Access Control Lists

Antes de Imprimir este documento
considere si es necesario
Ayudemos al Ambiente !!!!

Universidad San Carlos de Guatemala

DANILO ESCOBAR

Access Control Lists

Access Control Lists (ACLs)

Las listas de control de acceso, mejor conocidas como *access control lists* (ACLs), son una herramienta que permite identificar o marcar un flujo de datos acorde a ciertos criterios a manera de realizar una operación especial sobre él, siendo por esto utilizadas para control de acceso, calidad de servicio, enrutamiento basado en políticas, traducción de direcciones, etc.,

Consisten en una lista de sentencias que pueden ser permitir *(permit)* o denegar *(deny)* marcando o ignorando el tráfico que se les indique, aunque su efecto siempre dependerá de cómo y dónde estas sean aplicadas.

Las listas son examinadas sentencia por sentencia en orden secuencial deteniéndose la operación al hallar la primera coincidencia, razón por la cual las ACLs deben de diseñarse con cuidado, colocando las sentencias más específicas al principio para que estas pueden llegar a ser evaluadas.

La configuración de estas listas está separada de su implementación, aunque no hay ningún mecanismo que impida aplicar ACLs inexistentes, en cuyo caso se permitirá (o marcará) todo el tráfico que sea comparado con ella, hasta que esta sea creada y sentencias sean agregadas.

No obstante, se recomienda diseñar y configurar las listas de control de acceso antes de su implementación debido al comportamiento que estas presentan.

Una lista vacía (o inexistente) permitirá todo el tráfico de la manera descrita anteriormente, sin embargo, al ingresar la primera sentencia dentro de la misma se creará automáticamente otra al final de la lista, implícita e invisible, encargada de desestimar toda aquella información que no haya encontrado una coincidencia en las sentencias previas.

Dicha sentencia, es comúnmente conocida como “denegar todo” y siempre se encuentra al final de toda lista de control, razón por la cual siempre debe incluirse por lo menos una sentencia “permitir” cuando se pretende utilizar ACLs para regular tráfico.

Existen varios tipos de listas de control de acceso:

- Estándares
- Extendidas
- Reflexivas
- Basadas en el tiempo
- Etc.,

Al crear una lista de control se recomienda utilizar y apegarse a una convención, agregar las observaciones pertinentes y recordar al implementarlas que los nombres son sensibles a minúsculas y mayúsculas (*Case sensitive*).

Access Control Lists

Listas de control de acceso estándares

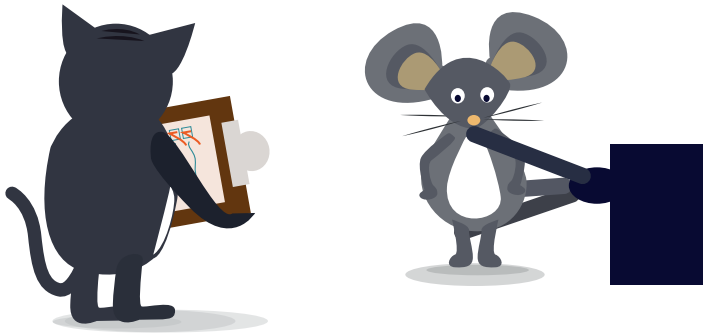
Son las listas de control más simples, utilizan como único parámetro de comparación el origen del tráfico empleando *Wildcard Masks* (Introducidas en la sección de OSPF) para seleccionar rangos específicos de direcciones e impactan muy poco al procesador.

