**Parte I: Información, estado y dominios de SELinux.**

16

**SEGURIDAD INFORMÁTICA**

**Práctica 3**

**Jorge Andrés Galindo - 679155**

**Javier Aranda García - 679184**

Tras importar la máquina, se ha iniciado sesión con el usuario “u”, y se han obtenido los diferentes usuarios efectivos de los procesos en ejecución. De todos los procesos, se han obtenido los siguientes usuarios:

* root.
* rpc.
* dbus.
* rpcuser.
* 68.
* posfix.
* gdm.
* rtkit.
* u.

A continuación, se ha ejecutado el comando “ps auxZ”, para saber la información de seguridad de SELinux. La primera información a obtener fue qué usuarios de Linux estaban ejecutando procesos en dominios “unconfined\_t”, algo que se ha obtenido pasando mediante una tubería el comando anterior al comando “grep”, y sacando como resultado tan sólo al usuario “u”. Posteriormente, siguiendo el mismo procedimiento, se han obtenido los usuarios que se estaban ejecutando con usuario y rol “system\_u” y “system\_r”. Los usuarios obtenidos han sido “root”, “rtkit”, “gdm”, “posfix”, “dbus”, rpc” y “rpcuser”. Dada toda esta información, se ha deducido que algunos procesos están en dominios “unconfined\_t” ya que es el asignado a los procesos nuevos creados, y aquellos que no tienen este dominio es que han realizado una transición a otro distinto.

Seguidamente, se ha procedido a invocar el comando “passwd” en otra terminal para ver los datos de SELinux del proceso generado. El proceso generado tiene como usuario Linux “root” (lógico ya que se ejecuta con dicho usuario como usuario efectivo), usuario SELinux “unconfined\_u” y rol “unconfined\_r”. Esto es debido a que el usuario “u” lo ha ejecutado con un dominio determinado, y este ha comenzado otro proceso en otro dominio diferente, logrando cambiar de dominio gracias a que tiene permiso “entrypoint”.

Mediante la aplicación de gestión gráfica de SELinux, se han observado todos los dominios cargados y su estado en “Process Domain”. A continuación, se ha realizado un filtro para obtener tan sólo la lista de los permisivos (“semanage permissive -l”), obteniendo algunos como “nova\_api\_t”, “sblim\_reposd\_t”, “boinc\_t"…

**Parte II: Usuarios en SELinux.**

En este apartado de la práctica, en primer lugar, se ha accedido a una terminal como usuario “root”. En dicha terminal se ha creado una nueva cuenta de usuario “c” con el comando “adduser -Z user\_u c”. Este comando lo que ha realizado es crear un nuevo usuario llamado “c” y, además, le ha asignado el usuario SELinux “user\_u”. A continuación, gracias al comando “passwd c”, se le ha cambiado la contraseña al usuario asignándole “resu”. Mediante el comando “id”, se han obtenido los siguientes datos de contexto:

“unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023”

Estos datos indican el usuario, rol y dominio en el que se está ejecutando el proceso “id” en esta sesión.

Posteriormente, mediante el comando “semanage login -l”, se ha comprobado la asignación de la nueva cuenta creada “c” al usuario SELinux “user\_u”. La tabla indica que, como se ha indicado al crearlo, el usuario Linux “c” tiene asignado el usuario SELinux “user\_u”. Al observar la tabla, se puede apreciar que el usuario “u” con el que se ha iniciado sesión no aparece, esto es debido a que aparece en el caso “default” asignado al usuario “unconfined\_u”.

A continuación, se ha salido de la sesión inicial del usuario “u” para acceder con el nuevo usuario creado “c”. Se ha ejecutado en una consola de esta sesión el comando “id” y se ha obtenido la siguiente salida diferente a la anterior:

“user\_u:user\_r:user\_t:s0”

La salida es diferente ya que, aunque el usuario Linux se puede cambiar fácilmente mediante el comando “su”, el usuario SELinux, normalmente, es obligado a permanecer el mismo todo el tiempo. Por esto, la salida del apartado del anterior correspondía con el usuario SELinux del usuario Linux “u”, y en esta nueva salida, ya se muestra el usuario SELinux real de “c”, ya que se ha iniciado una nueva sesión.

