SELinux: Actividad práctica

Tiempo: 3 horas.

Archivos, órdenes y servicios que se utilizarán:

- SELinux: getenforce, getsebool, restorecon, runcon, seinfo, semanage, sestatus, setenforce, setsebool.
- Servicio systemd: systemctl
- Servicio de cortafuegos firewalld: firewall-cmd
- Servicio httpd: archivo /etc/httpd/conf/httpd.conf

Prerrequisito para abordar la actividad práctica:

- Tener instalado el paquete httpd (servicio Apache) y configurado correctamente el sistema para que este servicio funcione correctamente. Recuerde que esto se abordó en la Práctica 1 de la asignatura.
- Tener configurado correctamente el redireccionamiento de puertos en la configuración de la máquina virtual en la que se ejecuta el servicio Apache para que, con un navegador desde el sistema anfitrión, se pueda acceder al servidor Apache.

Bibliografía

- Product Documentation for Red Hat Enterprise Linux 8. Uso de SELinux. https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/using_SELinux/index

1. Introducción

SELinux es un entorno de seguridad para sistemas Linux de tipo MAC (Mandatory Access Control). Esto significa que el control del acceso que hacen los procesos a los recursos del sistema se basa en políticas que son responsabilidad del administrador del sistema y no de los usuarios propietarios de los recursos. Con el fin de mejorar la seguridad que proporciona el esquema tradicional y básico que los sistemas Linux (modelo DAC), en la mayoría de sus distribuciones, SELinux se instala por defecto. En un sistema en el que se ejecuta SELinux, el control del acceso se lleva a cabo, primero, realizando el control de acceso basado en el modelo DAC y, a continuación, si el acceso es permitido, se lleva a cabo el control de acceso de SELinux. La filosofía del control de acceso que hace SELinux se resume en esta pregunta:
¿Puede este **Proceso** realizar esta **Acción** en este **Objeto**?

Las políticas utilizadas por SELinux se implementan mediante reglas que, utilizando datos de control propios de SELinux, permiten o no permiten la realización de la acción.

Como se verá durante esta actividad práctica, en un sistema Linux, los dos pilares básicos en los que se sustenta un control de acceso seguro son SELinux y el servicio de cortafuegos. Por ello, se realizarán actividades que implican el manejo de ambos servicios. Estas actividades ayudarán a comprender el papel que juegan ambos servicios, que a veces se confunde.

El objetivo de esta actividad práctica es el aprendizaje de aspectos básicos de SELinux. Concretamente:

- Responder a la pregunta ¿por qué es conveniente que SELinux se ejecute en el sistema?
- Entender el significado de los contextos SELinux que tienen archivos y procesos y las variables boolenas.
- Manejo de los modos de ejecución de SELinux, conociendo sus implicaciones.
- Administrar usuarios SELinux.
- Administración la seguridad SELinux de aplicaciones y servicios con configuraciones no estándares.
- Entre los distintos servicios del sistema que intervienen en la seguridad del sistema, saber identificar el papel que juega SELinux.

2. Información de contexto SELinux y variables booleanas SELinux

La implementación de las reglas SELinux utiliza tipos de datos: las etiquetas de contexto SELinux y las variables boolenas SELinux. En este apartado se describe el significado de estos dos tipos de datos y cómo obtener sus valores.

2.1. Etiquetas SELinux

Una etiqueta SELinux es una estructura de datos que poseen archivos y procesos. Una etiqueta SELinux consta de cinco atributos, tres de ellos obligatorios y dos optativos.

```
Usuario:Rol:Tipo:Nivel:Categoria
```

El primer atributo obligatorio es `Usuario`. SELinux posee su propio conjunto de usuarios. Este atributo define qué usuario SELinux tiene asociado un archivo o proceso. No confundir este usuario con el usuario propietario del archivo o el proceso. SELinux realiza una asociación entre los usuarios del sistema y sus usuarios SELinux. En todo momento, un usuario del sistema en sesión tiene asociado un usuario SELinux.

El atributo `Rol` es el segundo atributo obligatorio. En el caso de un proceso, este atributo define qué rol posee el proceso en el sistema. En el caso de un recurso del sistema, este atributo posee el valor `object_r`. El uso de este atributo permite a SELinux controlar el tipo de acciones que puede realizar un usuario del sistema en función del rol que posea.

