327.2 Mandatory Access Control

Weight: 4

Description: Candidates should be familiar with Mandatory Access Control systems for Linux. Specifically, candidates should have a thorough knowledge of SELinux. Also, candidates should be aware of other Mandatory Access Control systems for Linux. This includes major features of these systems but not configuration and use.

Key Knowledge Areas:

- Understand the concepts of TE, RBAC, MAC and DAC
- Configure, manage and use SELinux
- Be aware of AppArmor and Smack

Terms and Utilities:

- getenforce, setenforce, selinuxenabled
- getsebool, setsebool, togglesebool
- fixfiles, restorecon, setfiles
- newrole, runcon
- semanage
- sestatus, seinfo
- apol
- seaudit, seaudit-report, audit2why, audit2allow
- /etc/selinux/*

Conceptos Previos

Security-Enhanced Linux o SELinux, es una arquitectura de seguridad integrada a partir del kernel 2.6.x usando los módulos de seguridad linux (o linux security modules, **LSM**). Este es un proyecto de la Agencia de Seguridad Nacional (NSA) de los Estados Unidos y de la comunidad SELinux. La integración de SELinux en Red Hat Enterprise Linux fue un esfuerzo conjunto entre la NSA y Red Hat.

SELinux proporciona un sistema flexible de control de acceso obligatorio (Mandatory Access Control→ **MAC**) incorporado en el kernel. Bajo el Linux estándar se utiliza el control de acceso a discreción (Discretionary Access Control→ **DAC**), en el que un proceso o aplicación ejecutándose como un usuario (UID o SUID) tiene los permisos de ese usuario en los objetos, archivos, sockets y otros procesos. Al ejecutar un kernel SELinux, MAC protege al sistema de aplicaciones maliciosas o dañadas que pueden perjudicar o destruir el sistema. SELinux define el acceso y los derechos de transición de cada usuario, aplicación, proceso y archivo en el sistema. SELinux gobierna la interacción de estos sujetos y objetos usando una política de seguridad que especifica cuán estricta o indulgente debería ser una instalación de GNU/Linux.

En su mayor parte, SELinux es casi invisible para la mayoría de los usuarios. Solamente los administradores de sistemas se deben preocupar sobre lo estricto que debe ser una política a implementar en sus entorno de servidores. La política puede ser tan estricta o tan indulgente como se requiera, y es bastante detallada. Este detalle le dá al kernel SELinux un control total y granular sobre el sistema completo.

Cuando un sujeto, tal como una aplicación, intenta acceder a un objeto tal como un archivo, el servidor de aplicación de políticas verifica un caché de vector de acceso (**AVC**), donde se registran los permisos de objeto y del sujeto. Si no se puede tomar una decisión basado en los datos en el AVC, la petición continua al servidor de seguridad, el cual busca el contexto de securidad de la aplicación y del archivo en una matriz. Los permisos son entonces otorgados o negados, con un mensaje de avc: denied detallado en: /var/log/messages .

Los sujetos y objetos reciben su contexto de seguridad a partir de la política instalada, que también proporciona información para llenar la matriz de seguridad del servidor.

Además de ejecutarse en un modo impositivo u obligatorio, SELinux puede ejecutarse en un modo permisivo, donde el **AVC** es verificado y se registran los rechazos, pero SELinux no hace cumplir esta política.

Beneficios de utilizar SELinux !!

Todos los procesos y archivos están etiquetados con un tipo. Un tipo define un dominio para los procesos y un tipo para los archivos.

Los procesos se separan entre sí, ejecutándose en sus propios dominios, y las reglas de la política SELinux definen cómo los procesos interactúan con los archivos, así como entre si.

El acceso sólo se permite si existe una regla de política SELinux que lo permita específicamente.

El control de acceso es de mayor granularidad. Los permisos UNIX tradicionales se controlan de forma discrecional y se basan en el ID de usuario y grupo. Las decisiones de acceso de SELinux se basan en toda la información disponible, ademas de un usuario SELinux, una función, un tipo y, opcionalmente, un nivel y categoría.

La política de SELinux se define administrativamente, se aplica en todo el sistema y no se establece a discreción del usuario.

Reducción de la vulnerabilidad para privilegiar ataques de gran escala. Los procesos se ejecutan en dominios y están por lo tanto separados entre sí.

Las reglas de la política SELinux definen cómo los procesos acceden a los archivos y otros procesos.

Si un proceso se ve comprometido, el atacante sólo tiene acceso a las funciones de ese proceso.

Por ejemplo, si el Servidor Apache HTTP está comprometido, un atacante no puede usar ese proceso para leer archivos en los directorios de inicio del usuario, a menos que se haya agregado o configurado una regla de política SELinux específica.

SELinux puede utilizarse para reforzar la confidencialidad e integridad de los datos, así como para proteger los procesos de entradas no confiables.

↘ SELinux 'NO' es:

→ Un software antivirus.

→ Un sistema de reemplazo de contraseñas, firewalls u otros sistemas de seguridad.

→ Una Solución de Seguridad Todo en uno.

Entendiendo como funciona SELinux

SELinux funciona como un modulo en el kernel, logrando un mayor nivel de abstracción para los usuarios.