En orden de presentar un ejemplo, se creará una lista de control de acceso estándar destinada a identificar el tráfico proveniente de la red 192.168.10.0/24 y que ignore todo lo demás, misma que será nombrada como “Técnicos”.

```
Router(config)# ip access-list standard Tecnicos
Router(config-std-nacl)# remark [> Esta lista identifica a los
tecnicos. <]
Router(config-std-nacl)# permit 192.168.10.0 ?
A.B.C.D  Wildcard bits
<cr>

Router(config-std-nacl)# permit 192.168.10.0 0.0.0.255
Router(config-std-nacl)# deny any
```

Nótese el uso del comando *remark* y la palabra clave *any*. *Remark* nos permite agregar una observación a la lista de control, mientras que *any* es un atajo para seleccionar todas las direcciones posibles y es el equivalente de utilizar la instrucción deny 0.0.0.0 255.255.255.255.

Además, se incluyó la sentencia *deny any* aunque no era necesario, ya que está implícita al final de toda lista, debido a que facilita la resolución de problemas y permite ver la cantidad de paquetes que han llegado a esta instancia al ser ahora visible, por lo que es una buena práctica.



Access Control Lists

Listas de control de acceso extendidas

Son mucho más granulares que las listas estándares, permiten seleccionar tanto el origen como el destino del tráfico, protocolos específicos y números de puerto.

A continuación se muestran los protocolos que pueden ser evaluados con una lista extendida. Durante el resto de esta discusión se tratará exclusivamente con TCP, UDP e IP. Donde la palabra clave "IP" abarca todos los protocolos disponibles en este tipo de lista.

```
Router(config)# ip access-list extended EJEMPLO
Router(config-ext-nacl)# permit ?
  <0-255>  An IP protocol number

  ahp    Authentication Header Protocol
  eigrp  Cisco's EIGRP routing protocol
  esp    Encapsulation Security Payload
  gre    Cisco's GRE tunneling
  icmp   Internet Control Message Protocol
  igmp   Internet Gateway Message Protocol
  ip     Any Internet Protocol
  ipinip IP in IP tunneling
  nos    KA9Q NOS compatible IP over IP tunneling
  ospf   OSPF routing protocol
  pcp    Payload Compression Protocol
  pim    Protocol Independent Multicast
  tcp    Transmission Control Protocol
  udp    User Datagram Protocol
```

A manera de ejemplo supóngase que se pretende identificar el tráfico que se origina en el *host* con la dirección IP 192.168.1.1/24 con destino al servidor web 192.168.2.1/24 (escuchando en el puerto 80) e ignorar el resto de la transmisión, podría utilizarse una lista extendida de la siguiente manera.

```
Router(config)# ip access-list extended EJEMPLO2
Router(config-ext-nacl)# permit tcp host 192.168.1.1 host
192.168.2.1 eq 80
Router(config-ext-nacl)# deny ip any any
```

En esta ocasión se ha utilizado la palabra clave "*host*", que identifica una dirección específica y es el equivalente de utilizar una *wildcard* 0.0.0.0. La última parte de la primera sentencia "eq 80" significa que se seleccionará el tráfico destinado al host 192.168.2.1 cuyo puerto de destino sea equivalente al puerto 80 y "*deny ip any any*" ignorará cualquier otro protocolo dirigido desde cualquier origen hacia cualquier destino.

Cuando se emplean listas de control de acceso extendidas se debe tener cuidado de no seleccionar el puerto utilizado por la parte que origina el tráfico en vez de aquella a la que este está destinado.

Regresando al ejemplo anterior, si se hubiera ingresado por error la siguiente sentencia, difícilmente se hubiera encontrado una coincidencia debido a que las transmisiones regresan al origen en un puerto generado aleatoriamente.

```
Router(config-ext-nacl)# permit tcp host 192.168.1.1 eq 80 host 192.168.2.1
```

Access Control Lists

Listas de control de acceso aplicadas para regular tráfico en una interfaz

Las listas de control de acceso son capaces de regular el tráfico tanto en interfaces físicas o virtuales (líneas VTY). En estos casos la aplicación de las mismas es directa, las sentencias permitir (*permit*) dejarán pasar el tráfico mientras que las sentencias denegar (*deny*) lo descartarán.

