There are no translations available.

Siguiendo con los temas de Seguridad Informática, abordamos ahora la utilización de las listas de control de acceso (ACLs) como mecanismo de seguridad lógica.

Comenzamos definiendo el concepto de Seguridad lógica como conjunto de procedimientos utilizados para controlar el acceso lógico no autorizado al información, tanto si se intenta hacer estando almacenada dicha información como si está transmitiendo.

La Seguridad Lógica consiste, entonces, en la aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo.

Resumiendo podemos decir que la Seguridad Lógica debe garantizar que 'todo lo que no está permitido debe estar prohibido'.

Introducción

¿Qué son las Listas de Control de Acceso o ACLs?

Las ACL (Access Control Lists), como su nombre indica, son listas de condiciones que permiten designar permisos de acceso a cualquier elemento del sistema o directorio en

general. En función de estas condiciones, se concede o deniega el acceso a la modificación de las propiedades de estos elementos a los diferentes usuarios o procesos del sistema. Es, por tanto, un mecanismo de seguridad del sistema.En el caso de referirnos, por ejemplo, al sistema de archivos, las ACL proporcionan un nivel adicional de seguridad a los archivos, ya que extiende el clásico esquema de permisos.En el caso de referirnos, por ejemplo, a routers las ACL establecen condiciones que se aplican al tráfico que viaja a través de la interfaz del router. Es decir, las ACL informan al router de qué tipo de paquetes debe aceptar y cuáles rechazar.En el artículo vamos a referirnos a ambos tipos de ACLs. En el caso de los sistemas de archivos ext4, referidas a entornos GNU/Linux y en el caso de de los routers referidas a los dispositivos enrutadores de Cisco.

Sistemas de archivos con ACLs

Las ACLs permiten asignar permisos a usuarios o grupos concretos. Por ejemplo, se pueden dar ciertos permisos a dos usuarios sobre unos archivos sin necesidad de incluirlos en el mismo grupo.

Los archivos y directorios tienen conjuntos de permisos configurados para el propietario del archivo, el grupo al que pertenece el dueño del archivo y los otros usuarios del sistema. Sin embargo, estos permisos tienen sus limitaciones. Por ejemplo, no se pueden configurar diferentes permisos para usuarios diferentes. Para solucionar este problema se crearon las *Listas de Control de Acceso (Access Control Lists, ACLs)*
.

El concepto de ACL permite un control más preciso que los permisos del archivo por sí solos, al dar al propietario de un objeto la capacidad de conceder o denegar accesos usuario por usuario.

Este mecanismo está disponible en la mayoría de Unix (Solaris, AIX, HP-UX, etc.), mientras que en otros que no lo proporcionan por defecto, como GNU/Linux, pero puede instalarse como un paquete adicional (acl).

GNU/Linux soporta 2 tipos básicos de ACL:

- **ACL estándar**: ACL de control de directorios y ficheros. Contiene únicamente las entradas para los tipos propietario, grupo propietario y otros, que corresponden a los bits de permiso convencionales para archivos y directorios.
- **ACL extendida**: ACL que incluye más elementos. Una entrada máscara y puede contener al mismo tiempo distintas entradas de los tipos usuario nombrado y grupo nombrado.

Las ACLs se pueden configurar:

1. Por usuario
2. Por grupo
3. A través de la máscara de permisos

Requerimientos del sistema

En general, para trabajar con ACL debemos comprobar que:

- El núcleo debe soportar y estar compilado para soportar atributos extendidos y ACL.El sistema de archivos tiene que montarse con atributos extendidos y ACL.
- Se tienen que instalar las utilidades de espacio de usuario para establecer el ACL: paquete acl.

Respecto al primer punto y en el caso de Ubuntu, la utilización de ACL está activada en el Kernel.

