CURSO A DISTANCIA CCNA: **Técnico experto en redes e Internet.**

MATERIAL DIDÁCTICO COMPLEMENTARIO:

Máscaras Wildcard y ACL. CCNA 2: Módulo 11.

CNICE RUBÉN MUÑOZ HERNÁNDEZ.

1.- INTRODUCCIÓN.

Aunque los materiales didácticos del CCNA son de una excelente calidad, en nuestra opinión, hemos creído conveniente aportar ciertos contenidos complementarios dadas las características específicas de un curso a distancia. En este documento encontrará información detallada sobre cómo calcular máscaras **wildcard** y sobre cómo actúan.

Este anexo está pensado para ser estudiado después de que usted haya preparado el módulo 11 del CCNA 2, y más concretamente la sección 11.1.4. En el momento de redactar este material, la figura 3 de dicha sección contenía errores que podrían oscurecer el modus operandi de estas máscaras.

2.- DEFINICIÓN Y OBJETIVO DE LAS MÁSCARAS WILDCARD.

Una máscara wildcard es sencillamente una agrupación de 32 bits dividida en cuatro bloques de ocho bits cada uno (octetos). La apariencia de una máscara wildcard le recordará problablemente a una máscara de subred. Salvo esa apariencia, no existe otra relación entre ambas.

Por ejemplo, una máscara wildcard puede tener este aspecto: 0.0.0.255, o escrito en binario:

Máscara Wildcard en binario.			
00000000	00000000	00000000	11111111

Las máscaras Wildcard se emplean junto con un valor IP para selecionar direcciones IP, esto es así gracias a que la máscara wildcard indica con sus ceros y sus unos qué bits han de compararse o no. **Un cero indica que el bit ha de compararse y un uno indica que se ignore.**

Por ejemplo, suponga que empleamos la IP 192.168.1.0 junto con la máscara wildcard 0.0.0.255 para seleccionar direcciones IP. Los ceros nos están diciendo que debemos comparar los valores de los tres primeros octetos y los unos del cuarto octeto nos dicen que da igual que valor tenga dicho octeto.

Por tanto, quedarán selecionados valores como:

192.168.1.1,
192.168.1.23,
192.168.1.145,

y se quedarán descartadas IP como:

192.168.2.145,
100.168.1.0,
192.167.1.76.

Es decir, con la máscara wildcard 0.0.0.255 todas las IPs que se seleccionen deberán tener los tres primeros octetos de la forma 192.168.1, en tanto que el cuarto octeto, como queda oculto por los unos de la máscara, puede tomar cualquier valor.

Suponga que tenemos la máscara wildcard 0.0.255.255 y la IP 10.1.0.0, ¿qué valores de la siguienta lista se seleccionarán? Piénselo un poco antes de ver la respuesta.

a) 10.1.1.1 b) 10.2.1.1 c) 20.1.0.1 d)10.1.255.255 e)10.10.0.1

f) 10.1.20.2 g) 11.1.3.2 h) 100.1.0.0 i) 10.10.1.1 j) 10.3.0.1

En efecto, las IPs seleccionadas son a, d y f.

Las IP seleccionadas deberán tener los dos primeros octetos de la forma 10.1, mientras que no importa qué valores ocupen las posiciones de los octetos tercero y cuarto.

Sin embargo, no todas las máscaras wildcard se interpretan tan rápidamente. Hasta ahora hemos visto ejemplos en los que los octetos o bien tenían todos sus bits a cero o bien a uno.

Supongamos ahora una máscara wildcard de la forma 0.0.0.**15** asociada con la IP 192.168.1.**48**, ¿qué IPs quedarán seleccionadas?

Es fácil ver que los tres primeros octetos habrán de ser de la forma 192.168.1.