Los comandos para aplicar dichas listas dependen del tipo de interfaz en donde se pretenda configurar, debiendo indicarse además, si la lista será evaluada cuando los paquetes entran o salen de la misma.

Para aplicar una lista sobre una interfaz física se utiliza el comando *ip access-group* como se muestra a continuación.

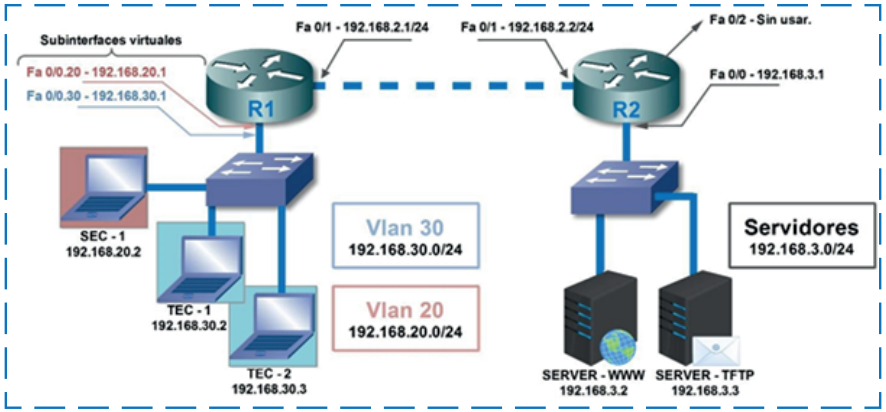
```
Router(config-if)# interface fastethernet 0/0
Router(config-if)# ip access-group ?
  <1-199>
IP access list (standard or extended)
  <1300-2699>   IP expanded access list (standard or extended)
  WORD
Access-list name

Router(config-if)# ip access-group NOMBRE_LISTA ?
  in    inbound packets
  out  outbound packets

Router(config-if)# ip access-group NOMBRE_LISTA in
```

Mientras que para aplicar una lista sobre una línea VTY se utiliza el comando *access-class*, siendo recomendado que está siempre se evalué cuando los paquetes están entrando (*in*) en orden de evitar comportamiento errático. Las listas aplicadas sobre las líneas mencionadas son muy útiles para limitar el acceso remoto a los dispositivos ya sea a través de *telnet* o SSH.

```
Router(config-if)# line vty 0 4
Router(config-line)# access-class ?
  <1-199>
IP access list
  <1300-2699>   IP expanded access list
  WORD
Access-list name

Router(config-line)# access-class NOMBRE_LISTA ?
  in   Filter incoming connections
  out  Filter outgoing connections

Router(config-line)# access-class NOMBRE_LISTA in
```

Para presentar un ejemplo completo de la configuración se presenta la siguiente topología, donde la configuración necesaria para establecer conexión de extremo a extremo ha sido ingresada previamente.

Access Control Lists



Topología para mostrar la configuración y aplicación de ACLs.

Este ejercicio presenta tres redes: Técnicos, secretarias y servidores. Y tiene los objetivos que se enumeran a continuación, nótese que los mismos han sido elegidos en base a su valor pedagógico y no reflejan necesariamente escenarios reales:

1. Limitar el acceso a los routers (Telnet o SSH) exclusivamente a la red de técnicos.

2. Impedir que la red perteneciente a las secretarias pueda comunicarse con la red de servidores.

3. La computadora del técnico 1 (TEC - 1) no podrá ingresar en el servidor web mientras que la computadora del técnico 2 (TEC - 2) no podrá ingresar al servidor TFTP. Todos los demás servicios serán permitidos.

Para cumplir con el primer objetivo se pueden seguir dos aproximaciones diferentes. Es posible denegar específicamente a la red de las "Secretarías", sin embargo, es factible que en un futuro aparezca otra red que tampoco deba tener acceso a la configuración de los dispositivos (Ej.: Ventas), por esta razón, una mejor solución es permitir solamente a la red de "Técnicos" y denegar a todas las demás.