Respecto al segundo punto, cuando se va a trabajar con ACL en un sistema de archivos determinado (/home) en el archivo /etc/fstab hay que indicarlo de la forma siguiente:

```
# cat /etc/fstab|grep home  /dev/sda2 /home ext4 defaults,acl 0  # cat /etc/fstab|grep home# /dev/sda2 /home ext4 defaults,acl 0  /dev/sda2
```

Para no tener que hacer un reboot del sistema, simplemente podemos remontar la partición para que use acl ejecutando la orden siguiente:

```
# mount -o remount -o acl /dev/sda2 /home#
```

Ahora podemos ya utilizar ACLs en nuestro sistema. Respecto al tercer punto, instalamos el paquete acl (puede que esté instalado por defecto).

```
# sudo apt-get install acl#
```

En este paquete están las utilidades que nos permitirán establecer y manejar las ACLs: chacl,

getfacl y setfacl.

También se puede instalar el paquete acl desde el Centro de Software de Ubuntu.

Utilización de ACLs en el sistema de archivos

En este punto vamos conocer la sintaxis y el funcionamiento de las ACLs en Ubuntu. También aprenderemos a utilizar herramientas de gestión asociadas a las ACLs siempre dentro del ámbito del sistema de archivos.

También comprobaremos la ejecución y analizaremos la salida generada por las ACLs.

Sintaxis

Una ACL se representa por tres campos separados por ":".

[Tipo]:[Calificador]:ListaPermisos[,...]

Tipo:

- "u" Usuario
- "g" Grupo
- "o" Otro
- "m" Máscara

Calificador:

- UID (ID numérico de usuario)
- GID (ID numérico de grupo)
- Vacío (Asume UID,GID del creador)

ListaPermisos:

El tercer campo es el de acceso y puede ser representado de 2 maneras:

- Cadena estándar rwx (las cadenas se pueden remplazar por "-" si no queremos dar acceso de ese tipo)
- Cadena simbólica + ^

Utilidades

El paquete acl contiene las órdenes:

- **chacl**: Permite cambiar, examinar o eliminar ACL.
- **getfacl**: Devuelve la lista de control de acceso a un fichero o directorio.
- **setfacl**: Asigna, modifica o elimina una lista de control de acceso de un archivo o directorio.

Funcionamiento de las ACL

1. Mostrar ACL:

Nos situamos en el directorio ~/datos en el que tenemos permisos de escritura. Recordar que el carácter ~ equivale al directorio home del usuario. Por ejemplo, para el usuario *admin* sería equivalente a
/home/admin/
.

La ACL para un directorio al que no se le ha asignado una ACL explícitamente, toma los permisos de umask:

```
$ umask
0022
```

Recordar que el significado de 0022, como máscara general del sistema, es el siguiente:

- Los directorios de crean con permisos: rwx r-x r-x

- Los archivos se crean con permisos: rw- r-- r--

En nuestro caso y para ver la ACL por defecto (sin asignar ninguna) del directorio (actual '.') utilizamos la orden siguiente:

```
 $ getfacl .
# file: .
# owner: admin
# group: admin
user::rwx
group::r-x
other::r-x
```

Creamos en el directorio actual un par de ficheros 'archivo1' y 'archivo2' con la orden touch.

```
 $ ls -l arch*
-rw-r--r-- 1 admin admin 0 2012-09-14 11:56 archivo1
-rw-r--r-- 1 admin  admin 0 2012-09-14 11:56 archivo2
```

La acl por defecto para el fichero archivo1 es:

```
 $ getfacl archivo1
# file: archivo1
# owner: admin
# group: admin
user::rw-
group::r--
other::r--
```

2. Establecer ACL

Importante: Antes de ejecutar la orden comprobar los permisos existentes para comparar después de establecer la ACL.

Hay tres maneras de establecer un ACL:

1. Usando el comando **setfacl** que sobrescribe cualquier ACL anterior.
2. Usando el comando **setfacl** con la opción **-m** (modifica ACL).
3. Usando **chacl** para modificar una ACL existente.