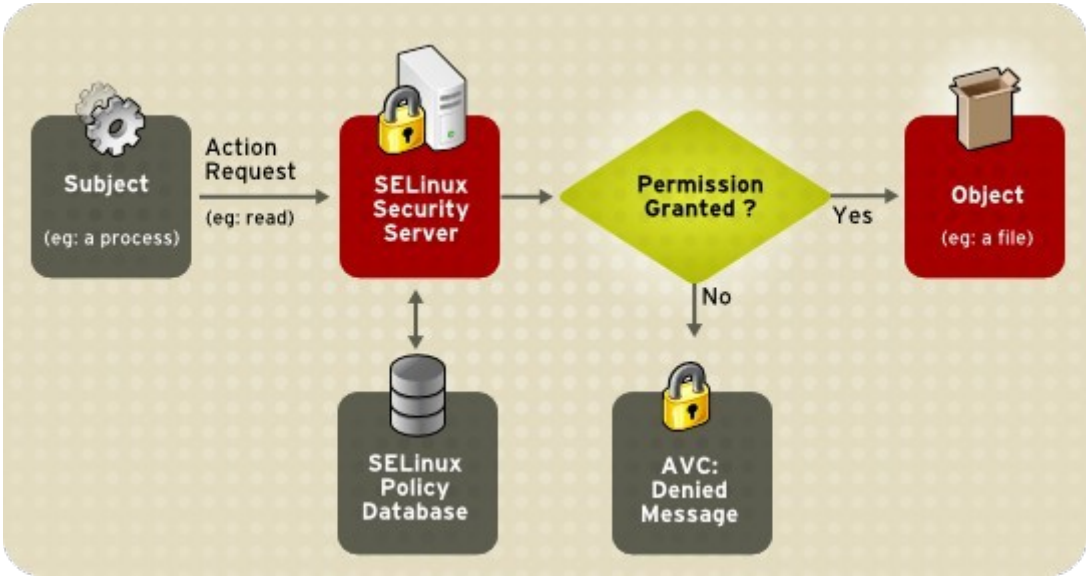
SELinux trabaja con contextos de seguridad, controles de acceso impositivos u obligatorios y control de acceso en base a roles; ofreciendo un control mas granular del acceso a los recursos del sistema por parte de los objetos (programas y aplicaciones) y los sujetos (roles, usuarios y grupos).

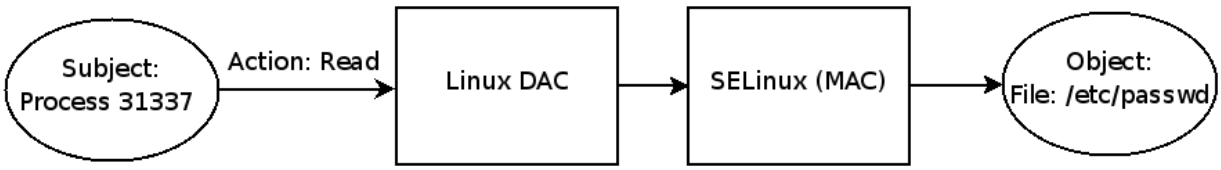
No se reemplaza el modelo tradicional se seguridad de los sistemas tipo Unix; por el contrario, sirve de complemento de este, en los puntos que la seguridad tradicional no es suficiente, en el que la seguridad esta dividida por niveles de usuarios, grupos, derechos de accesos, listas de control de acceso y atributos extendidos de acceso, y en donde un usuario puede ejecutar un conjunto de aplicaciones a las que tiene derecho y ejecutadas con los niveles de acceso que posee el usuario. Este tipo de seguridad es llamada **DAC** (Discretionary Access Control), control de acceso discrecional.

La NSA introdujo un sistema de control tipo MAC (Mandatory Access Control), basado en contextos donde se indica cuando un objeto o sujeto puede acceder a otro objeto.

En este caso, el administrador debe definir los derechos de cada usuario que acceda a algunas aplicaciones que accedan a cualquier objeto del sistema. Para evitar que esta operacion sea tediosa, se definen controles de acceso por roles (Role-Based Access Control→ **RBAC**).

Esquema de funcionamiento





Los métodos de seguridad de SELinux no reemplazan los tradicionales controles de acceso discrecionales, por el contrario, los complementa para brindar un control completo y granular de la seguridad. Todo inicialmente es consultado con la capa de control de acceso discrecional (**DAC**), si esta niega, no hay necesidad de consultar la capa de control de acceso obligatorio (**MAC**); en caso contrario si el **DAC** permite el acceso, entonces se hace la consulta hacia la capa **MAC** para terminar de verificar los controles de acceso.

Términos Clave

↘ **Sujeto**: Cualquier usuario o proceso que accede a un objeto.

↘ **Objeto**: Es un recurso tal como: fichero, directorio, dispositivo de hardware, interfaz de red, puerto, pipe o socket al que accede un Sujeto.

↘ **Acceso**: Acción realizada por un Sujeto en un Objeto. Crear, leer, actualizar un fichero, acceder a un puerto o socket, crear o acceder a un directorio, ...

↘ **Política**: Conjunto de reglas que refuerzan un sistema. Como puede acceder un sujeto a un objeto o interactuar con otros sujetos u objetos. En ausencia de reglas el comportamiento por defecto es de Denegación. Existen dos políticas estandar en RHEL: *targeted* y *strict*, siendo targeted la política por defecto. Un sujeto *unconfined* dentro de un dominio es mas vulnerable que si esta *confined* por ejemplo.

↘ **Contexto**: Se utiliza para almacenar atributos de seguridad de los Sujetos y Objetos. Esta relacionado con el Labeling o Etiquetado. SELinux lo utiliza para tomar decisiones de control de acceso. Cada Sujeto y Objeto tienen un contexto asignado el cual consiste en un SELinux Usuario, Rol, Tipo o Dominio y opcionalmente un nivel de sensibilidad.

↘ **Etiquetado**: (Labeling). Mapeo entre los ficheros y el stma. de ficheros dentro de un contexto.

↘ **SELinux User**: Existen usuarios SELinux predefinidos con unos roles predefinidos →user_u.

↘ **Rol**: Atributo basado en RBAC (Role Based Access Control) . Define que sujetos que sujetos pueden acceder a diferentes tipos o dominios.

↘ **Tipo y Dominio**: Atributo del tipo de enforzamiento. Es un grupo de objetos con requerimientos de seguridad uniforme para archivos o procesos. Ejemplo: user_home_dir_t .

↘ **TE**: (Tipo de Enforzamiento). Identifica y limita a un Sujeto la capacidad de acceder a dominios y tipos a los procesos y ficheros.

↘ **Level**: Es un atributo de MLS y de MCS. Configuran una pareja ***sensitivity:category***, valores que definen el nivel de seguridad en el contexto. **MCS** solo soporta sensitivity 0→ s0-s0, y 1024 categorys c0.c1023 => Expresado de la forma: s0-s0:c0.c1023. **MLS** soporta varios niveles de sensitivity y 1024 categorys => Expresado por ejemplo de la forma: s0-s7:c0.c1023. MLS permite mayor granularidad que MCS.