Para este propósito se creará una lista llamada "PermitirTecnicos" y será aplicada en las líneas VTY de ambos *routers.*

```
R1(config)# ip access-list standard PermitirTecnicos
R1(config-std-nacl)# remark [> Permite solo a los tecnicos a traves
de SSH o telnet <]
R1(config-std-nacl)# permit 192.168.30.0 0.0.0.255
R1(config-std-nacl)# deny any
```

```
R1(config)#line vty 0 4
R1(config-line)# access-class PermitirTecnicos in
```

Para ver la composición de las listas creadas así como para verificar su funcionamiento, se puede emplear el comando *show ip access-list*, que se muestra a continuación después de que usuarios pertenecientes a ambas VLANs han tratado de establecer una conexión a través de *telnet*.

Access Control Lists

```
R1# show ip access-lists
Standard IP access list PermitirTecnicos
10 permit 192.168.30.0 0.0.0.255 (2 match(es))
20 deny any (10 match(es))
```

Cada sentencia está numerada para indicar el orden en que estas se ejecutan utilizando un incremento de diez para que nuevas sentencias puedan ser agregadas fácilmente, nótese también que junto a cada una de ellas aparece el número de paquetes que han encontrado en la misma una coincidencia.

Para limitar el acceso en R2, basta con crear la lista nuevamente en este dispositivo y aplicarla a las líneas VTY de la misma manera. Las observaciones hechas a cada una (*Remark*) aparecerán al examinarse la configuración del dispositivo.

En orden de cumplir el segundo objetivo de este ejercicio, puede utilizarse la siguiente lista de control.

```
Router(config)# ip access-list standard DenegarSecretarias
Router(config-std-nacl)# remark [> Deniega a las secretarias <]
Router(config-std-nacl)# deny 192.168.20.0 0.0.0.255
Router(config-std-nacl)# permit any
```

Es necesario advertir la presencia de la sentencia "*permit any*" al final de la lista, ya que de otro modo no solo las secretarias sino que todo el tráfico sería denegado en la interfaz donde llegara a aplicarse debido a las razones explicadas anteriormente.

Una vez creada la lista, al menos de manera conceptual, es necesario decidir en qué router, en que interface y en qué dirección es que esta va a aplicarse.

Una posibilidad sería aplicar la lista en R2, en la interfaz *FastEthernet* 0/1 (Fa 0/1), cuando los paquetes estén entrando (*in*), lo que ciertamente cumpliría el propósito original. No obstante, si se llegara a implementar una nueva red en la interfaz *FastEthernet* 0/2, ahora sin utilizarse, esta también estaría negada a las secretarias a pesar de no estar incluida en el alcance original.

Lo explicado anteriormente constituye la principal desventaja de las listas estándares, ya que al utilizar solamente la dirección de origen del tráfico, se corre el riesgo de impedir el acceso a partes de la red que no debían de ser restringidas si se aplican en la interfaz incorrecta. Por dicho motivo es una buena práctica configurar las listas de este tipo lo más cerca posible a su destino (para no restringir de más).

Access Control Lists

De manera que en este ejemplo se aplicara la lista recién creada en R2 en la interfaz *FastEthernet* 0/0 cuando el tráfico está saliendo de dicha interface.

```
R2(config)# ip access-list standard DenegarSecretarias
R2(config-std-nacl)# remark [> Deniega a las secretarias <]
R2(config-std-nacl)# deny 192.168.20.0 0.0.0.255
R2(config-std-nacl)# permit any
```

```
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip access-group DenegarSecretarias out
```

Para concluir este ejercicio hay que impedir que la computadora del técnico 1 (192.168.30.2/24) tenga acceso al servidor web (192.168.3.2:80 (TCP)) y que la computadora del técnico 2 (192.168.30.3/24) tenga acceso al servidor TFTP (192.168.3.3:69 (UDP)). Todos los otros servicios deben de ser permitidos.