Ejemplos:

En el directorio ~/datos y sobre el fichero 'archivo1' queremos dar permisos de lectura y escritura al usuario prueba que deberemos crear previamente.

```
$ setfacl -m user:prueba:rw- ~/datos/archivo1
```

Ahora queremos que todos los archivos o directorios creados dentro de un directorio 'archivos' en
~/datos
puedan ser leídos y modificados por el usuario
prueba
. La orden es la siguiente:

```
$ setfacl -d -m u:prueba:rw- ~/datos/archivos/$
```

Es importante el orden en el que aparecen los argumentos. En concreto las opciones -d y -m deben aparecer en este orden. Y, en general, la opción -m debe estar siempre junto a
[Tipo]:[Calificador]:ListaPermisos
.

Algunas opciones de setfacl son:

Opción	Descripción
-R	Cambia permisos a archivos y directorios de forma descendente a partir de un directorio dado.
-d	Asigna los permisos por defecto.

-b

Borra todos los permisos adicionales, conservando únicamente los básicos de UGO.

-k

Borra los permisos por defecto.

-m

Modifica los permisos agregando/cambiando por los nuevos valores.

3. Ver las ACL

La orden getfacl obtiene la lista de control de acceso del archivo o directorio dado. Podemos consultar el estatus del fichero archivo1 después de asignar una nueva ACL de la forma:

```
 $ getfacl ~/datos/archivo1
# file: datos/archivo1
# owner: admin
# group: admin
user::rw-
user:prueba:rw-
group::r--
mask::rw-
other::r--
```

Como podemos observar, *prueba* tiene permisos de lectura y escritura sobre el fichero archivo1

.

Algunas opciones de getfacl son:

Opción	Descripción
-R	Cambia permisos a archivos y directorios de forma descendente a partir de un directorio dado.
-d	Muestra los permisos por defecto.

4. Modificar las ACL

La orden chacl modifica una ACL existente.

El formato es algo mas complicado. Por ejemplo, añadimos al usuario *prueba* como alguien que puede leer el fichero
archivo2
:

```
$ chacl u::rw-,g::r--,o::---,u:prueba:r--,m::rw archivo2
```

La parte u::rw,g::r--,o::--- es la ACL existente y la parte u:prueba:r--,m::rw- especifica el nuevo usuario que quiero añadir a dicha acl y la máscara de derechos efectivos cuando se añada. Como se puede observar la sintaxis es diferente.

La máscara de derechos efectivos es la unión de todos los permisos ya existentes de UGO (*Us*

uario, Grupo, Otros
) para un fichero o directorio. Se obtiene al añadir un usuario a la ACL.

Para comprobar que han sido añadidos estos permisos a la ACL usaremos:

```
 $ getfacl archivo2
# file: archivo2
# owner: admin
# group: admin
user::rw-
user:prueba:r--
group::r--
mask::rw-
other::---$
```

Es importante saber que los comandos cp y mv copian o mueven cualquier ACL asociada a ficheros y directorios.

5. ACL por defecto

Las ACLs por defecto permiten indicar cuál es el juego de ACLs que queremos que se aplique automáticamente a los nuevos ficheros y directorios que se creen dentro de un directorio dado. Se dice en estos casos que los permisos se heredan desde el directorio padre.

La sintaxis es idéntica a la de una ACL normal, pero se debe usar además la opción -d.

Los permisos de una ACL por defecto de un directorio se transfieren a los archivos y subdirectorios de dicho directorio de dos formas:

- El subdirectorio hereda la ACL por defecto del directorio padre tanto como ACL por defecto como de ACL de acceso.
- El archivo hereda la ACL por defecto como ACL de acceso.

Si el directorio padre no dispone de una ACL por defecto, los bits de permiso definidos en umas k se

restan de los permisos al pasar por parámetro
mode
, y el resultado se asigna al nuevo objeto.