El tercer campo obligatorio es el `Tipo`. Mediante este atributo se definen distintos dominios, de manera que los procesos pertenecientes a un determinado dominio pueden acceder a los recursos adscritos a ese dominio. Si la seguridad de un proceso perteneciente a un dominio se ve comprometida, entonces solo estaría comprometida la seguridad de ese dominio y no la del resto de dominios definidos en el sistema. Un ejemplo de dominio es el que se define para un servicio Apache. En este dominio, los procesos (por ejemplo, el proceso httpd) y los recursos (por ejemplo, ficheros de configuración, ficheros de contenidos y sockets de conexión) que intervienen en este servicio dan lugar a un dominio específico http. La política de seguridad SELinux denominada “Targeted” se fundamenta en esta filosofía de control de acceso.

El cuarto atributo de una etiqueta SElinuix es opcional y se denomina `Nivel`. Este atributo permite ejecutar políticas de seguridad basadas en el grado de sensibilidad de los datos, representándose los distintos grados de sensibilidad mediante valores enteros que se

denominan niveles. En general a este tipo de políticas se le denominan Políticas de Seguridad Multinivel (MLS). Un ejemplo esquema clásico de este tipo de política consiste en asumir cuatro niveles de sensibilidad:

- Alto secreto (nivel más alto).
- Secreto (alto).
- Confidencial (bajo).
- No clasificado (más bajo)

En las políticas MLS, a los usuarios y procesos se les denomina sujetos y a los recursos objetos. Un sujeto puede leer objetos etiquetados con su mismo nivel de seguridad o que tengan niveles inferiores. Sin embargo, sólo puede escribir en objetos que posean su mismo nivel de seguridad. Por ejemplo, supongamos un proyecto en una empresa en el que participan varios equipos de personas pertenecientes a distintos departamentos. Además, existe un responsable del proyecto y, en un nivel inferior de jerarquía, cada equipo posee un responsable de equipo. Supongamos que toda la documentación del proyecto se gestiona en un sistema en el que el control de acceso a esta documentación se realiza mediante una política de seguridad MLS. Asumamos también que en este sistema al jefe de proyecto y los procesos que éste ejecute pertenecen a la categoría más alta ("alto secreto") y los responsables de equipo y sus procesos pertenecen a la categoría inferior ("secreto"). Como consecuencia de esta clasificación, los responsables de equipo podrán leer y modificar un documento etiquetado como "secreto" y el jefe de proyecto solo podrá leerlo. Por ejemplo, si este documento fuera el prepuesto del proyecto, entonces solo los jefes de equipo podrían crear o modificar el presupuesto; el jefe de proyecto sólo podrá supervisar cómo los jefes de equipo van modificando el presupuesto, pero no modificar el presupuesto. El resto de los miembros que participan en el proyecto que tengan un nivel inferior no podrían acceder de ninguna forma al presupuesto. Actualmente, las distribuciones de Linux basadas en Red Hat incorporan una política MLS en la que se definen hasta 15 niveles de sensibilidad: `s0` el nivel menos sensible y `s15` el nivel más sensible.

El quinto campo de una etiqueta SELinux se denomina Categoría y es opcional como el campo anterior. Este campo permite desplegar políticas de seguridad basadas en los distintos niveles de confidencialidad que puedan tener los datos del sistema. En una política de este tipo, los procesos y los recursos del sistema tienen asignados un grado de confidencialidad, denominado categoría, de manera que un recurso sólo puede ser accedido por un proceso que tenga su misma categoría. En general a este tipo de políticas se les denomina Políticas de Seguridad Multicategoría (MCS). Actualmente, para este tipo de políticas, en las distribuciones de Linux basadas en Red Hat se pueden establecer hasta 1024 categorías de confidencialidad diferentes representándose por `c0, c1, …, c1023`.

En este punto es importante indicar que las políticas basadas en dominios, que hacen uso del atributo `Tipo`, las políticas basadas en la sensibilidad de los datos, que hacen uso del atributo `Nivel`, y las políticas basadas en la confidencialidad de los datos, que hacen uso del atributo `Categoría`, no son excluyentes y se pueden aplicar conjuntamente.

En esta actividad práctica se trabajará con un entorno SELinux en el que se aplica la política de seguridad denominada "`Targeted`" anteriormente mencionada. Actualmente, los sistemas Linux basados en la distribución Red Hat, cuando se instalan, por defecto aplican esta política. Un ejemplo de uso combinado de estas políticas es el entorno de virtualización KVM para sistemas Linux. Cuando se instala este entorno de virtualización, los procesos y recursos que

forman parte de este entorno configuran un dominio SELinux virt y además se utiliza una política MCS para controlar los recursos que deben ser accesibles de manera exclusiva o de manera compartida por parte de las máquinas virtuales.