Atributos de seguridad

SELinux usa una combinacion de un modelo de identidad, control de acceso en base a roles (RBAC), tipo de enforzamiento (**TE**) y categorias de acceso a la informacion. SELinux RBAC autoriza a cada usuario (de SELinux) para un conjunto de roles. Cada rol es autorizado para un conjunto de tipos.

Esto se logra utilizando cuatro atributos de seguridad:

1. **Identidad de usuario**: SELinux tiene su propia base de datos de usuarios que esta asociada a la base de datos normal de usuarios de Linux. Las identidades son usadas en ambos, sujetos y objetos. Solamente unos cuantos usuarios de SELinux son definidos. Pueden ser listados con el comando '**semanage user -l**'

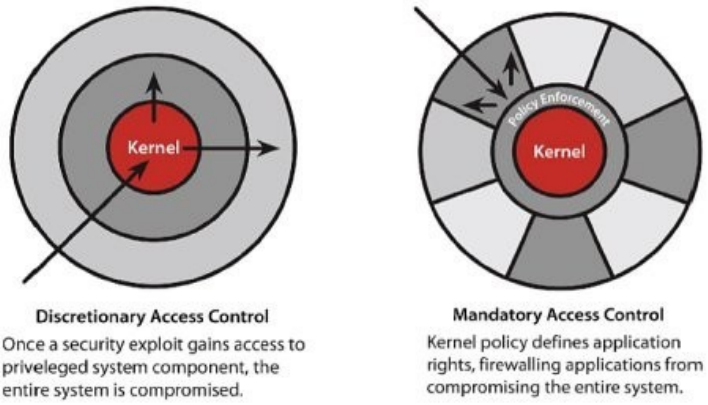
- **user_u** – Usuarios normales.
- **system_u** – Procesos iniciados (al arranque).
- **root** – Administrador.

2. **Role**: Los usuarios pueden entrar en diferentes roles. Diferentes roles pueden entrar en diferentes dominios. Para objetos (archivos), esto es siempre object_r.

3. **Tipo / dominio**: Es el atributo “principal” en SELinux. Tambien llamado el “atributo primario”. Por lo general son solo unos pocos usuarios/roles, pero centenares de tipos. No existe diferencia entre “tipos” y “dominios”, pero los "dominios" son usados cuando hablamos de procesos y los “tipos” cuando hablamos de archivos. Cada proceso esta confinado en su propia caja de arena, (sandbox) con acceso restringido, también llamado “Tipo de Enforzamiento”→ **TE**”.

4. **Sensitive Level/Categoria** : Podra establecer la categoria / nivel, para habilitar los controles de seguridad Multi-nivel (**MLS**) o seguridad Multi-categoria (**MCS**). Estos cuatro atributos de seguridad, constituyen lo que se denomina un “contexto de seguridad”:

usuario	**_u**	**usuario_u**
rol	**_r**	**object_r**
tipo/dominio	**_t**	**unconfined_t**
sensitive level/categoria		**s0-s0:c0.c1023**



Cuando tenemos todos los servicios de red y sistema confinados, tenemos un mayor control de seguridad, porque no comprometemos al sistema, aislando cada servicio como una entidad independiente.

Escalamiento de Privilegios → roles

La seguridad esta segmentada en **roles principales:**

user_r: El mínimo privilegio, asignado simplemente a los usuarios. No puede hacer escalamiento de privilegios ni alternar de rol.

staff_r: Similar al rol user_r, con la diferencia de que con este rol se puede alternar a los diferentes roles asignados con el perfil del usuario.

auditadm_r: Rol encargado de las herramientas de auditoria de manera única y exclusiva,

los usuarios de este rol solamente desempeñan esta función.

secadm_r: Rol encargado de manejar las herramientas de seguridad, declarar nuevos roles,

crear nuevas políticas y ver los log de auditoria, pero no puede modificarlos.

sysadm_r: Rol maximo en el sistema, pero no puede usar las herramientas de auditoria ni modificar estos reportes.

El sistema de archivos virtual → /sys/fs/selinux

El pseudo-sistema de archivos **/sys/fs/selinux** contiene los comandos que son utilizados más a menudo por el kernel. Este tipo de sistema de archivos es similar al pseudo sistema /proc/.

El ejemplo siguiente presenta contenidos de muestra del directorio:

[root@selinux tmp]# ll /sys/fs/selinux/

-rw-rw-rw-. 1 root root 0 mar 20 07:35 access

dr-xr-xr-x. 2 root root 0 mar 20 07:35 avc

dr-xr-xr-x. 2 root root 0 mar 20 07:35 booleans

-rw-r--r--. 1 root root 0 mar 20 07:35 checkreqprot

dr-xr-xr-x. 93 root root 0 mar 20 07:35 class

--w-------. 1 root root 0 mar 20 07:35 commit_pending_bools

-rw-rw-rw-. 1 root root 0 mar 20 07:35 context

-rw-rw-rw-. 1 root root 0 mar 20 07:35 create

-r--r--r--. 1 root root 0 mar 20 07:35 deny_unknown

--w-------. 1 root root 0 mar 20 07:35 disable

-rw-r--r--. 1 root root 0 mar 20 07:35 enforce

dr-xr-xr-x. 2 root root 0 mar 20 07:35 initial_contexts

-rw-------. 1 root root 0 mar 20 07:35 load

-rw-rw-rw-. 1 root root 0 mar 20 07:35 member

-r--r--r--. 1 root root 0 mar 20 07:35 mls

crw-rw-rw-. 1 root root 1, 3 mar 20 07:35 null

-r--r--r--. 1 root root 3698727 mar 20 07:35 policy

dr-xr-xr-x. 2 root root 0 mar 20 07:35 policy_capabilities

-r--r--r--. 1 root root 0 mar 20 07:35 policyvers

-r--r--r--. 1 root root 0 mar 20 07:35 reject_unknown

-rw-rw-rw-. 1 root root 0 mar 20 07:35 relabel

-r--r--r--. 1 root root 0 mar 20 07:35 status

-rw-rw-rw-. 1 root root 0 mar 20 07:35 user

El archivo de configuración → /etc/sysconfig/selinux

Hay dos formas de configurar SELinux bajo Red Hat Enterprise Linux y CentOS: usando → **system-config-securitylevel** (modo gráfico), o manualmente editando el archivo de configuración → **/etc/sysconfig/selinux**.