Si existe una ACL por defecto para el directorio padre, los bits de permiso asignados al nuevo objeto se corresponden con la porción coincidente de los permisos del parámetro mode y con aquellos que se definen en la ACL por defecto. El comando
umask
se descarta en este caso.

Ejemplo:

Creamos un directorio llamado 'midirectorio' y comprobamos que tiene estas características:

```
 # file: midirectorio
# owner: admin
# group: admin
user::rwx
group::r-x
other::---
```

Y sobre midirectorio ejecutamos la orden siguiente:

```
 $setfacl -m user:prueba:rwx,group:mascotas:rwx midirectorio
```

Añadir una ACL por defecto al directorio 'midirectorio' con:

```
 $setfacl -d -m group:mascotas:r-x midirectorio
```

La opción -d del comando setfacl hace las siguientes modificaciones (opción -m) en la ACL por defecto. El resultado es el siguiente:

```
  $getfacl midirectorio
# file: midirectorio
# owner: admin
# group: admin
user::rwx
user:prueba:rwx
group::r-x
group:mascotas:rwx
```

```
mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:mascotas:r-x
default:mask::r-x
default:other::---
```

El comando getfacl muestra tanto la ACL de acceso como la ACL por defecto.

La ACL por defecto se compone de todas las líneas que comienzan por default.

Aunque se ejecutó solamente el comando setfacl con una entrada para el grupo *mascotas* para obtener la ACL por defecto, el comando
setfacl
ha copiado automáticamente las entradas restantes de la ACL de acceso para crear una ACL por defecto válida.

Las ACL por defecto únicamente intervienen cuando se crean nuevos elementos,
que solamente heredan permisos de la ACL por defecto de su directorio padre.

La entrada *mask* define los permisos de acceso máximos que tienen validez para todas las entradas de la clase *group*.

Si ahora se crea un subdirectorio 'misubdir' éste heredará la ACL por defecto.

```
 mkdir midirectorio/misubdir
getfacl midirectorio/misubdir
# file: midirectorio/misubdir
# owner: admin
# group: admin
user::rwx
user:prueba:rwx
group::r-x
group:mascotas:rwx
```

mask::rwx
other::---
default:user::rwx
default:group::r-x
default:group:mascotas:r-x
default:mask::r-x
default:other::--- **6. ACL extendida**

Una ACL *extendida (extended)* contiene además una entrada *mask* (máscara) y puede incluir varias entradas del tipo
named user
(usuario identificado por el nombre) y
named group
(grupo identificado por el nombre).

Tipo	Formato en texto
dueño	
	user::rwx
nombre usuario	

user:name:rwx

grupo del dueño

group::rwx

nombre grupo

group:name:rwx

mascara

mask::rwx

otros

other::rwx

Los permisos definidos en las entradas *dueño y otros* siempre tienen vigencia.

Excepto la entrada mascara, el resto de entradas (*nombre usuario, grupo del dueño y nombre grupo*) pueden estar activadas o bien enmascaradas. Si se han definido permisos tanto en las entradas mencionadas en primer lugar como en la máscara, tendrán validez. Los permisos que sólo han sido definidos en la máscara o en la propia entrada, no tienen validez.

Si en la primera columna de la salida de la orden ls -l aparece un signo + hace referencia a una ACL *extendida*.

```
  $ ls -dl dir1
drwxrwx---+ 2 admin admin 4096 oct 14 19:47 dir1
```

Ejemplos

1. Vamos a explicar qué hace la siguiente orden y cómo quedaría el esquema de permisos después de su ejecución:

```
$ getfacl fich1 | setfacl --set-file=- fich2
```

La orden obtiene los permisos del fichero 'fich1' y los asigna (utilizando la redirección de entrada '|') al archivo 'fich2'. Estamos utilizando el argumento – set-file con el '-' que indica que se está tomando como permisos los que le llegan por la redirección '|'. En resumen, después de la ejecución de la orden fich1 y fich2 tendrán los mismos permisos, incluidas ACLs establecidas en fich1.