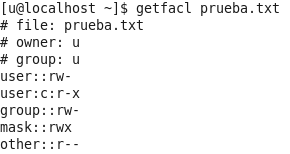
Se ha abierto una nueva terminar y se ha intentado acceder con el usuario “root” mediante el comando “su -“. La salida indica que no encuentra el comando “su”, algo que no ocurría para el usuario “u”. Esto es debido a que, para los dominios, usuarios, roles y tipos no confinados, SELinux prácticamente no interviene, y se aplican los permisos DAC de Unix. Así pues, el usuario “c”, al haberle sido asignado el usuario SELinux “user\_u”, se le aplican las políticas de SELinux que no le permiten ejecutar el comando.

**Parte III: Listas de Control de Accesos basados en estándar POSIX de Unix del modelo DAC**

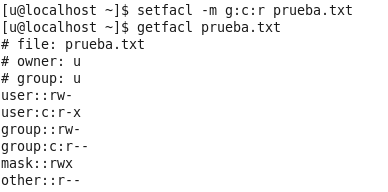
En este apartado, se van a analizar y utilizar las listas de control de accesos basadas en el estándar POSIX de Unix. En primer lugar, se ha extraído información de los comandos “getfacl” y “setfacl”, cuya función es manipular las listar nombradas anteriormente, permitiendo una asignación más fina y amplia de permisos a usuarios que sólo los ámbitos clásicos. Se ha creado un fichero en el “home” del usuario “u”, y se le han añadido los permisos de lectura y ejecución al usuario “c” mediante el comando “setfacl”. Además, haciendo uso del mismo comando, se ha permitido la lectura tan sólo al grupo “c”. Los comandos utilizados para la asignación de permisos son los siguientes:

* “setfacl -m u:c:rx prueba.txt”: cambia los permisos indicado con “-m” al usuario(“c”) a lectura y ejecución.
* “setfacl -m g:c:r prueba.txt”: sintaxis idéntica al anterior pero modificando al grupo “c” para solo lectura.

Tras realizar la asignación de permisos, se ha comprobado que se han asignado correctamente con el comando “getfacl prueba.txt”, obteniendo el siguiente resultado:



En la imagen se ve como el usuario “c” tiene permiso de lectura y ejecución sobre el archivo como se pretendía con la ejecución del comando.



Se ha añadido correctamente el permiso de lectura para el grupo “c” sobre el fichero “prueba.txt”.

**Parte IV: Reglas de control de accesos y acceso a dominios**

A continuación, se van estudiar distintas reglas de SELinux. Se han obtenido los dominios a los que se les permite acceder a ficheros tipo “shadow\_t” y clase “file” con permisos de escritura “write”. Para ello, se ha ejecutado el siguiente comando:

“sesearch -A -t shadow\_t -c file -p write”

Como resultado de la ejecución del comando se han obtenido los siguientes 9 dominios:

* groupadd\_t.
* passwd\_t.
* useradd\_t.
* sysadm\_passwd\_t.
* sandbox\_domain.
* updpwd\_t.
* files\_unconfined\_type.
* yppasswdd\_t
* mount\_t.

El permiso para acceder a fichero de tipo “shadow\_t” y poder escribirlos desde todos estos dominios significa que todos los procesos ejecutados y cuyo dominio sea éste, ya sea al crearlo o al cambiar de dominio mediante un “entrypoint”, podrán escribir y modificar ficheros de este tipo. Por ejemplo, un proceso creado mediante la ejecución del comando “passwd\_t”, que sirve para cambiar las contraseñas de un cierto usuario, tendrá un dominio “passwd\_t”, y como para realizar su función es necesario que tenga acceso al fichero “shadow” de contraseñas, deberá poder tener acceso a ficheros de tipo “shadow\_t” como el nombrado.

Finalmente, mediante el comando “setinfo –ruser\_r x”, se ha comprobado a que dominios puede acceder el rol “user\_t”. Como resultado de la ejecución del comando, se han obtenido los siguientes dominios (algunos de ellos):

* git\_session\_t.
* gpg\_t.
* ping\_t.
* ssh\_keygen\_t.
* gnome\_home\_t.