Para cumplir dichos requerimientos puede elaborarse una lista de control de acceso extendida, como se muestra a continuación.

```
Router(config)# ip access-list extended servicios
Router(config-ext-nacl)# deny tcp host 192.168.30.2 host
192.168.3.2 eq 80
Router(config-ext-nacl)# deny udp host 192.168.30.3 host
192.168.3.3 eq 69
Router(config-ext-nacl)# permit ip any any
```

Las listas extendidas tienen la ventaja que al ser más granulares pueden aplicarse en muchos puntos de la topología y aun así cumplir su propósito. Sin embargo, para evitar tráfico y procesamiento innecesario es una buena práctica colocarlas lo más cerca posible al origen de la transmisión.

La última consideración antes de aplicar la lista extendida involucra nuevamente la diferencia existente entre la topología física y la lógica. La red asignada a los técnicos utiliza la VLAN 20 y como puerta de enlace predeterminada la subinterfaz *virtual FastEthernet* 0/0.20, de aplicarse la lista en la interfaz *FastEthernet* 0/0 esta no tendrá ningún efecto, ya que a nivel lógico, esta interfaz no recibe tráfico.

```
R1(config)# ip access-list extended servicios
R1(config-std-nacl)# remark [> Deniega http y ftp a ciertos hosts <]
R1(config-ext-nacl)# deny tcp host 192.168.30.2 host 192.168.3.2 eq 80
R1(config-ext-nacl)# deny udp host 192.168.30.3 host 192.168.3.3 eq 69
R1(config-ext-nacl)# permit ip any any
```

```
R1(config)#inter fastEthernet 0/0.20
R1(config-subif)#ip access-group servicios in
```

Access Control Lists

Otras herramientas

Uno de los riesgos de trabajar con listas de control de acceso consiste en que un mal diseño o aplicación de las mismas puede cortar la comunicación en una red o terminar una sesión remota de manera inesperada.

Así mismo, es una tarea común modificar las mismas ya sea para agregar o quitar sentencias u optimizarlas en algún sentido, por lo que a continuación se presentan algunas herramientas para minimizar el riesgo y ayudar al mantenimiento de las ACLs.

Números de secuencia

Como ya se había mencionado las sentencias de una lista de control poseen un número de secuencia que indica el orden en que estas serán evaluadas y que facilitan la introducción y remoción de las mismas.

Tomando la lista extendida del último ejemplo, tenemos:

```
R1# show ip access-lists

Extended IP access list servicios
    10 deny tcp host 192.168.30.2 host 192.168.3.2 eq www
    20 deny udp host 192.168.30.3 host 192.168.3.3 eq tftp
    30 permit ip any any
```

Si se pretende modificar esta lista, para permitir TFTP y denegar al *host* 192.168.30.2 acceso al servidor web, pueden utilizarse los números de secuencia de estas para remover e incluir las sentencias necesarias.

```
R1(config)# ip access-list extended servicios
R1(config-ext-nacl)# no 20
R1(config-ext-nacl)# 15 deny tcp host 192.168.30.3 host
192.168.3.2 eq 80
R1(config-ext-nacl)# do show ip access-lists

Extended IP access list servicios
  10 deny tcp host 192.168.30.2 host 192.168.3.2 eq www
  15 deny tcp host 192.168.30.3 host 192.168.3.2 eq www
  30 permit ip any any
```

Y en dado caso el incremento entre las sentencias no sea suficiente para incluir nuevas de ellas, puede utilizarse el comando resequence, especificando el número de secuencia inicial y el incremento a utilizarse.

```
R1(config)#ip access-list resequence servicios 10 10

R1#show ip access-list
Extended IP access list servicios
  10 deny tcp host 192.168.30.2 host 192.168.3.2 eq www
  20 deny tcp host 192.168.30.3 host 192.168.3.2 eq www
  30 permit ip any any
```

Access Control Lists

Reinicio programado

Una manera burda de prevenir los problemas ocasionados por una lista de control mal aplicada es el reinicio programado, existiendo dos maneras de programar el mismo utilizando las siguientes palabras clave:

- At: Reinicia el dispositivo en una fecha específica.
- In: Reinicia el dispositivo en una cantidad determinada de minutos.