El archivo /etc/sysconfig/selinux es el archivo de configuración principal para habilitar o inhabilitar SELinux, así como también para configurar cuál política de debe imponer en el sistema y cómo hacerlo.

El archivo **/etc/sysconfig/selinux** contiene un enlace simbólico al archivo de configuración real, **/etc/selinux/config**.

[root@selinux carlos]# ls -al /etc/sysconfig/selinux

lrwxrwxrwx. 1 root root 17 feb 13 15:30 /etc/sysconfig/selinux -> ../selinux/config

A continuación se explica el subconjunto completo de opciones disponibles para la configuración.

[root@selinux carlos]# cat /etc/sysconfig/selinux

//

This file controls the state of SELinux on the system.

SELINUX= can take one of these three values:

enforcing - SELinux security policy is enforced.

permissive - SELinux prints warnings instead of enforcing.

disabled - No SELinux policy is loaded.

SELINUX=enforcing

SELINUXTYPE= can take one of three two values:

targeted - Targeted processes are protected,

minimum - Modification of targeted policy. Only selected processes are protected.

mls - Multi Level Security protection.

SELINUXTYPE=targeted

//

↘ SELINUX=

enforcing → Impositivo. Se impone la política de seguridad SELinux.

permissive → Permisivo. Se advierte, pero no se impone la política SELinux. Útil para propósitos de depuración. Se registran mas rechazos, aunque los sujetos podrán continuar con acciones que serian rechazadas en modo enforcing. Por ejemplo navegar en un arbol de directorios producirá varios mensajes '**avc : denied'** para cada nivel de directorio leído. En un kernel enforcing se detendría en la primera acción.

disabled → Inhabilitado. SELinux esta completamente deshabilitado. Los ganchos no estan conectados al kernel y el pseudo sistema de archivos no esta registrado.

↘ SELINUXTYPE=

targeted → Objetivo dirigido. Solo se protegen ciertos daemons. La imposición de políticas para estos daemons se puede activar o desactivar utilizando booleans en modo consola, o a traves de **system-config-securitylevel** en modo gráfico.

minimun → Versión mas ligera que targeted.

mls → Multi-Level-Security (MLS). Se definen capas de seguridad para cada objeto, donde solo los usuarios que posean los niveles correspondientes o superiores pueden acceder a las secciones, pero nunca al revés.

mcs → Multi-Category-Security (MCS). Muy similar a MCS, pero con patrones mas flexibles y faciles de aplicar. Segmenta la información al estilo Departamental.

strict → Protección SELinux completa para todos los daemons. Se definen los contextos de seguridad para todos los sujetos y objetos.

Alternativas a SELinux

↘ AppArmor

AppArmor ("Application Armor") es un programa de seguridad para Linux, lanzado bajo la licencia GPL. Actualmente se encarga de mantenerlo la empresa Novell. AppArmor permite al administrador del sistema asociar a cada programa un perfil de seguridad que restrinja las capacidades de ese programa. Complementa el modelo tradicional de control de acceso discrecional de Unix (DAC) proporcionando el control de acceso obligatorio (MAC).

Además de la especificación manual de perfiles, AppArmor incluye un modo de aprendizaje, en el que las violaciones del perfil son registradas pero no prevenidas. Este registro puede utilizarse para crear un perfil basado en el comportamiento típico del programa.

Está implementado utilizando la interfaz del núcleo "Linux Security Modules".

AppArmor fue creado en parte como alternativa a SELinux, que era criticado por los administradores por ser demasiado difícil de instalar y mantener. Al contrario que SELinux, que se basa en añadir etiquetas a los archivos, AppArmor trabaja con las rutas de los ficheros. Según sus autores, AppArmor es menos complejo y más fácil de aprender a utilizar para un usuario medio que SELinux. Añaden además que AppArmor necesita realizar pocas modificaciones en el sistema de ficheros mientras que SELinux necesita un sistema de ficheros que soporte sus atributos extendidos, lo que implica que no pueda controlar el acceso a archivos montados vía NFS. Con AppArmor no importa en qué clase de sistema de ficheros estén montados los archivos.

AppArmor representa una aproximación al problema de restringir las acciones que el software instalado puede realizar.

Otra aproximación similar a AppArmor es SELinux. Una diferencia importante es que identifica los objetos del sistema de ficheros por el nombre del inodo en lugar de hacerlo por la ruta. Esto significa que, por ejemplo, datos inaccesibles pueden llegar a serlo bajo SELinux si el usuario crea una nueva versión del mismo (una técnica usada con frecuencia), mientras que AppArmor continuaría denegando el acceso. Por otra parte, un archivo que es inaccesible puede llegar a serlo bajo AppArmor creando un enlace permanente al archivo, mientras que SELinux denegaría el acceso al enlace creado. (En ambos casos, una política por defecto de "ningún acceso" evitaría el problema.)

Mientras se sigue debatiendo cual de las dos aproximaciones es mejor, hasta la fecha no hay ninguna evidencia definitiva que haga a uno preferible al otro. La discusión sobre las ventajas y desventajas de cada método suele girar en torno a cual de los dos se alinea más con los mecanismos de control de UNIX/Linux, pero UNIX y Linux usan una combinación de control de acceso basado en la ruta y en el inodo de los ficheros. Observar también que cualquier sistema operativo tiene mecanismos de control de acceso.

SELinux y AppArmor también se diferencian significativamente en cómo se administran e integran en el sistema.

El aislamiento de procesos también se puede lograr por los mecanismos como la virtualización; el proyecto de OLPC, por ejemplo, usa cajas de arena (sandboxes) individuales para las aplicaciones en VServer.

AppArmor empezó a utilizarse en SUSE y openSUSE. En los últimos lanzamientos, está disponible desde la herramienta de administración (YaST), pero no está activado por defecto. En abril de 2007

fue portado/empaquetado para Ubuntu. La distribución francesa Mandriva Linux incluye AppArmor como uno de los componentes de su distribución en la edición Mandriva Linux 2008.