1. Eliminamos los permisos predeterminados en la lista de control de acceso de un directorio. Para ello utilizamos la opción -k (--remove-default):

```
$setfacl -k ~/dir1
```

1. Si queremos eliminar todos los permisos en la lista de control de acceso de un directorio, volviendo a la situación inicial utilizamos la opción -b (--remove-all):

```
$setfacl -b ~/dir1
```

1. Vamos a explicar qué hace la siguiente orden y cómo quedaría el esquema de permisos después de su ejecución:

```
$setfacl -R -b ~/dir1
```

1. Veamos qué hace la orden siguiente:

```
$ setfacl -R -d -m group:grupo1:rwx "./dir/"
```

La orden actúa sobre el directorio 'dir' que es un subdirectorio del directorio actual. Establece como ACL para usuarios del grupo 'grupo1' sobre 'dir' todos los permisos (rwx), lo hace de forma recursiva (-R) sobre toda la estructura que cuelgue de 'dir' y establece esta ACL por defecto (-d).

1. Escribimos ahora la orden que guarde en un archivo las ACL de todo el directorio y su contenido en un archivo dentro del propio directorio, llamado dir.acl.

Primero comprobamos los permisos de 'dir':

```
$ getfacl dir/
# file: dir/
# owner: admin
# group: admin
user::rwx
```

```
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:grupo1:rwx
default:mask::r-x
default:other::r-x
```

Ejecutamos ahora la orden que guarda las ACL:

```
$ getfacl -R dir > dir/dir.acl
```

Ahora podríamos asignar estas ACL a cualquier otro directorio que quisiéramos.

Utilización de ACLs en routers

Definición

En el ámbito de los dispositivos routers, las ACLs son listas de condiciones que se aplican al tráfico que viaja a través de la interfaz del router.

Las ACL indican al router qué tipo de paquetes aceptar o rechazar en base a las condiciones establecidas en ellas y que permiten la administración del tráfico y aseguran el acceso, bajo esas condiciones, hacia y desde una red.

La aceptación y rechazo se pueden basar en la dirección origen, dirección destino, protocolo de capa superior y números de puerto.

Por lo tanto, una ACL es un grupo de sentencias que define cómo se procesan los paquetes que:

- Entran a las interfaces de entrada
- Se reenvían a través del router
- Salen de las interfaces de salida del router

En principio si las ACL no están configuradas en el router, todos los paquetes que pasen a través del router tendrán acceso a todas las partes de la red.

Es posible crear ACL en protocolos de red enrutados, como el Protocolo de Internet (IP) y el Intercambio de paquetes de internetwork (IPX), entre otros. Se debe definir una ACL para cada protocolo enrutado habilitado en la interfaz.

Además, se necesita crear una ACL por separado para cada dirección, una para el tráfico entrante y otra para el saliente.

Como hemos comentado, las ACL se definen según el protocolo, la dirección o el puerto. Por ejemplo, si el router tiene dos interfaces configuradas para IP, IPX y AppleTalk, se necesitan 12 ACLs separadas. Una ACL por cada protocolo, multiplicada por dos por dirección entrante y saliente, multiplicada por dos por el número de interfaces.

Se puede configurar una ACL por protocolo, por dirección y por interfaz.

- Una ACL por protocolo: para controlar el flujo de tráfico de una interfaz, se debe definir una ACL para cada protocolo habilitado en la interfaz.
- Una ACL por dirección: las ACL controlan el tráfico en una dirección a la vez de una interfaz. Deben crearse dos ACL por separado para controlar el tráfico entrante y saliente.
- Una ACL por interfaz: las ACL controlan el tráfico para una interfaz, por ejemplo, Fast Ethernet 0/0.