De esta manera, puede programarse el reinicio de un dispositivo para que dado el caso de una mala aplicación de una ACL este pueda arrancar de nuevo utilizando la última configuración guardada.

```
Router# reload in 5
Reload scheduled in 5 minutes by console
Reload reason: Reload Command
Proceed with reload? [confirm]
Router#

***
*** --- SHUTDOWN in 0:05:00 ---
***

Router#
*Mar  1 00:01:02.571: %SYS-5-SCHEDULED_RELOAD: Reload
requested for 00:06:00 UTC Fri Mar 1 2002 at 00:01:00 UTC Fri
Mar 1 2002 by console. Reload Reason: Reload Command.

Router# show reload
Reload scheduled in 4 minutes and 52 seconds by console
Reload reason: Reload Command
```

Adviértase el uso del comando *show reload*, para visualizar cuándo será el siguiente reinicio programado.
Para cancelar el reinicio del dispositivo puede utilizarse la instrucción reload cancel, como se muestra a continuación.

```
Router# reload cancel
Router#

 ***
*** --- SHUTDOWN ABORTED ---
***

Router#
*Mar  1 00:03:38.599: %SYS-5-SCHEDULED_RELOAD_CAN-
CELLED:  Scheduled reload cancelled at 00:03:38 UTC Fri Mar 1
2002
```

Access Control Lists

Configuration rollback

Una manera más moderna de retornar a una configuración funcional después de haber cometido un error es realizar un *configuration rollback*, donde la palabra inglesa *rollback* hace referencia a desplegar o traer algo de regreso, en este caso una configuración anterior.

Esta instrucción en particular tiene ciertos requerimientos, entre ellos que la memoria disponible del dispositivo sea más grande que el tamaño de los dos archivos de configuración (actual/anterior) combinados y que la capacidad de archivar (*archive*) configuraciones se encuentre activa.

Utilizando este comando es posible revertir la configuración automáticamente a un estado anterior si las instrucciones ingresadas no son confirmadas en cierto límite de tiempo.

Para activar la capacidad de archivar configuraciones, se utilizará la siguiente secuencia de comandos, donde se indica la ruta donde será almacenada la copia de seguridad y la acción que desencadenará la creación del mismo. En este caso se creará un respaldo cada vez que se guarde una nueva configuración.

```
Router(config)# archive
Router(config-archive)# path flash:/backup/backup.cfg
Router(config-archive)#write-memory

Router#dir flash:backup/
Directory of flash:/backup/


    5  -rw-    1056   Mar 1 2002 00:38:06 +00:00  backup.cfg-1

876544 bytes total (851968 bytes free)
```

Para retornar a una configuración anterior después de 10 minutos se puede ejecutar la siguiente instrucción.

```
Router# configure replace flash:/backup/backup.cfg-1 time 10
Timed Rollback: Backing up to flash:/backup/backup.cfg-2

This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
Total number of passes: 0
Rollback Done
```