↘ Smack (tortazo).

Smack (nombre completo: **Kernel de control de acceso obligatorio simplificado**) es un módulo de seguridad del kernel de Linux que protege los datos y procesa la interacción de la manipulación maliciosa mediante un conjunto de reglas de control de acceso obligatorio (MAC). Se ha unido oficialmente desde la versión 2.6.25 de Linux, y era el mecanismo principal del control de acceso para el sistema operativo móvil de MeeGo. También se utiliza para las aplicaciones web HTML5 de sandbox en la arquitectura de Tizen, en las soluciones comerciales de Wind River Linux para el desarrollo de dispositivos embebidos, en los productos de Philips TV Digital. Y en el sistema operativo Ostro ™ de Intel para dispositivos IoT .
Smack consta de tres componentes:

- Un módulo del kernel que se implementa como un módulo de seguridad de Linux . Funciona mejor con sistemas de archivos que admiten atributos extendidos.
- Una secuencia de comandos de inicio que garantiza que los archivos de dispositivo tengan los atributos correctos de Smack y cargue la configuración de Smack.
- Un conjunto de parches para el paquete GNU Core Utilities para darle a conocer los atributos de archivo ampliado de Smack. También se creó un conjunto de parches similares a Busybox . SMACK no requiere soporte para el espacio del usuario.

Smack ha sido criticado por haber sido escrito como un nuevo módulo LSM en lugar de una política de seguridad SELinux que puede proporcionar funcionalidad equivalente. Tales políticas SELinux se han propuesto, pero no se ha demostrado ninguna. El autor de Smack respondió que no sería práctico debido a la complicada sintaxis de configuración de SELinux y la diferencia filosófica entre los diseños de Smack y SELinux.

Contextos

Definen los atributos de seguridad de sujetos y objetos individualmente. Cada contexto contiene un tipo o dominio y un nivel de seguridad asociado al sujeto u objeto. Los contextos se muestran con la opción '**-Z**'.

↘ Contexto de Usuarios

[root@selinux carlos]# id -Z

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

Para el usuario **root** no existen restricciones (**unconfined**), para el nivel de seguridad **0** y para las **1024** categorías.

[root@selinux carlos]# yum install setools-console

[root@selinux carlos]# seinfo -u

Users: 8

sysadm_u

system_u

xguest_u

root

guest_u

staff_u

user_u

unconfined_u

Por defecto SELinux incluye 7 usuarios ***confined*** ademas del usuario ***unconfined***.

[root@selinux carlos]# semanage login -l

Nombre de Ingreso	Usuario SELinux	Rango MLS/MCS	Servicio
__default__	unconfined_u	s0-s0:c0.c1023	*
root	unconfined_u	s0-s0:c0.c1023	*
system_u	system_u	s0-s0:c0.c1023	*

↘ Contexto de Procesos

[root@selinux carlos]# ps -eZ

LABEL	PID	TTY	TIME	CMD
system_u:system_r:init_t:s0	1	?	00:00:01	systemd
system_u:system_r:kernel_t:s0	2	?	00:00:00	kthreadd

…

El usuario Linux → **root** (#) es mapeado por SELinux como **system_u**, **init_t**, e indica la protección aplicada al proceso para un **sensitivity level 0 (s0)**.

↘ Contexto para Ficheros

[root@selinux carlos]# ll -Z /etc/passwd

-rw-r--r--**.** root root system_u:object_r:passwd_file_t:s0 /etc/passwd

==> El punto '**.**' final de '-rw-r—r--**.**', indica → **Contexto SELinux**.

==> El nivel de seguridad se define en:

→ **/etc/selinux/targeted/setrans.conf**

[root@selinux carlos]# cat /etc/selinux/targeted/setrans.conf

#

Multi-Category Security translation table for SELinux

#

Uncomment the following to disable translation libary

disable=1

#

Objects can be categorized with 0-1023 categories defined by the admin.

Objects can be in more than one category at a time.

Categories are stored in the system as c0-c1023. Users can use this

table to translate the categories into a more meaningful output.

Examples:

s0:c0=CompanyConfidential

s0:c1=PatientRecord

s0:c2=Unclassified

s0:c3=TopSecret

s0:c1,c3=CompanyConfidentialRedHat

s0=SystemLow

s0-s0:c0.c1023=SystemLow-SystemHigh

s0:c0.c1023=SystemHigh

==> Observad los mapeos humanamente mas legibles:

...

→ **s0:c0=CompanyConfidential**

...

==> Los contextos por defecto se definen en:

→ **/etc/selinux/targeted/contexts/files/file_contexts**

==> Los nuevos contextos creados se definen en:

→ **/etc/selinux/targeted/contexts/files/file_contexts.local**

↘ Contexto para Puertos

Se definen también atributos de seguridad SELinux para puertos de red individuales.

[root@selinux carlos]# semanage port -l

Tipo de Puerto SELinux Proto Número de Puerto

…

dhcpc_port_t tcp 68, 546, 5546

dhcpc_port_t udp 68, 546, 5546

dhcpd_port_t tcp 547, 548, 647, 847, 7911

dhcpd_port_t udp 67, 547, 548, 647, 847

dict_port_t tcp 2628

distccd_port_t tcp 3632

dns_port_t tcp 53

dns_port_t udp 53

...

==> SELinux perrmite escuchar / restringir puertos de red.

↘ Dominios de Transición

Un proceso en un dominio transita a otro dominio para ejecutar una aplicación , la cual tiene un '**entrypoint**' para el nuevo dominio o de entrada. El '**entrypoint**' permite definir políticas y controles SELinux para poder ser utilizadas en el nuevo dominio y poder ejecutar la aplicación correspondiente bajo el control de SELinux.

[root@selinux carlos]# ll -Z /usr/bin/passwd

-rwsr-xr-x. root root system_u:object_r:**passwd_exec_t**:s0 /usr/bin/passwd

[root@selinux carlos]# ll -Z /etc/shadow

----------. root root system_u:object_r:**shadow_t**:s0 /etc/shadow

[root@selinux carlos]# passwd

Cambiando la contraseña del usuario root.