Las ACL no actúan sobre paquetes que se originan en el mismo router. Las ACL se configuran para ser aplicadas al tráfico entrante o saliente.

- ACL de entrada: los paquetes entrantes se procesan antes de ser enrutados a la interfaz de salida.
- ACL de salida: los paquetes entrantes se enrutan a la interfaz de salida y luego son procesados a través de la ACL de salida.

Objetivos de las ACL

En resumen, los objetivos que se persiguen con la creación de ACL son:

- Limitar el tráfico de red y mejorar el rendimiento de la red. Al restringir el tráfico de vídeo, por ejemplo, las ACL pueden reducir ampliamente la carga de la red y en consecuencia mejorar el rendimiento de la misma.
- Controlar el flujo del tráfico. Las ACL pueden restringir el envío de las actualizaciones de enrutamiento. Si no se necesitan actualizaciones debido a las condiciones de la red, se preserva el ancho de banda.
- Proporcionar un nivel básico de seguridad para el acceso a la red. Por ejemplo, las ACL pueden permitir que un host acceda a una parte de la red y evitar que otro acceda a la misma área. Por ejemplo, el host-1 se le permite el acceso a la red de
- Producción, y al host-2 se le niega el acceso a esa red.
- Establecer qué tipo de tráfico se envía o se bloquea en las interfaces del router. Por ejemplo, permitir que se envíe el tráfico relativo al correo electrónico, y se bloquea el tráfico de ftp.
- Otorgar o denegar permiso a los usuarios para acceder a ciertos tipos de archivos, tales como FTP o HTTP.

Funcionamiento de las ACL

Para explicar el funcionamiento utilizaremos el software Cisco IOS.

El orden de las sentencias ACL es importante .

- Cuando el router está decidiendo si se envía o bloquea un paquete, el IOS prueba el paquete, verifica si cumple o no cada sentencia de condición, en el orden en que se crearon las sentencias .
- Una vez que se verifica que existe una coincidencia, no se siguen verificando otras sentencias de condición .

Por lo tanto, Cisco IOS verifica si los paquetes cumplen cada sentencia de condición de arriba hacia abajo, en orden. Cuando se encuentra una coincidencia, se ejecuta la acción de aceptar o rechazar y ya no se continua comprobando otras ACL.

Por ejemplo, si una ACL permite todo el tráfico y está ubicada en la parte superior de la lista, ya no se verifica ninguna sentencia que esté por debajo.

Si no hay coincidencia con ninguna de las ACL existentes en el extremo de la lista se coloca por defecto una sentencia implícita **deny any** (denegar cualquiera). Y, aunque la línea deny any no sea visible sí que está ahí y no permitirá que ningún paquete que no coincida con alguna de las ACL anteriores sea aceptado. Se puede añadir de forma explícita por aquello de 'verla' escrita y tener esa tranquilidad.

Veamos el proceso completo:

1. Cuando entra una trama a través de una interfaz, el router verifica si la dirección de capa 2 (MAC) concuerda o si es una trama de broadcast.
2. Si se acepta la dirección de la trama, la información de la trama se elimina y el router busca una ACL en la interfaz entrante.
3. Si existe una ACL se comprueba si el paquete cumple las condiciones de la lista.
4. Si el paquete cumple las condiciones, se ejecuta la acción de aceptar o rechazar el paquete.
5. Si se acepta el paquete en la interfaz, se compara con las entradas de la tabla de enrutamiento para determinar la interfaz destino y conmutarlo a aquella interfaz. Luego el router verifica si la interfaz destino tiene una ACL.
6. Si existe una ACL, se compara el paquete con las sentencias de la lista y si el paquete concuerda con una sentencia, se acepta o rechaza el paquete según se indique.
7. Si no hay ACL o se acepta el paquete, el paquete se encapsula en el nuevo protocolo de capa 2 y se envía por la interfaz hacia el dispositivo siguiente.