Respecto al cuarto octeto de la máscara wildcard observe que su expresión binaria es:

00001111

Es decir, no debemos tener en cuenta el valor de los últimos cuatro bits a la hora de compararlo con el número **48**, que en binario se escribe como:

00110000

Analicemos, por ejemplo, si la IP 192.168.1.51 es selecionada. Observe que 51 se escribe como:

00110011

Al aplicar la máscara wildcard, ignoramos el valor de los últimos cuatro bits:

0011~~0011~~

Fíjese que al ignorar el valor de estos bits el número resultante es 48 y por tanto la IP sí es selecionada.

Es fácil ver la lista de valores que serán seleccionados, basta con calcular el valor de las combinaciones :

de la

0011 0000

hasta la

0011 1111

Es decir, desde la IP 192.168.1.48 hasta la 192.168.1.63. En resumen, la IP 192.168.1.48 con máscara wildcard 0.0.0.15 selecciona un rango de direcciones IP.

Como usted ya sabe por la teoría del módulo 11 del CCNA 2, una aplicación fundamental de las máscaras wildcard es su uso al redactar ACL. Las ACL son filtros de paquetes IP y permiten aceptar o descartar paquetes en función de la IP de origen (ACL estándar) o bien en función de la IP de origen, IP de destino y protocolo(tcp, udp, icmp...) (ACL extendidas). Como ha visto por los ejemplos anteriores, las máscaras wildcard nos van a permitir seleccionar rangos de IPs.

A continuación veremos algunos ejemplos más de máscaras wildcard y posteriormente los combinaremos con sentencias ACL. Los ejemplos que mostramos se han presentado en orden graduado de complejidad. Como verá, una vez que se ha entendido bien como actúan las mascaras wildcard, no suponen ninguna dificultad.

3.- APLICANDO MÁSCARAS WILDCARD.

Una vez visto el método de trabajo de las máscaras Wildcard, conviene mostrar diversos ejemplos de aplicación.

3.1.- Ejemplo 1. Con este ejercicio se pretende consolidar el método de trabajo con máscaras Wildcard. Dada la siguiente IP y máscara wildcard:

IP	202	20	0	0
Máscara Wildcard	0	0	255	255

Decir qué IPs quedarán selecionadas de entre las siguientes: 202.20.2.1, 102.20.0.0 y 202.20.22.0

Solución:

En este caso, la máscara wildcard nos obliga a comparar los dos primeros octetos, que habrán de ser por tanto de la forma 202.20. Los dos últimos octetos son ignorados de modo que pueden tomar valores cualesquiera. Las IPs seleccionadas serán:

202.20.2.1 y 202.20.22.0

3.2.- Ejemplo 2. Aquí tomaremos la IP 172.16.0.0 con máscara Wildcard 0.0.**127**.255 y la aplicaremos sobre las siguientes IPs: 172.16.130.1, 121.0.127.2 y 172.16.15.253. Tras realizar el ejercicio, interpretaremos el significado de esta máscara.

La información más directa que podemos sacar de la IP 172.16.0.0 con máscara wildcard 0.0.127.255, es que los dos primeros octetos habrán de ser de la forma 172.16 y el último podrá tomar cualquier valor.

Es claro, por consiguiente, que la IP 121.0.127.2 queda descartada.

Estudiemos ahora qué ocurre en el tercer octeto. Para ello es útil escribirlo en forma binaria:

IP	172	16	0	0
Máscara Wildcard	0	0	**01111111**	255

El significado de esta máscara es que sólo se comparará el primer bit del tercer octeto, los demás serán ignorados. Al aplicar esta máscara sobre las dos IPs que nos quedan obtenemos:

IPs				
172.16.130.1	172	16	1~~0000010~~	~~1~~
172.16.15.253	172	16	0~~0001111~~	~~253~~

Tenga en cuenta que el bit no tachado es el primero y que un 1 en esa posición corresponde con el valor decimal 128 . Así pues, al ignorar los bits que mostramos tachados las IPs resultantes son:

IPs	***IPs + Máscara Wildcard***			
172.16.130.1	172	16	128	0
172.16.15.253	172	16	0	0

Observe que sólo la última IP será seleccionada.

