleanos para que los Usuarios Ejecuten Aplicaciones"

Revisiones menores del texto

Revisión 1.0 Tue Nov 25 2008

Murray McAllister [mmcallis@redhat.com](mailto:mmcallis@redhat.com)

El contenido inicial se guarda en <http://docs.fedoraproject.org/>