Creación de ACL

Utilizamos la herramienta de simulación Packet Tracer y una topología de red muy sencilla, formada por un router, dos switch y 2PCs, cada uno de ellos en una subred.

Trabajaremos desde el modo de configuración global: (config)#

Hay dos tipos de ACL y utilizan una numeración para identificarse:

- ACL estándar: del 1 al 99
- ACL extendida: del 100 al 199

ACLs estándar: sintaxis

Las ACL estándar en un router Cisco siempre se crean primero y luego se asignan a una interfaz.

Tienen la configuración siguiente:

Router(config)# access-list numACL permit|deny origen [wild-mask]

El comando de configuración global access-list define una ACL estándar con un número entre 1 y 99.

Se aplican a los interfaces con:

Router (config-if)# ip access-group numACL in|out

- **In**: tráfico a filtrar que ENTRA por la interfaz del router
- **out** : tráfico a filtrar que SALE por la interfaz del router.
- **wild-mask**: indica con 0 el bit a evaluar y con 1 indica que el bit correspondiente se ignora. Por ejemplo, si queremos indicar un único host 192.168.1.1 especifico: 192.168.1.1 con wild-mask 0.0.0.0 y si queremos especificar toda la red clase C correspondiente lo hacemos con 192.168.1.0 y wild-mask 0.0.0.255.

Para la creación de ACL estándar en importante:

- Seleccionar y ordenar lógicamente las ACL.
- Seleccionar los protocolos IP que se deben verificar.
- Aplicar ACL a interfaces para el tráfico entrante y saliente.
- Asignar un número exclusivo para cada ACL.

Ejemplo 1

Supongamos que queremos crear en un Router0 una ACL con el número 1 (numACL) que deniegue el host 192.168.1.2. Desde configuración global:

```
Router0(config)# access-list 1 deny 192.168.1.2  0.0.0.0
```

Si queremos eliminar una ACL:

```
Router0(config)# no access-list 
```

Para mostrar las ACL:

```
Router0# show access-list
Standard IP access list 1
deny host 192.168.1.2
permit any
```

Recordar que para salir del modo de configuración global (config) hay que escribir 'exit'.

Ahora hay que utilizar el comando de configuración de interfaz para seleccionar una interfaz a la que aplicarle la ACL:

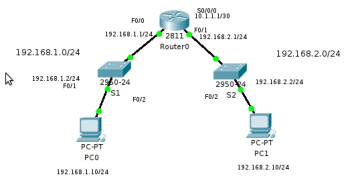
```
Router0(config)# interface FastEthernet 0/0
```

Por último utilizamos el comando de configuración de interfaz ip access-group para activar la ACL actual en la interfaz como filtro de salida:

```
Router0(config-if)# ip access-group 1 out
```

Ejemplo 2

Tenemos la siguiente topología de red.



Vamos a definir una ACL estándar que permita el trafico de salida de la red 192.168.1.0/24.

La primera cuestión que se plantea es ¿dónde instalar la ACL? ¿en qué router? ¿en qué interfaz de ese router?.

En este caso no habría problema porque solo tenemos un router, el Router0. Pero la regla siempre es **instalar la ACL lo más cerca posible del destino**.

```
Router0#configure terminal
Router0(config)#access-list 1 permit  192.168.1.0    0.0.0.255
Router0(config)#interface S0/0/0
Router0(config-if)#ip access-group 1 out
```

Ahora borramos la ACL anterior y vamos a definir una ACL estándar que deniegue un host concreto.

```
Router0(config)#no access-list 1
Router0(config)#access-list 1 deny    192.168.1.10   0.0.0.0     Router0(config)
#access-list 1 permit  192.168.1.0    0.0.0.255
Router0(config)#interface S0/0/0
Router0(config-if)#ip access-group 1 out
```

ACLs extendidas

Las ACL extendidas filtran paquetes IP según:

- Direcciones IP de origen y destino
- Puertos TCP y UDP de origen y destino
- Tipo de protocolo (IP, ICMP, UDP, TCP o número de puerto de protocolo).