La información de contexto de un archivo se puede obtener usando la orden `ls` con la opción `-Z`. Por ejemplo, la siguiente orden obtiene la información de contexto de los archivos ubicados en el directorio raíz:

```
# ls -Z /
```

La siguiente orden obtiene la información de contextos de los archivos ubicados en el directorio `/var/www`. Observe el atributo tipo en los contextos de que poseen los directorios `cgi-bin` y `html`.

```
# ls -Z /var/www
```

Los procesos también poseen información de contexto SELinux. La orden `ps` puede reportar la información de contexto de los procesos, para ello se debe utilizar la opción `-Z`. La siguiente orden obtiene la información de contexto de todos los procesos que consistan en la ejecución de un archivo ejecutable cuyo nombre es `httpd`.

```
# ps -Z -C httpd
```

Para obtener la información de contexto de la sesión de usuario en la que estamos, ejecute la orden:

```
# id -Z
```

La información de contexto de la sesión será heredada por todas las órdenes que ejecute en la sesión.

2.2. Variables boolenas SELinux

SELinux utiliza este tipo de variables para controlar si un determinado servicio o aplicación puede utilizar una determinada funcionalidad del sistema. En general, por funcionalidad se entiende una capacidad específica que tiene un servicio. Por cada servicio y funcionalidad que SELinux controla de esta manera, se define una variable que puede almacenar el valor “on” si está permitido o el valor “off” si no está permitido. Mediante este tipo de variables SELinux lleva a cabo un control de acceso “de grano grueso” sobre el uso que puede hacer un servicio o aplicación de una determinada funcionalidad del sistema. Si la variable asociada a un determinado servicio y funcionalidad posee el valor de “off”, entonces bajo ningún concepto se permitirá que el servicio o aplicación utilice la funcionalidad. Si la variable posee el valor “on”, entonces se permitirá el uso de la funcionalidad según las reglas de control de acceso basadas en la información de contexto. Por tanto, modificando el valor de estas variables se puede cambiar la política de control de acceso sin necesidad de conocer los detalles de la política. Para obtener el repertorio de variables booleanas definidas en el sistema, su valor y el significado de cada variable, ejecute la siguiente orden:

```
# semanage boolean -l
```

Para obtener el listado de variables con sus valores:

```
# getsebool -a
```

Para obtener el valor de una variable o lista de variables:

```
# getsebool variable1 variable2 …
```

Por ejemplo, la siguiente orden permite obtener el valor de la variable `httpd_can_network_connect_db`. Esta variable establece si se permite (valor "on") o no se permite (valor "off") que el servicio httpd pueda utilizar un servicio de base de datos remoto.

```
# getsebool httpd_can_network_connect_db
```

3. Modos de ejecución de SELinux:

SElinix se puede ejecutar de tres maneras diferentes:

- Modo `enforcing`: en este modo se realiza el control de acceso según las reglas definidas en la política de seguridad que se aplica.
- Modo permissive: en este modo se aplican las reglas definidas en la política de seguridad, sin embargo, no se producen denegaciones de acceso. En el caso de que, según las reglas, un proceso no deba acceder a un recurso, entonces solo se emite un mensaje indicando la denegación de acceso, pero no se realiza la denegación de acceso.
- Modo `disabled`: en este modo SELinux no se ejecuta y, por tanto, el control de acceso sólo se realiza siguiendo el modelo básico DAC (Discretionary Access Control).

Para conocer en qué modo de ejecución se encuentra SELinux, ejecutar la orden:

```
# getenforce
```

Otra manera que se puede utilizar y que proporciona más información es mediante la orden:

```
# sestatus
```

El archivo **`/etc/selinux/config`** es un archivo de configuración que especifica el modo de ejecución que tendrá el SELinux, una vez que arranque el sistema. Además, también se especifica la política de control de acceso que se aplica, que por defecto es la política denominada `targeted`. Obsérvese que se existen dos políticas alternativas: `munimum` y `mls`.

SELinux es un módulo se seguridad que forma parte del núcleo del sistema operativo. Actualmente se trata de un módulo estático que se incorpora al núcleo cuando éste se compila. Por tanto, no se puede configurar que se cargue dinámicamente durante el funcionamiento del sistema. La siguiente orden lista el estado de los diferentes módulos dinámicos que formar parte del núcleo del sistema.

```
# lsmod
```

Ninguno de los módulos reportados por la orden se corresponde con SELinux y, sin embargo, se está ejecutando. La explicación es que, tal y como se ha expresado, se trata de un componente estático del núcleo.

Los mensajes que emite durante su ejecución se almacenan en el archivo **/var/log/audit/audit.log**, estos mensajes comienzan con el texto **type=AVC**. La siguiente orden obtiene estos mensajes:

```
# grep "^type=AVC" /var/log/audit/audit.log
```

Se aborda ahora cómo cambiar el modo de ejecución del SELinux. Existen tres formas de hacerlo, mediante la orden `setenforce`, cambiando el contenido del fichero `/etc/SELinux/config` o modificando los parámetros de carga del núcleo en el momento del arranque del sistema.