Fíjese también que los dígitos binarios de un octeto que comienzan por 0 representan la mitad inferior de los valores de ese octeto: del 0 al 127. Los que comienzan por uno suponen la mitad superior del octeto: del 128 al 255.

Por tanto, la máscara Wildcard del ejemplo anterior (0.0.127.255) estaría seleccionando la mitad inferior de las direcciones IP de la red 172.16.0.0/16.

La posibilidad de seleccionar partes de una red es muy interesante a la hora de establecer criterios de seguridad en una red. Por ejemplo, podríamos estar interesados en dar permiso sólo a una parte de los hosts de una LAN para acceder a un determinado servidor.

3.3.- Ejemplo 3. En este ejercicio profundizaremos en la idea apuntada en el caso anterior. Le aconsejamos que intente resolverlo antes de leer la solución.

Se pide dar una IP junto con una máscara Wildcard que permita seleccionar la segunda mitad de las IPs correspondientes a la red 180.42.111.0/24.

Para comenzar listemos las IPs que pertenecen a dicha red:

IPs decimal	***IPs con el cuarto octeto en binario***			
180.42.111.0	180	42	111	00000000
180.42.111.1	180	42	111	00000001
...	...	...	...	...
180.42.111.127	180	42	111	01111111
180.42.111.128	180	42	111	10000000
...	...	...	...	...
180.42.111.255	180	42	111	11111111

Como todas las IPs que queremos seleccionar pertenecen a la red 180.42.111.0/24, la parte fácil del problema es resolver los valores de los tres primeros octetos:

180.42.111.---

con máscara wildcard (los tres octetos primeros han de ser comparados):

0.0.0.---

Veamos qué ocurre con el cuarto octeto. Queremos tomar las IPs con valores que van del 128 al 255. El patrón común de todos estos valores es que el primer bit es siempre un 1, y el resto de bits puede tomar cualquier valor. Los bits que pueden tomar cualquier valor debemos tacharlos con la máscara wildcard, de modo que ésta quedará de la forma:

Máscara Wildcard			
0	0	0	01111111
0	0	0	127

Una vez que hemos tachado esos siete bits en la máscara wildcard, nos interesa que el bit a comparar (el primero) tome el valor 1. Ese bit a 1, por su posición relativa, vale 128, por tanto la IP que debemos usar es la 180.42.111.128.

Verifique que efectivamente la IP 180.42.111.128 con máscara wildcard 0.0.0.127 selecciona el rango de IPs comprendido entre la 180.42.111.128 y la 180.42.111.255.

3.4.- Ejemplo 4. Veamos otro ejemplo curioso con la misma red. Cree una máscara Wildcard para seleccionar los valores **pares** (par en el cuarto octeto) de las IPs correspondientes a la red 180.42.111.0/24.

Lo que ya sabemos es la estructura de los tres primeros octetos es:

180.42.111.---

con máscara wildcard:

0.0.0.---

Respecto al cuarto octeto, observe en la tabla que las IPs expresadas en binario terminan siempre en cero cuando son valores pares. Una vez que nos percatamos de cuál es el patrón común el ejercicio está prácticamente resuelto.

IPs decimal	*IPs con cuarto octeto en binario*	
180.42.111.0	**180.42.111**	**00000000**
180.42.111.1	180.42.111	00000001
...	...	...
180.42.111.126	**180.42.111**	**01111110**
180.42.111.127	180.42.111	01111111
180.42.111.128	**180.42.111**	**10000000**
...	...	...
180.42.111.254	**180.42.111**	**11111110**
180.42.111.255	180.42.111	11111111

Puesto que en este caso el valor que tomen los siete primeros bits del cuarto octeto han de ser ignorados, la máscara Wildcard los tachará.

Máscara Wildcard			
0	0	0	11111110
0	0	0	254

Al ignorar los siete primeros bits del octeto, las IPs de la red tomarán el valor 0 si son pares y el valor 1 si son impares (vea la tabla de más abajo). Por tanto debemos escoger la IP 180.42.111.0 para realizar nuestra selección.