Las ACLs extendidas usan un número dentro del intervalo del 100 al 199. Al final de la sentencia de la ACL extendida se puede especificar, opcionalmente, el número de puerto de protocolo TCP o UDP para el que se aplica la sentencia:

- 20 y 21: datos y programa FTP
- 23: Telnet
- 25: SMTP
- 53: DNS
- 69: TFTP

Definir ACL extendida, sintaxis: Router(config)# access-list numACL {permit|deny}

protocolo fuente
[mascara-fuente destino mascara-destino operador operando] [established]

- **numACL**: Identifica número de lista de acceso utilizando un número dentro del intervalo 100-199
- **protocolo**: IP, TCP, UDP, ICMP, GRE, IGRP
- **fuente** | **destino**: Identificadores de direcciones origen y destino
- **mascara-fuente** | **mascara-destino**: Máscaras de wildcard
- **operador**: lt, gt, eq, neq
- **operando**: número de puerto
- **established**: permite que pase el tráfico TCP si el paquete utiliza una conexión establecida.
- Respecto a los protocolos:
- Sólo se puede especificar una ACL por protocolo y por interfaz.
- Si ACL es entrante, se comprueba al recibir el paquete.
- Si ACL es saliente, se comprueba después de recibir y enrutar el paquete a la interfaz saliente.
- Se puede nombrar o numerar un protocolo IP.

Asociar ACL a interfaz, sintaxis:

```
Router(config-if)# ip access-group num_ACL {in | out}
```

Ejemplo 1

En el esquema anterior, denegar FTP entre las subredes y permitir todo lo demás.

```
Router0(config)# access-list 101 deny tcp 192.168.2.0  0.0.0.255 192.168.1.0  0.0.0.255 eq 21
Router0(config)# access-list 101 deny tcp 192.168.2.0  0.0.0.255 192.168.1.0 0.0.0.255 eq 20
Router(config)# access-list 101 permit ip any any
Router(config)# interface F0/1
Router0(config-if)#ip access-group 101 in
```

Ejemplo 2

En el esquema anterior, denegar solo telnet a la subred 192.168.1.0.

```
Router0(config)# access-list 101 deny tcp 192.168.1.0  0.0.0.255  any eq 23
```

```
 Router(config)# access-list 101 permit ip any any
Router(config)# interface F0/0
 Router0(config-if)#ip access-group 101 out
```

Ubicación de las ACLs

Es muy importante el lugar donde se ubique una ACL ya que influye en la reducción del tráfico innecesario.

El tráfico que será denegado en un destino remoto no debe usar los recursos de la red en el camino hacia ese destino.

La regla es colocar las:

- ACL estándar lo más cerca posible del destino (no especifican direcciones destino).
- ACL extendidas lo más cerca posible del origen del tráfico denegado. Así el tráfico no deseado se filtra sin atravesar la infraestrucra de red

Conclusión

Hemos visto dos formas diferentes de definir y utilizar las ACL. Como mecanismo para extender/reducir los privilegios de archivos y directorios en un sistema de archivos y como mecanismo de control del trafico entrante y saliente en dispositivos router. Pero en ambos casos el objetivo es la separación de privilegios y así establecer los permisos adecuados en cada caso particular.

Pero existe otro ámbito de utilización de las listas de control de acceso también muy interesante, que es en los servidores proxys. Servidores proxy como Squid, entre otros, disponen de esta herramienta que permite determinar qué equipos accederán a Internet a través del proxy y cuáles no.

En cualquiera de estos casos de aplicación de las listas de control de acceso comprobamos el gran papel que desempeñan como media de seguridad lógica, ya que su cometido siempre es

controlar el acceso a los recursos o activos del sistema.