Para cambiar el modo de funcionamiento del SELinux, estando el sistema ya arrancado, se debe usar la orden `setenforce`. La siguiente orden pone el modo de funcionamiento del SELinux en modo `permissive`.

```
# setenforce 0
```

Alternativamente la siguiente orden lo pone hace que el SELinux aplique la política de control de acceso `enforcing`.

```
# setenforce 1
```

La orden `setenforce` no configura de manera permanente el modo de funcionamiento de SELinux. Si se quiere establecer de manera permanente su modo de funcionamiento, entonces se debe modificar el contenido del archivo `/etc/SELinux/config`.

La tercera forma de establecer el modo de funcionamiento del SELinux consiste en especificar, durante el arranque del sistema, que el núcleo no ejecute el SELinux en modo enforced (en los sistemas que se instalan con el SELinux, por defecto el núcleo asume que debe ejecutar el SELinux). Para ello, si:

- Se especifica `enforcing=0`, entonces, independientemente de cómo esté configurado el sistema, el SELinux se ejecutará en modo permisivo.
- Se especifica `SELinux=0`, entonces, independientemente de cómo esté configurado el sistema, el SElinux no se ejecutará, es decir su modo de ejecución será deshabilitado.

Hay que tener presente que el interactuar con el sistema con el SELinux deshabilitado implica que no solo no se lleva a cabo el control de acceso; también implica que, mientras SELinux esté deshabilitado (modo `disabled`), las operaciones de etiquetado de contextos de los archivos no se realizan. Esto puede dar lugar a que cuando el SELinux vuelva a ejecutarse (`enforcing` o `permissive`), éste no funcione correctamente debido a que existen archivos con información de contexto errónea. Esta es la razón por la que, en ocasiones, cuando se realiza alguna operación de mantenimiento crítico del sistema con SELinux desactivado, es necesario comprobar la información de contexto de los archivos y, si fuera necesario, actualizar dicha información. Por ello, se recomienda que para realizar labores de mantenimiento no se deshabilite SELinux. Se recomienda que se ejecute en modo `permissive`.

Existen dos maneras de forzar la regeneración de los datos de contexto de los archivos. La primera es especificando en el arranque del sistema el parámetro de carga del núcleo `autorelabel=1`. Otra formar de regeneración consiste en ejecutar las siguientes órdenes:

```
# touch /.autorelabel
# reboot
```

4. Gestión de usuarios: usuarios confinados y no confinados

SELinux posee su propio conjunto de usuarios. A todo usuario del sistema, SELinux le asocia un usuario SELinux. Esta asociación se define en las reglas. La siguiente orden obtiene información de los usuarios:

```
# semanage login -l
```

En los sistemas Linux basados en la distribución Red Hat, por defecto todos los usuarios se asocian al usuario `unconfined_u` de SELinux.

La siguiente orden reporta los usuarios SELinux definidos en el sistema:

```
# seinfo -u
```

Como se observará, el intérprete de órdenes nos informa que no puede ejecutar esta orden debido a que no la encuentra. Esto se debe a que, como consecuencia de la instalación del sistema realizada, solo se instaló un repertorio mínimo de órdenes SELinux. Por tanto, para poder ejecutarla orden anterior, tal y como el sistema indica, se debe instalar el paquete setools-console.

```
# dnf install setools-console -y
```

Una vez instalado el paquete, la orden reporta los usuarios SELinux existentes en el sistema.

Para conocer los roles SELinux definidos en el sistema, ejecute la siguiente orden:

```
# seinfo -r
```

Los usuarios del sistema poseen un atributo que especifica a qué usuario SElinux se asocia cuando éste inicia una sesión. Este atributo de configuración se establece con la opción `-Z` en las órdenes `useradd` o `usermod`. En el caso de que el usuario no tenga definido este atributo, entonces al usuario se le establece un contexto de usuario no confinado y se le asocia al tipo de usuario del sistema **`__default__`**. Dependiendo del contexto SELinux que tenga el usuario, SELinux permitirá o no permitirá ejecutar órdenes. Para comprobar este comportamiento, realice las siguientes acciones:

- Cree tres nuevos usuarios denominados u1SELinux, u2SELinux y u3SELinux. En la creación del primero de ellos, utilice especifique la opción -Z user_u. En la creación del segundo de ellos, especifique la opción -Z sysadm_u. En la creación del tercero no utilice esta opción.
- Para que pueda iniciar una sesión con estos nuevos usuarios, establezca sus palabras de paso.
- Con cada una de las nuevas cuentas de usuario, realice las siguientes comprobaciones:
  - Qué ocurre si intenta autenticarse mediante el entorno gráfico.
  - En las sesiones que ha podido iniciar desde el entono gráfico, si abre un terminal, qué contextos tienen asignados las sesiones.
  - Qué ocurre en cada una de las sesiones cuando el usuario intenta ejecutar la orden `su.`