IPs decimal	*IPs + Máscara Wildcard*			
180.42.111.0	180	42	111	~~0000000~~0
180.42.111.1	180	42	111	~~0000000~~1
...	...	...	...	...
180.42.111.126	180	42	111	~~0111111~~0
180.42.111.127	180	42	111	~~0111111~~1
180.42.111.128	180	42	111	~~1000000~~0
...	...	...	...	...
180.42.111.254	180	42	111	~~1111111~~0
180.42.111.255	180	42	111	~~1111111~~1

Resumiendo la IP 180.42.111.0 con máscara 0.0.0.254 seleciona las IPs con valores pares de la red 180.42.111.0/24.

3.5.- Ejemplo 5. Este es un ejercicio un poco más complejo. De hecho es improbable que encuentre ejercicios más complicados en el CCNA.

Segmente la dirección de red 193.150.27.0/24 en seis subredes y seleccione mediante una máscara Wilcard el tercer cuarto de la quinta subred.

Para la segmentación de la dirección de red dada, hemos de usar los tres primeros bits del cuarto octeto. El listado se las subredes queda:

Subredes en decimal	*Subredes en binario*			
193.150.27.32	11000001	10010110	00011011	001 00000
193.150.27.64	11000001	10010110	00011011	010 00000
193.150.27.96	11000001	10010110	00011011	011 00000
193.150.27.128	11000001	10010110	00011011	100 00000
193.150.27.160	11000001	10010110	00011011	101 00000
193.150.27.192	11000001	10010110	00011011	110 00000

Con máscara de subred: 255.255.255.224

La quinta subred es:

Subred en decimal	***Subred en binario***			
193.150.27.160	11000001	10010110	00011011	101 00000

Observe la tabla de direcciones IP asociada a dicha subred:

Direcciones IP en decimal	***Cuarto***	***Direcciones IP en binario***			
193.150.27.160	1º	11000001	10010110	00011011	101 **00** 000
193.150.27.161	1º	11000001	10010110	00011011	101 **00** 001
193.150.27.162	1º	11000001	10010110	00011011	101 **00** 010
193.150.27.163	1º	11000001	10010110	00011011	101 **00** 011
193.150.27.164	1º	11000001	10010110	00011011	101 **00** 100
193.150.27.165	1º	11000001	10010110	00011011	101 **00** 101
193.150.27.166	1º	11000001	10010110	00011011	101 **00** 110
193.150.27.167	1º	11000001	10010110	00011011	101 **00** 111
193.150.27.168	2º	11000001	10010110	00011011	101 **01** 000
...	2º	...	...	...	...
193.150.27.175	2º	11000001	10010110	00011011	101 **01** 111
193.150.27.176	**3º**	11000001	10010110	00011011	101 **10** 000
...	**3º**	...	...	...	...
193.150.27.183	**3º**	11000001	10010110	00011011	101 **10** 111
193.150.27.184	4º	11000001	10010110	00011011	101 **11** 000
...	4º	...	...	...	...
193.150.27.191	4º	11000001	10010110	00011011	101 **11** 111

Fíjese que en el último octeto los tres primeros bits están dedicados a la dirección de subred. De los cinco bits restantes, los dos primeros nos sirven para discernir en qué cuarto estamos. Los hemos resaltado en negrita.

La combinación 00 aparece en todas la direcciones que pertenecen al primer cuarto de la subred, la combinación 01 al segundo cuarto, la 10 al tercero, que es la que buscamos, y la 11 al último cuarto.

Con esta información ya podemos dar la IP que necesitamos y la máscara wildcard asociada:

IP	193	150	27	10110000
Máscara Wildcard	0	0	0	00000111

Observe que la clave para resolver este ejercicio es ver que tanto los tres primeros bits del último octeto (los bits de subred), como los dos siguientes (los bits que definen cada cuarto) conforman el patrón común de todas las IPs que queremos seleccionar.