Nueva contraseña:

==> En una sesión 'ssh' <u>diferente</u> comprobamos:

[root@selinux carlos]# ps -eZ | grep passwd

unconfined_u:unconfined_r:**passwd_t**:s0-s0:c0.c1023 1838 pts/1 00:00:00 passwd

==> '**passwd_t**' es el dominio '***entrypoint***' que permite efectuar el cambio de password aplicando las políticas SELinux ejecutando '**passwd_exec_t**'.

↘ Crear, Copiar, Mover, ... Archivos en un entorno SELinux sin valorar riesgos

'**Asghar Ghori**' advierte sobre la responsabilidad de copiar, mover y empaquetar archivos como una tarea principal del administrador de sistemas. Podemos exponer la seguridad del mismo si desconocemos una serie de normas:

1) Si se copia un archivo existente en el directorio actual u otro diferente, el contexto original del archivo se modifica por el contexto de destino.

[root@selinux carlos]# ls -Z /etc/passwd

-rw-r--r--. root root **system_u:object_r:passwd_file_t:s0** /etc/passwd

[root@selinux carlos]# cp /etc/passwd /tmp/

[root@selinux carlos]# ls -Z /tmp/passwd

-rw-r--r--. root root **unconfined_u:object_r:user_tmp_t:s0** /tmp/passwd

2) Si se copia un fichero en un directorio diferente como un nuevo fichero, el contexto original es reemplazado por el contexto por defecto del directorio destino, salvo que se especifique la opción '**--preserve=context**'.

[root@selinux carlos]# ll -Z /etc/passwd

-rw-r--r--. root root **system_u:object_r:passwd_file_t:s0** /etc/passwd

[root@selinux carlos]# cp /etc/passwd /tmp/passwd_01

[root@selinux carlos]# ll -Z /tmp/

-rw-r--r--. root root **unconfined_u:object_r:user_tmp_t:s0** passwd_01

[root@selinux carlos]# ll -Z /etc/passwd

-rw-r--r--. root root **system_u:object_r:passwd_file_t:s0** /etc/passwd

[root@selinux carlos]# cp --preserve=context /etc/passwd /tmp/passwd_02

[root@selinux carlos]# ll -Z /tmp/

-rw-r--r--. root root **unconfined_u:object_r:user_tmp_t:s0** passwd_01

-rw-r--r--. root root **system_u:object_r:passwd_file_t:s0** passwd_02

3) Si un fichero es movido en el mismo directorio u otro distinto el tipo o dominio permanecerá inalterado, si bien el atributo de usuario algunas veces, NO SIEMPRE cambiará a '**unconfined_u**'.

[root@selinux carlos]# ll -Z /tmp/

-rw-r--r--. root root **system_u:object_r:passwd_file_t:s0** passwd_02

[root@selinux carlos]# mv /tmp/passwd_02 /temporal/

[root@selinux carlos]# ll -Z /temporal/

-rw-r--r--. root root **system_u:object_r:passwd_file_t:s0** passwd_02

4) Si un fichero es empaquetado con '**tar**' debe utilizarse la opción '**--selinux**', para preservar el contexto.

[root@selinux html]# touch /var/www/html/archivo{1,2,3}

[root@selinux html]# ll -Z

-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

[root@selinux html]# tar --selinux -cf prueba.tar archivo{1,2,3}

[root@selinux html]# ll -Z

-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

-rw-r--r--. root root **unconfined_u:object_r:httpd_sys_content_t:s0** prueba.tar

Política Targeted → Política Dirigida (por defecto en RHEL)

Es la política por defecto en RHEL. Los procesos targeted son ejecutados en un dominio confinado, el resto de los procesos se ejecutan en un dominio NO confinado. En tiempo de ejecución se pueden modificar las políticas SELinux a traves de los Booleans.

↘ Procesos confinados

Cuando un proceso se confina corre en su propio dominio, de esta forma si es comprometido por un atacante el daño es limitado.

Verificamos con **sestatus**. La configuracion de arranque se configura en '**/etc/selinux/config'**, o en su enlace simbolico '**/etc/sysconfig/selinux**'.

[root@selinux selinux]# cat /etc/sysconfig/selinux

This file controls the state of SELinux on the system.

SELINUX= can take one of these three values:

enforcing - SELinux security policy is enforced.

permissive - SELinux prints warnings instead of enforcing.

disabled - No SELinux policy is loaded.

SELINUX=enforcing

SELINUXTYPE= can take one of three two values:

targeted - Targeted processes are protected,

minimum - Modification of targeted policy. Only selected processes are protected.

mls - Multi Level Security protection.

SELINUXTYPE=targeted

[root@selinux selinux]# sestatus

SELinux status: enabled

SELinuxfs mount: /sys/fs/selinux

SELinux root directory: /etc/selinux

Loaded policy name: targeted

Current mode: enforcing

Mode from config file: enforcing

Policy MLS status: enabled

Policy deny_unknown status: allowed

Max kernel policy version: 28

[root@selinux selinux]# touch /var/www/html/testfile

[root@selinux selinux]# ls -Z /var/www/html/testfile

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

==> Por defecto en RHEL, los usuarios se utilizan de forma NO confinada. RBAC se aplica para los procesos NO para los ficheros → **object_r** es un rol genérico, **httpd_sys_content_t** permite acceso al archivo.

[root@selinux html]# systemctl start httpd.service

[root@selinux tmp]# wget http://localhost/testfile

--2017-03-22 13:30:53-- http://localhost/testfile

Resolviendo localhost (localhost)... ::1, 127.0.0.1

Conectando con localhost (localhost)[::1]:80... conectado.

Petición HTTP enviada, esperando respuesta... 200 OK

Longitud: 0 [text/plain]

Grabando a: “testfile”

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

2017-03-22 13:30:53 (0,00 B/s) - “testfile” guardado [0/0]

==> Hacemos un relabel diferente de ‘**httpd_sys_content_t’** para testfile. Por ejemplo: ‘**samba_share_t**’, con el comando de cambio de contexto temporal ‘**chcon**’.

[root@selinux tmp]# seinfo -t |grep samba_

...

samba_unit_file_t

samba_share_t

samba_initrc_exec_t

...