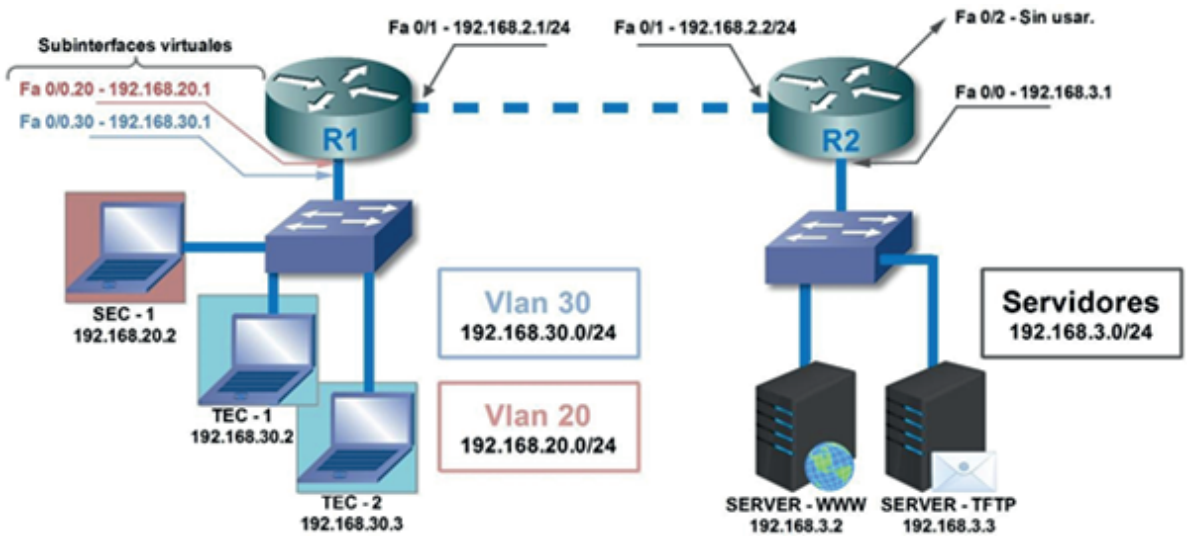
Para guardar los nuevos cambios puede utilizarse el comando *configure confirm* antes de que se cumpla el tiempo asignado para ejecutar el *rollback*.

```
Router# configure confirm
```