Este conjunto de IPs puede tomar para los tres últimos bits del cuarto octeto cualquiera de las combinaciones posibles. Por tanto, la máscara Wildcard deberá ocultar esos tres últimos bits.

La expresión en valores decimales de la máscara Wildcard y de la IP será:

IP	192	150	27	176
Máscara Wildcard	0	0	0	7

Verifique que con la combinación anterior seleccionamos las IPs del rango:

192.150.27.176 - 192.150.27.183

3.6.- Ejemplo 6. El caso anterior muestra cómo discriminar direcciones IP de host dentro de una subred con una máscara Wildcard. Ahora veremos cómo seleccionar varias subredes completas. La metodología de trabajo es exactamente la misma, por eso le animamos a que antes de leer la solución intente resolverlo.

Se ha segmentado la dirección de red 20.0.0.0/8 empleando todos los bits del segundo octeto. Se quiere establecer una selección de direcciones IP que tenga en cuenta todas las direcciones pertenecientes a las subredes que van de la 20.128.0.0/16 a la 20.143.0.0/16

Fíjese que aquí la clave para resolver el problema es extraer el patrón común del segundo octeto. El primer octeto ha de tomar siempre el valor 20 y los octetos tercero y cuarto pueden tomar valores cualesquiera, de modo que la IP que buscamos y su máscara wildcard serán de la forma:

IP: 20.---.0.0

y máscara wildcard: 0.---. 255.255

La tabla de más abajo muestra el patrón que necesitamos:

IPs	***IPs con el segundo octeto en binario***			
20.0.0.0	20	00000000	0	0
20.1.0.0	20	00000001	0	0
20.2.0.0	20	00000010	0	0
...	...	...	...	...
20.127.0.0	20	01111111	0	0
20.**128**.0.0	20	**1000**0000	0	0
20.**129**.0.0	20	**1000**0001	0	0
...	...	...	...	...
20.**142**.0.0	20	**1000**1110	0	0
20.**143**.0.0	20	**1000**1111	0	0
20.144.0.0	20	10010000	0	0
20.145.0.0	20	10010001	0	0
...	...	...	...	...
20.254.0.0	20	11111110	0	0

Observe que ese patrón común de las subredes 128 a la 143 es tener los cuatro primeros bits del segundo octeto como 1000. Con esta información debería estar claro que para seleccionar las IPs solicitadas tenemos que usar la combinación:

IP	20	10000000	0	0
Máscara Wildcard	0	00001111	255	255

Sus valores decimales son:

IP	20	128	0	0
Máscara Wildcard	0	15	255	255

4.- MÁSCARAS WILDCARD Y ACL.

Como hemos comentado anteriormente una aplicación fundamental de las máscaras wildcard es su uso en las listas de control de acceso o ACL.

Como usted sabe por la teoría del módulo 11 del CCNA 2, las ACL son filtros de paquetes IP. Estas listas de control de acceso toman la decisión de aceptar o descartar los paquetes en función de una serie de parámetros como, por ejemplo, la dirección IP origen en una ACL estándar.

4.1.- ACL estándar.

Veamos a continuación una red donde su administrador tendrá la necesidad de colocar una ACL estándar.