[root@selinux tmp]# ls -Z /var/www/html/testfile

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

[root@selinux tmp]# chcon -t samba_share_t /var/www/html/testfile

[root@selinux tmp]# ls -Z /var/www/html/testfile

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

[root@selinux tmp]# wget http://localhost/testfile

--2017-03-23 07:39:54-- http://localhost/testfile

Resolviendo localhost (localhost)... ::1, 127.0.0.1

Conectando con localhost (localhost)[::1]:80... conectado.

Petición HTTP enviada, esperando respuesta... 403 Forbidden

2017-03-23 07:39:54 ERROR 403: Forbidden.

==> Se ha producido una denegación de acceso al cambiar el contexto.

[root@selinux tmp]# cat /var/log/messages

Mar 23 07:40:02 selinux setroubleshoot: SELinux is preventing httpd from getattr access on the file /var/www/html/testfile. For complete SELinux messages. run sealert -l 884b91aa-0769-447a-b0d4-067cc591fc91

[root@selinux tmp]# cat /var/log/audit/audit.log

type=AVC msg=audit(1490251194.730:70): avc: denied { getattr } for pid=1140 comm="httpd" path="/var/www/html/testfile" dev="dm-0" ino=34098251 scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:samba_share_t:s0 tclass=file

↘ Procesos NO confinados

Los procesos no confinados corren en dominios no confinados, por ejemplo los programas **init** corren en el dominio no confinado **unconfined_service_t**, los procesos no confinados del **kernel** corren en el dominio no confinado **kernel_t**, y los **usuarios no confinados** de Linux corren en el dominio no confinado **unconfined_t**. Los procesos que corren en dominios no confinados usan exclusivamente reglas DAC. SELinux es una mejora de seguridad NO reemplaza DAC.

[root@selinux tmp]# touch /var/www/html/test2

[root@selinux tmp]# ls -Z /var/www/html/test2

-rw-r--r--. root root **unconfined_u:object_r:httpd_sys_content_t:s0** /var/www/html/test2

==> Por defecto el usuario corre como no confinado→ **unconfined_u**. **RBAC** no tiene sentido para usuarios, si para procesos→ **object_r**, en este caso es un rol generico. **httpd_sys_content_t** permite a **httpd** acceder al archivo.

[root@selinux tmp]# chcon -t samba_share_t /var/www/html/test2

[root@selinux tmp]# ls -Z /var/www/html/test2

-rw-r--r--. root root **unconfined_u:object_r:samba_share_t:s0** /var/www/html/test2

==> Ejecutamos httpd a continuación como un proceso NO confinado.

[root@selinux tmp]# chcon -t bin_t /usr/sbin/httpd

[root@selinux tmp]# ls -Z /usr/sbin/httpd

-rwxr-xr-x. root root **system_u:object_r:bin_t:s0** /usr/sbin/httpd

[root@selinux tmp]# systemctl start httpd.service

[root@selinux tmp]# systemctl status httpd.service

● httpd.service - The Apache HTTP Server

Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)

Active: active (running) since jue 2017-03-23 10:18:37 CET; 10s ago

├─1471 /usr/sbin/httpd -DFOREGROUND

├─1472 /usr/sbin/httpd -DFOREGROUND

├─1473 /usr/sbin/httpd -DFOREGROUND

├─1474 /usr/sbin/httpd -DFOREGROUND

├─1475 /usr/sbin/httpd -DFOREGROUND

└─1476 /usr/sbin/httpd -DFOREGROUND

mar 23 10:18:37 selinux.example.com systemd[1]: Starting The Apache HTTP Server...

mar 23 10:18:37 selinux.example.com systemd[1]: Started The Apache HTTP Server.

[root@selinux tmp]# ps -eZ |grep httpd

system_u:system_r:unconfined_service_t:s0 1471 ? 00:00:00 httpd

system_u:system_r:unconfined_service_t:s0 1472 ? 00:00:00 httpd

system_u:system_r:unconfined_service_t:s0 1473 ? 00:00:00 httpd

system_u:system_r:unconfined_service_t:s0 1474 ? 00:00:00 httpd

system_u:system_r:unconfined_service_t:s0 1475 ? 00:00:00 httpd

system_u:system_r:unconfined_service_t:s0 1476 ? 00:00:00 httpd

[root@selinux tmp]# wget http://localhost/test2

--2017-03-23 10:24:11-- http://localhost/test2

Resolviendo localhost (localhost)... ::1, 127.0.0.1

Conectando con localhost (localhost)[::1]:80... conectado.

Petición HTTP enviada, esperando respuesta... 200 OK

Longitud: 0 [text/plain]

Grabando a: “test2”

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

2017-03-23 10:24:11 (0,00 B/s) - “test2” guardado [0/0]

Nos ha permito hacer un wget.

==> Para restaurar de nuevo el contexto por defecto, utilizamos **restorecon**:

[root@selinux tmp]# ls -Z /usr/sbin/httpd

-rwxr-xr-x. root root system_u:object_r:bin_t:s0 /usr/sbin/httpd

[root@selinux tmp]# restorecon -v /usr/sbin/httpd

restorecon reset /usr/sbin/httpd context system_u:object_r:bin_t:s0->system_u:object_r:httpd_exec_t:s0

[root@selinux tmp]# ls -Z /usr/sbin/httpd

-rwxr-xr-x. root root **system_u:object_r:httpd_exec_t:s0** /usr/sbin/httpd

↘ Usuarios confinados y NO confinados

Los usuarios Linux se mapean como usuarios SELinux, lo que permite que hereden las restricciones SELinux.

[root@selinux tmp]# semanage login -l

Nombre de Ingreso	Usuario SELinux	Rango MLS/MCS	Servicio
__default__	unconfined_u	s0-s0:c0.c1023	*
root	unconfined_u	s0-s0:c0.c1023	*
system_u	system_u	s0-s0:c0.c1023	*

Los usuarios Linux se mapean al ingreso como **__default__** de forma no confinada de la misma forma que **root**.

[root@selinux tmp]# id -Z

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

==> Creamos un nuevo usuario Linux→ **Linux_usuario** para mapearlo como usuario SELinux→ **SELinux_user_u**.