Como habrá comprobado, cuando un usuario inicia una sesión, a ésta se le asigna automáticamente un contexto SELinux y como consecuencia de ello se le asigna un dominio SELinux asociado al atributo tipo. En el ejemplo que se ha propuesto, se tendrán tres dominios: `user_t`, `sysadm_t` y `unconfined_t`. Dependiendo del dominio al que pertenezca la sesión, entonces se permitirá o no permitirá autenticarse mediante el entorno gráfico o se permitirá ejecutar la orden su. Si el contexto SELinux de la sesión indica que pertenece al dominio `unconfined_t`, entonces la ejecución de cualquier orden sólo estará controlada por el modelo básico de control de acceso del sistema (modelo DAC).

Tal y como ya se ha mencionado, cuando un usuario inicia una sesión, si su atributo de usuario SELinux no está definido, entonces, por defecto, el sistema le asocia automáticamente un usuario y un contexto de no confinado (**`unconfined_u:unconfined_r:unconfined_t`**). Si se quiere que este comportamiento por defecto cambie, es decir, asociar a un usuario que no tiene definido el usuario SELinux, un usuario y contexto SELinux por defecto, entonces ejecutar la orden:

```
#  semanage login -m -s user_u -r s0 __default__
```

La orden anterior configura el sistema para que en el caso de que una cuenta de usuario no tenga definido el atributo de usuario SELinux, entonces al iniciar una sesión se le asocie automáticamente el usuario SELinux **`user_u`**.

Ejecute la orden anterior y observe qué ocurre ahora cuando inicia una sesión mediante el entorno gráfico con el usuario u3SELinux y ejecuta la orden:

```
# id -Z
```

Si quisiera que el comportamiento del sistema fuera el mismo que tenía antes de ejecutar la última orden `semanage`, es decir que los usuarios del sistema que no tengan establecido el atributo de usuario SELinux al iniciar una sesión se les asocie automáticamente el usuario SELinux `unconfined_u` ¿qué orden tendría que ejecutar?

5. Configuración de SELinux de aplicaciones y servicios

Tal y como ya se ha expresado, todas las distribuciones de Linux basadas en Red Hat, como es el caso de Fedora, incorporan SELinux. Estas distribuciones cuando se instalan configuran el sistema para que SELinux se ejecute en modo `enforced` y aplicando la política de seguridad `targeted`. El administrador del sistema, además de cambiar su modo de ejecución y modificar la configuración de usuarios SELinux, puede cambiar las reglas que implementan la política de seguridad que se ejecuta. Como norma general, se recomienda cambiar lo menos posible la configuración de SELinux. Sin embargo, cuando una aplicación o servicio posee alguna característica de configuración que varía con respecto a una instalación estándar, entonces puede que sea necesario hacer algún cambio puntual de la configuración de SELinux de esta aplicación o servicio. Este apartado trata de cómo realizar este tipo de cambios puntuales.

El caso de demostración que se va a desarrollar consiste en cambiar la configuración SELinux del servicio httpd (servidor Apache) debido a que se quiere que:

1. Establecer como puerto de uso del servidor Apache un puerto distinto al asumido en una configuración estándar. En una configuración estándar se asume que el puerto 80 se utiliza para conexiones no seguras y el puerto 443 para conexiones seguras. El

cambio que se debe realizar consiste en configurar como nuevo puerto de uso del servicio el puerto 3131, en vez del puerto estándar 80 para conexiones no seguras.

2. Ubicar los archivos de contenidos que el servicio Apache debe mostrar a los usuarios en una ubicación distinta a la establecida en una configuración estándar. El cambio que se debe realizar consiste en ubicar los archivos de contenidos en el directorio /Apache/www.
3. Permitir que el servidor Apache pueda acceder a archivos ubicados en espacios de almacenamiento proporcionados por un servidor NFS remoto.

Como paso previo a la descripción de las acciones a realizar para que el servicio httpd funcione con las nuevas especificaciones, cree la página inicial que el servidor Apache mostrará en el caso de que éste funcione correctamente. El motivo de realizar este paso previo es que cuando se instala el servicio httpd, aun no estando completamente configurado, éste muestra una página inicial de prueba que se corresponde con el contenido del archivo `/usr/share/fedora-testpage/index.html`. Sigua los siguientes pasos para crear esta página inicial:

- Cree en el directorio `/var/www/html` el archivo `index.html`
- Edite este archivo e introduzca el siguiente código html

```
<html>
   <body>Enhorabuena, tu configuración es correcta</body>
</html>
```

- Obtenga los atributos de propietario, grupo propietario y permisos de acceso del archivo `/usr/share/fedora-testpage/index.html` y asígnele esos mismos atributos al archivo `/var/www/html`.