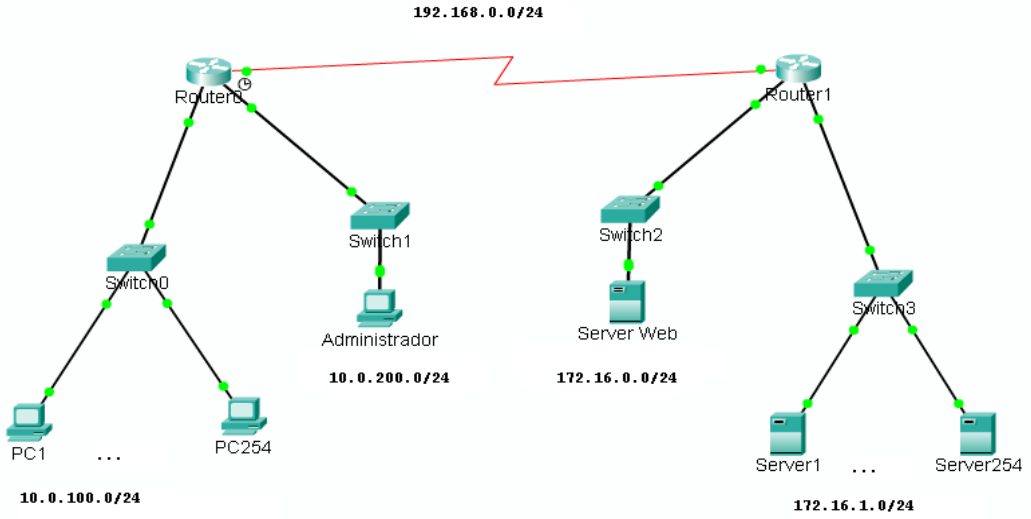


Figura 4.1

En la topología mostrada en la imagen superior se ha dado el direccionamiento de la red. Para las interfaces de los routers se han escogido los valores IPs más bajos de cada red. Por ejemplo para la interfaz ethernet del router 1 conectada a la red 172.16.0.0/24 la IP será 172.16.0.1.

Ejemplo 1. Supongamos que el administrador quiere bloquear el acceso de los equipos de la red 10.0.100.0/24 a la red administración (la 10.0.200.0/24). Una opción es crear una ACL estándar y asignarla a la interfaz ethernet 10.0.200.1. Recuerde que las ACL estándar han de situarse lo más cerca posible del destino. La ACL se redactaría en el modo global de configuración del router 0 del siguiente modo:

router0(config)# access-list 1 deny 10.0.100.0 0.0.0.255

router0(config)# access-list 1 permit any

Observe que la máscara wildcard 0.0.0.255 junto con la IP 10.0.100.0 permite seleccionar el rango que va de la IP 10.0.100.1 a la 10.0.100.255. Por tanto, abarca el

conjunto de IPs que pertenecen a la red cuyo tráfico deseamos restringir. Fíjese que es necesario añadir la línea permit any, pues de otro modo actúa la negación implicita (deny any) que existe en toda lista de acceso.

A la hora de asignar la lista de acceso a una interfaz, escribiremos:

router0(config-if) ip access-group 1 out

Esta asignación ha de hacerse en la interfaz con IP 10.0.200.1, que es la más cercana al destino. Observe que visto desde el interior del router los paquetes IP saldrían hacia la red 10.0.200.0/24, por ese motivo ponemos al final de la sentencia la palabra out.

Ejemplo 2. Hemos visto en el caso anterior cómo negar el acceso a una red completa, pero usted ya sabe seleccionar partes de una red empleando máscaras wildcard.

Siguiendo con la topología anterior, supongamos ahora que queremos negar el acceso al servidor web a los hosts que pretenecen a la mitad superior de la red 10.0.100.0/24 empleando una ACL estándar. (Recuerde el ejemplo de la sección 3.3).

Es claro que deberíamos escribir esa ACL en el router 1 y que deberíamos asignarla a la interfaz ethernet 172.16.0.1. Veamos cómo redactarla.

router1(config)# access-list 1 deny 10.0.100.128 0.0.0127

router1(config)# access-list 1 permit any

Para aplicarla en la interfaz 172.16.0.1 escribiremos:

router1(config-if) ip access-group 1 out

Ejemplo 3. Por último mostraremos cómo emplear ACL estándar para un solo host. En este ejercicio tenemos que permitir el acceso a la red 172.16.1.0/24 sólamente al host 10.0.100.199.