[root@selinux tmp]# useradd Linux_usuario

[root@selinux tmp]# passwd Linux_usuario

[root@selinux tmp]# su Linux_usuario

[usuario01@Linux_usuario tmp]$ id -Z

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

Si los usuarios Linux no confinados pueden ejecutar una aplicación en un dominio confinado se verán afectados por las políticas de ese dominio.

[root@selinux tmp]# semanage user -a -r s0-s0:c0.c1023 -R "staff_r" SELinux_usuario_u

[root@selinux tmp]# seinfo -u

Users: 10

sysadm_u

system_u

xguest_u

root

guest_u

staff_u

user_u

SELinux_usuario_u

unconfined_u

[root@selinux tmp]# semanage user -l

Etiquetado MLS/ MLS/

Usuario SELinux Prefijo Nivel MCS Rango MCS Roles SELinux

SELinux_usuario_u user s0 s0-s0:c0.c1023 staff_r

guest_u user s0 s0 guest_r

root user s0 s0-s0:c0.c1023 staff_r sysadm_r system_r unconfined_r

staff_u user s0 s0-s0:c0.c1023 staff_r sysadm_r system_r unconfined_r

sysadm_u user s0 s0-s0:c0.c1023 sysadm_r

system_u user s0 s0-s0:c0.c1023 system_r unconfined_r

unconfined_u usuario01 s0 s0-s0:c0.c1023 system_r unconfined_r

user_u user s0 s0 user_r

xguest_u user s0 s0 xguest_r

==> Mapeamos '**SELinux_usuario_u'** con '**Linux_usuario'**

[root@selinux tmp]# semanage login -l

Nombre de Ingreso Usuario SELinux Rango MLS/MCS Servicio

__default__ unconfined_u s0-s0:c0.c1023 *

root unconfined_u s0-s0:c0.c1023 *

system_u system_u s0-s0:c0.c1023 *

[root@selinux tmp]# semanage login -a -s SELinux_usuario_u -r s0-s0:c0.c1023 Linux_usuario

[root@selinux tmp]# semanage login -l

Nombre de Ingreso Usuario SELinux Rango MLS/MCS Servicio

Linux_usuario SELinux_usuario_u s0-s0:c0.c1023 *

__default__ unconfined_u s0-s0:c0.c1023 *

root unconfined_u s0-s0:c0.c1023 *

system_u system_u s0-s0:c0.c1023 *

==> Por último si queremos eliminar el mapeo del usuario y el usuario→ **SELinux_usuario_u**

[root@selinux tmp]# semanage login Linux_usuario --delete --seuser SELinux_usario_u

[root@selinux tmp]# semanage user -d SELinux_usario_u

[root@selinux tmp]# semanage login -l

Nombre de Ingreso Usuario SELinux Rango MLS/MCS Servicio

__default__ unconfined_u s0-s0:c0.c1023 *

root unconfined_u s0-s0:c0.c1023 *

system_u system_u s0-s0:c0.c1023 *

SELinux Booleans

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.

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:

[root@selinux tmp]# semanage boolean -l

[root@selinux tmp]# semanage boolean -l|grep ftp_home_dir

tftp_home_dir (apagado,apagado) Permite a tftp leer y escribir archivos en los directorios home de usuario

En el 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.

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:

[root@selinux tmp]# getsebool -a

abrt_anon_write --> off

abrt_handle_event --> off

abrt_upload_watch_anon_write --> on

antivirus_can_scan_system --> off

…

Una lista separada por espacio para listar los Booleanos múltiples:

[root@selinux tmp]# getsebool allow_console_login allow_cvs_read_shadow

login_console_enabled --> on

cvs_read_shadow --> off

[root@selinux tmp]# getsebool httpd_can_network_connect_db

httpd_can_network_connect_db --> **off**

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.

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.

[root@selinux tmp]# setsebool httpd_can_network_connect_db on

[root@selinux tmp]# getsebool httpd_can_network_connect_db

httpd_can_network_connect_db --> **on**

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

[root@selinux tmp]# setsebool -P httpd_can_network_connect_db on

Comandos de Control SELinux por Áreas

↘ **Por Contexto**

chcon → cambia el contexto de seguridad de un archivo en SELinux Temporalmente.

-u, --user=USER

set user USER in the target security context

-r, --role=ROLE

set role ROLE in the target security context

-t, --type=TYPE

set type TYPE in the target security context

-l, --range=RANGE

set range RANGE in the target security context

-R, --recursive

operate on files and directories recursively

-v, --verbose

output a diagnostic for every file processed

[root@selinux tmp]# chcon -vu staff_u -t var_run_t /root

cambiando el contexto de seguridad de «/root»

[root@selinux tmp]# ls -dZ /root/

dr-xr-x---. root root staff_u:object_r:var_run_t:s0 /root/

matchpathcon → muestra el contexto de seguridad SELinux por defecto de un directorio determinado desde el fichero de configuración de contexto.

[root@selinux tmp]# matchpathcon /root/

/root system_u:object_r:admin_home_t:s0

restorecon

semanage

↘ Por Control General

↘ Por Políticas

↘ Por Boolean Control

↘ Por Problemas (Troubleshooting)

↘ Por Entorno Gráfico

Compilación de Políticas

Archivos te, pp, sh y man pages, y mas...

BIBLIOGRAFIA Y REFERENCIAS:

RHCSA & RHCE RedHat Enterprise Linux 7 -Asghar Ghori

RedHat Enterprise Linux 7 – SELinux User's and Administrator's Guide

http://www.cadilinea.com/blog/linux-professional-institute-lpic/lpic-303-seguridad/

https://linuxitomex.wordpress.com/2010/05/28/manual-de-selinux-haciendo-nuestra-seguridad-impenetrable-gnulinux/

https://es.wikipedia.org/wiki/AppArmor

https://en.wikipedia.org/wiki/Smack_(software

https://nb.fedorapeople.org/cvsfedora/web/html/docs/selinux-user-guide/f11/es-ES/html/sect-Security-Enhanced_Linux-Working_with_SELinux-Maintaining_SELinux_Labels_.html

https://nb.fedorapeople.org/cvsfedora/web/html/docs/selinux-user-guide/f11/es-ES/pdf/