5.1. Cambio del puerto de servicio de Apache

El cambio del puerto de uso del servicio Apache para conexiones no seguras se realiza en tres pasos. En una configuración estándar el puerto utilizado es el 80 y se propone cambiarlo al puerto 3131. El primero paso consiste en cambiar la definición de este puerto en el fichero de configuración `/etc/httpd/conf/httpd.conf`. A continuación, se describe cómo realizar este cambio.

Por precaución se recomienda guardar una copia de este archivo. A continuación, con un editor de texto abra el archivo y localice la línea cuyo contenido es:

```
Listen 80
```

Una vez localizada la línea, modifíquela para que su nuevo contenido sea

```
Listen 3131
```

Una vez hecha la modificación, guarde los cambios realizados.

Para que los cambios que ha realizado en el fichero surjan efecto en el servicio httpd, primero detenga el servicio con la orden:

```
# systemctl stop httpd
```

A continuación, arránquelo con la orden:

```
# systemctl start httpd
```

Observará que se produce un mensaje que indica que el servicio no se ha podido arrancar debido a un error. Para obtener información del error producido, ejecute la orden:

```
# systemctl status httpd
```

En una de las líneas que reporta esta orden se indica que se ha producido un error por denegación de acceso. Pues bien, esta denegación de acceso la ha producido el SELinux, ya que en el servicio `httpd` es un servicio que está bajo su supervisión y sus reglas de control de acceso no permiten que este servicio utilice este puerto. Concretamente SELinux define el dominio httpd y las reglas que controlan el acceso a los recursos que forman parte de este dominio no permiten que se utilice el puerto 3131 para acceder a este servicio. Para obtener información sobre qué puertos y protocolos permite utilizar SELinux al proceso httpd, ejecute la orden:

```
# semanage port -l | grep http
```

Observará que el puerto 3131 no figura como puerto que puede utilizar el proceso `httpd`. Por ello, cuando el este proceso, al arrancar, intentó abrir una conexión con el puerto 3131, el SELinux no se lo permitió.

Para comprobar lo expuesto, puede modificar el modo de funcionamiento del SELinux, por ejemplo, poniéndolo en modo permisivo y volviendo a ordenar el arranque del servicio httpd. Utilice para ello la siguiente secuencia de órdenes.

```
# setenforce 0
# systemctl start httpd
```

Obteniendo el estado del servicio `httpd` se observará que ahora está funcionando. En este punto hay que matizar que el error se produjo, pero debido a que el modo de funcionamiento del SELinux es "`permisve`", el SELinux reporta el error y permite que el proceso httpd use el puerto. Pero el objetivo es que este cambio en la configuración del servicio funcione correctamente, ejecutándose el SELinux en modo "`enforced`". Seguidamente se muestra las acciones a realizar para conseguir este objetivo. Lo primero es volver a poner en modo "`enforced`" el modo de funcionamiento del SELinux. Para ello ejecute la orden:

```
# setenforced 1
```

Para configurar SELinux de tal manera que permita utilizar el puerto 3131 al proceso httpd, ejecute la siguiente orden:

```
# semanage port -a -t http_port_t -p tcp 3131
```

Una vez ejecutada esta orden, ejecute de nuevo la orden de inicio del servicio httpd y observará que ya no se genera un mensaje de error. Para confirmar que el servicio se está ejecutando correctamente puede volver a ejecutar la orden que ejecutó anteriormente para obtener información de estado del servicio. Observará que el servicio se está ejecutando y también que utiliza el puerto 3131.

Para verificar que el servicio se está ejecutando y utiliza como puerto de conexión no segura el puerto 3131, ejecute la orden:

```
# curl http://localhost:3131
```

Observará que la orden le devuelve por pantalla el contenido del archivo `/var/www/html/index.html`.

Sin embargo, si intenta usar el servicio Apache remotamente, no podrá. Esto es debido a que aún no se ha actualizado el servicio de cortafuegos para que permita al proceso `httpd` utilizar el puerto 3131. Puede comprobar la configuración actual ejecutando la orden:

```
# firewall-cmd --info-service=http
```

Para llevar a cabo esta actualización, lo primero que se debe hacer es eliminar el puerto 80 como puerto del servicio httpd. Esto se consigue mediante la orden:

```
# firewall-cmd --permanent --service=http --remove-port=80/tcp
```

La acción anterior no es estrictamente necesaria para lograr el objetivo perseguido. Esta acción se hace con el fin de cumplir con un principio de seguridad básico que consiste en permitir conexiones sólo en aquellos puertos que se utilicen.