Access Control Lists

Resumen de la configuración

Acces Control Lists

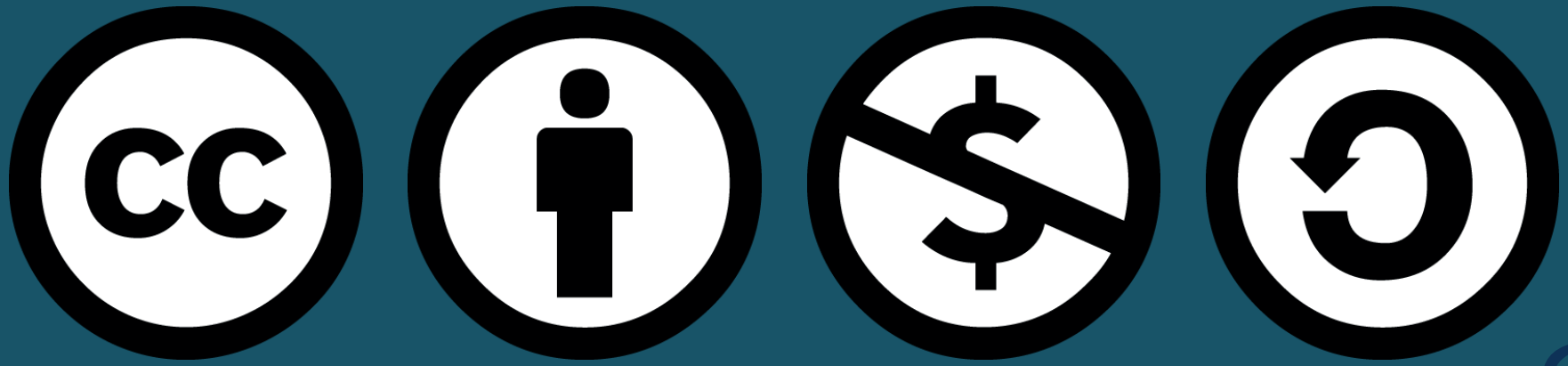


Access Control Lists

```
Switch(config)# vlan 20
Switch(config-vlan)# name Secretarias
Switch(config-vlan)# exit
Switch(config)# vlan 30
Switch(config-vlan)# name Tecnicos

Switch(config)# interface fastethernet 0/1
Switch(config-if)# description SEC-1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 20

Switch(config)# interface fastEthernet 0/2
Switch(config-if)# description TEC-1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 30

Switch(config)# interface fastEthernet 0/3
Switch(config-if)# description TEC-2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 30

Switch# show vlan brief

VLAN Name       Status  Ports
---- ---------- ------- ----------------
20   Secretarias active Fa0/1
30   Tecnicos    active Fa0/2, Fa0/3


Switch(config)# interface fastEthernet 0/10
Switch(config-if)# description trunk hacia R1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport nonegotiate
```

```
R1(config)# interface fastethernet 0/0
R1(config-if)# no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed
state to up

R1(config)# interface fastethernet 0/0.20
%LINK-5-CHANGED: Interface FastEthernet0/0.20, changed
state to up
R1(config-subif)# encapsulation dot1q 20
R1(config-subif)# description gateway de la Vlan 20
R1(config-subif)# ip address 192.168.20.1 255.255.255.0
R1(config-subif)# exit

R1(config)# interface fastethernet 0/0.30
R1(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.30, changed
state to up
R1(config-subif)# encapsulation dot1q 30
R1(config-subif)# description gateway de la vlan 30
R1(config-subif)# ip address 192.168.30.1 255.255.255.0
R1(config-subif)# exit

R1(config)# interface fastEthernet 0/1
R1(config-if)# ip address 192.168.2.1 255.255.255.0
R1(config-if)# no shutdown

R1(config)#ip route 0.0.0.0 0.0.0.0 fastEthernet 0/1
R1(config)#

R1(config)# ip access-list standard PermitirTecnicos
R1(config-std-nacl)# remark [> Permite solo a los tecnicos a
traves de SSH o telnet <]
R1(config-std-nacl)# permit 192.168.30.0 0.0.0.255
R1(config-std-nacl)# deny any

R1(config)#line vty 0 4
R1(config-line)# access-class PermitirTecnicos in
R1(config)# ip access-list extended servicios
R1(config-std-nacl)# remark [> Deniega http y ftp a ciertos
hosts <]
R1(config-ext-nacl)# deny tcp host 192.168.30.2 host
192.168.3.2 eq 80
R1(config-ext-nacl)# deny udp host 192.168.30.3 host
192.168.3.3 eq 69
R1(config-ext-nacl)# permit ip any any

R1(config)#inter fastethernet 0/0.20
R1(config-subif)#ip access-group servicios in
```

```
R2(config)# interface fastethernet 0/0
R2(config-if)# ip address 192.168.3.1 255.255.255.0
R2(config-if)# no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed
state to up


R2(config)# interface fastethernet 0/1
R2(config-if)# ip address 192.168.2.2 255.255.255.0
R2(config-if)# no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/1, changed
state to up

R2(config)# ip route 0.0.0.0 0.0.0.0 fastEthernet 0/1

R2(config)# ip access-list standard PermitirTecnicos
R2(config-std-nacl)# remark [> Permite solo a los tecnicos a
traves de SSH o telnet <]
R2(config-std-nacl)# permit 192.168.30.0 0.0.0.255
R2(config-std-nacl)# deny any

R2(config)#line vty 0 4
R2(config-line)# access-class PermitirTecnicos in

R2(config)# ip access-list standard DenegarSecretarias
R2(config-std-nacl)# remark [> Deniega a las secretarias <]
R2(config-std-nacl)# deny 192.168.20.0 0.0.0.255
R2(config-std-nacl)# permit any

R2(config)#interface fastEthernet 0/0
R2(config-if)#ip access-group DenegarSecretarias out
```



Diseño y edición:

María Esther Pineda
Carolina Villatoro