Seguidamente, añadir el puerto 3131, usando el protocolo tcp, como puerto del servicio httpd:

```
# firewall-cmd --permanent --service=http --add-port=3131/tcp
```

Los cambios anteriores actualizan de manera permanente los ficheros de configuración del cortafuegos que tendrían efecto en el siguiente arranque que se produjera del servicio de cortafuegos. Para que tengan un efecto inmediato, se debe ejecutar la siguiente orden:

```
# firewall-cmd --reload
```

En este punto ya se han realizado todas las acciones requeridas para conseguir el primero de los tres cambios propuestos en la configuración del servicio `httpd.` Por tanto, ya se tendría configurado el servicio Apache para que pueda admitir conexiones remotas no seguras por el puerto 3131. Recuerde que, para poder realizar estas conexiones remotas desde el sistema anfitrión, debe configurar correctamente el redireccionamiento de puertos en la configuración de la máquina virtual en la que se ejecuta el servicio Apache. Tenga presente que en su configuración original de la regla de direccionamiento el puerto en el sistema invitado es el 80 y ahora debe ser el 3131.

5.2. Cambio del directorio de contenidos de Apache

En este punto se describe llevar a cabo a el segundo de los cambios propuestos en la configuración del servicio `httpd`. Hay que recordar que este cambio consistía en configurar este servicio para que en el directorio `/Apache/www` estén ubicados los archivos de contendidos del sitio web. Para realizar este cambio hay que reproducir la organización de los archivos de contenidos en la nueva ubicación, hay que asignarle las etiquetas SELinux apropiadas a los archivos de contenidos en la nueva configuración y especificar la nueva ubicación de los archivos de contenidos en el archivo de configuración del servicio httpd. Seguidamente se detalla cómo realizar estas acciones.

Primero se debe iniciar una nueva sesión del intérprete de órdenes `bash`, siendo `system_u` el usuario SELinux de la sesión. Este paso es necesario debido a que los archivos y directorios que se van a crear en la nueva ubicación deben poseer este valor en el atributo usuario en sus etiquetas SELinux. Para ello, ejecute la orden:

```
# runcon -u system_u /bin/bash
```

Compruebe que el atributo usuario SELinux que posee la nueva sesión del bash es system_ u. Para ello, ejecute la orden:

```
# id -Z
```

El segundo paso consiste en crear el directorio `/Apache` y reproducir en él la estructura de directorios existente en el directorio `/var/www`. También se deben asignar a los nuevos directorios creados en `/Apache` los mismos atributos de propietario, grupo propietario y permisos de acceso que tienen los directorios existentes en `/var/www`. Una vez realizado este paso, si ejecuta la orden:

```
# ls -Z /Apache
```

Observará que el atributo usuario y rol SELinux que poseen los contextos SELinux de los directorios existentes poseen los valores `system_u` y `object_r`. Sin embargo, el atributo tipo posee el valor `default_t`.

El tercer paso consiste en copiar el archivo `/var/www/html/index.html` en el directorio `/Apache/www/html/` y asígnele los mismos atributos de propietario, grupo propietario y permisos de acceso que posee el archivo `/var/www/html/index.html`.

En este punto se procede a cambiar la configuración del servicio `httpd` para que asuma la nueva ubicación de los archivos de contenidos. Para ello debe modificar el archivo de configuración `/etc/httpd/conf/httpd.conf` de la siguiente manera:

La línea que cuyo contenido es `DocumentRoot "/var/www/html"` actualizarla a `DocumentRoot "/Apache/www/html"`.

La línea que cuyo contenido es `<Directory "/var/www">` actualizara a `<Directory "/Apache/www">`.

La línea que cuyo contenido es `<Directory "/var/www/html">` actualizara a `<Directory "/Apache/www/html">`.

Para que los cambios tengan efecto en el servicio `httpd`, primero detenga el servicio con la orden:

```
# systemctl stop httpd
```

A continuación, arránquelo con la orden:

```
# systemctl start httpd
```

Si intentara acceder a la página índice del servidor Apache, que es la contenida en el archivo `/Apache/www/html/index.html`, por ejemplo, ejecutando la orden:

```
# curl http://localhost:3131
```

Observará que se no se descarga el contenido del archivo `/Apache/www/html/index.html`, en su lugar se descarga el contenido del archivo de test del servicio Apache `/usr/share/fedora-testpage/index.html`. Esto indica que se ha producido un error. Este error se debe que SELinux no permite que el proceso `httpd` acceda al archivo `/Apache/www/html/index.html`. Para comprobar que ésta es la causa del error, ponga en modo de funcionamiento del SELinux en modo "permisive":

```
# setenforce 0
```

Vuelva a ejecutar la orden:

```
# curl http://localhost:3131
```

Se observa que ahora sí se puede acceder a la página índice del servidor Apache. Pero el objetivo es que este cambio en la configuración del servicio `httpd` funcione ejecutándose el SELinux en modo "`enforcded`". Seguidamente se explican las acciones a realizar para lograr este objetivo.

Se debe actualizar el atributo tipo de los contextos SELinux del directorio `/Apache/www` y sus descendientes al valor requerido. Para realizar esta actualización se debe ejecutar la orden semanage, existiendo dos formas alternativas de uso. La primera de ellas se muestra a continuación:

```
#  semanage fcontext -a -e /var/www   /Apache/www
```

Esta orden establece que los atributos SELinux del directorio `/Apache/www` y sus descendientes deben ser los mismos que los que posee el directorio `/var/www` y sus descendientes.

La segunda alternativa consiste en ejecutar la siguiente secuencia de órdenes:

```
# semanage fcontext –a –t httpd_sys_content_t "/Apache/www"
# semanage fcontext –a –t httpd_sys_content_t "/Apache/www/html"
# semanage fcontext –a –t httpd_sys_script_exec_t "/Apache/www/cgi-bin"
# semanage fcontext –a –t httpd_sys_content_t "/Apache/www/html(/.*)?"
# semanage fcontext –a –t httpd_sys_script_exec_t "/Apache/www/cgi-bin(/.*)?"
```

Las tres primeras órdenes establecen el atributo tipo de las etiquetas SELinux de los directorios `/Apache/www`, `/Apache/www/html` y `/Apache/www/cgi-bin` a los valores especificados en la orden. Las dos siguientes hacen lo mismo, pero en todos los archivos ubicados en `/Apache/www/html` y `/Apache/www/cgi-bin` y sus descendientes.

Por último, independientemente de la alternativa elegida en la acción anterior, para que los cambios tengan efecto en el directorio /Apache/www y sus descendientes, se debe ejecutar la orden:

```
# restorecon -Rv /Apache/
```

Esta orden hace que se regeneren las etiquetas SELinux en el directorio /Apache y sus descendientes.

Una vez realizados estos pasos ya el servidor Apache debería funcionar de acuerdo con las nuevas especificaciones y ejecutándose el SELinux en modo "enforced". Para verificar el cumplimiento de este objetivo, lo primero es volver a poner en modo "enforced" el modo de funcionamiento del SELinux. Seguidamente acceda a la página índice. Este acceso lo puede hacer de dos maneras. La primera ejecutando, en el propio servidor Apache, la orden:

```
# curl http://localhost:3131
```

Esta orden debe producir la visualización del contenido del archivo `/Apache/www/html/index.html`. Para completar la verificación se debe acceder al servicio desde el sistema anfitrión mediante un navegador, debiéndose aparecer el contenido de archivo `/Apache/www/html/index.html`. Si ambas pruebas tienen éxito, entones ya se ha completado el segundo cambio en la configuración del servicio `httpd`.

5.3. Permitir que Apache utilice espacio de almacenamiento NFS

Existen un conjunto de variables booleanas SELinux que controlan si el servicio httpd puede hacer uso de otros servicios. La siguiente orden permite obtener este conjunto de variables:

```
# getsebool -a | grep '^httpd_'
```

El valor de la variable `httpd_use_nfs` establece si se permite o no se le permite al proceso `httpd` acceder a directorios o archivos que están almacenados remotamente y que son accesibles mediante el servicio NFS. Esta funcionalidad podría ser de utilidad cuando se requiera que el sitio web se ejecute en alta disponibilidad, implicando ello que el proceso `httpd` se pueda ejecutar en varios sistemas y que el espacio de almacenamiento que alberga a los archivos de contenido sea un espacio de almacenamiento compartido. Para comprobar el valor de esta variable, ejecute la orden:

```
# getsebool httpd_use_nfs
```

Si su valor es "off", entonces se requiere cambiar su valor a "on". Para ello se debe ejecutar la orden:

```
#  setsebool httpd_use_nfs on
```

La orden anterior cambia temporalmente el valor de la variable, de manera que en el siguiente reinicio del sistema ésta adquiriría su valor de "off". Si se quiere que el cambio sea permanente, entonces se debe utilizar la opción -P en la orden.

```
# setsebool -P httpd_use_nfs on
```

Si se utiliza esta opción, el sistema tardará más tiempo en ejecutarla. Esto es debido a que, con esta opción, se requiere modificar las reglas que implementan la política utilizada por SELinux.