Deberemos aplicar la ACL estándar lo más cerca posible del destino (la red 172.16.1.0/24), lo cual significa que configuraremos nuestra ACL en el router 1 y la aplicaremos en la interfaz 172.16.1.1.

router1(config)# access-list 2 permit 10.0.100.199 0.0.0.0

router1(config)# access-list 2 deny any

Observe que en este caso la máscara wildcard ha de llevar todos sus bits a cero. Hay que comparar todos los bits de las IPs de los paquetes que entren en el router con el valor 10.0.100.199, que es el único que permitiremos. Hemos escrito la última línea por claridad. Como sabe, la sentencia deny any está implícita en toda lista de acceso, pero recomendamos escribirla para evitar posibles equívocos.

La máscara wildcard 0.0.0.0 puede sustituirse por la abreviatura host, y ciertamente es más directa su interpretación. La access-list 2 quedaría del siguiente modo:

router1(config)# access-list 2 permit host 10.0.100.199

router1(config)# access-list 2 deny any

Y podrá comprobar que la misma lista de acceso puede también escribirse como:

router1(config)# access-list 2 permit 10.0.100.199

router1(config)# access-list 2 deny any

Es decir, la ausencia de máscara wildcard implica por defecto el valor 0.0.0.0.

Para asignar la ACL anterior a la interfaz correcta (la 172.16.1.1):

router1(config-if) ip access-group 2 out

Fíjese que hemos escrito otra abreviatura habitual en las ACL: any. Esta abreviatura permite seleccionar cualquier IP. ¿Qué combinación de IP y máscara wildcard usaría usted para este propósito?

Suponga que escribimos una línea como esta:

router1(config)# access-list 2 deny 0.0.0.0 255.255.255.255

Al aplicar la máscara wildcard 255.255.255.255 sobre cualquier IP el resultado es siempre el mismo. Puesto que tachamos todos los bits, el valor de todos los octetos se pone a cero. Al comparar con la IP escrita en la ACL (0.0.0.0) siempre hay coincidencia.

Por consiguiente, any es la abreviatura de 0.0.0.0 255.255.255.255.

4.2.- ACL extendidas.

Como usted ya sabe por la teoría, las ACL extendidas nos permiten filtrar los paquetes IP en función de la dirección de origen, la dirección de destino y además podemos podemos concretar sobre qué puerto TCP o UDP queremos realizar la comprobación.

Vamos a ver varios ejemplos de aplicación tomando como red de trabajo la mostrada en la figura 4.1. En la red no habrá ninguna ACL configurada y cada ejemplo se tratará de modo independiente.

Ejemplo 4. Se desea filtrar el trafico telnet al servidor con IP 172.16.1.10 desde la red 10.0.100.0/24.

Lo primero a tener en cuenta es que la ACL extendida ha de colocarse lo más cerca posible del origen. En nuestro caso, por tanto, habremos de configurar la ACL en el router 0 y aplicarla en la interfaz ethernet 10.0.100.1.

router0(config)# access-list 100 deny tcp 10.0.100.0 0.0.0.255 host 172.16.1.10 eq telnet

router0(config)# access-list 100 permit ip any any

Asignación a la interfaz:

router0(config)# interface fastethernet 0/0

router0(config-if)# ip access-group 100 in

Observe que hemos de comprobar los paquetes que entran en la interfaz, por eso debemos escribir *in* al final de la sentencia de asignación. Hemos supuesto que la interfaz con IP 10.0.100.1 es la fastethernet 0/0.

Ejemplo 5. Se le pide al administrador de la red que sólamente los host con IP impar de la red 10.0.100.0/24 puedan acceder al servidor Web (172.16.0.2) y además sólo al puerto 80. Para construir la máscara wildcard, recuerde el ejemplo de la sección 3.4.

En el momento que le pidan filtrar tráfico en función de un puerto TCP o UDP, usted ya sabe que ha de emplear una lista de acceso extendida.

Una vez decidido que tipo de ACL necesitaremos, hay que buscar su ubicación. En nuestro caso, deberemos configurarla en el router 0 y asignarla a la interfaz 10.0.100.1 para los paquetes entrantes.

router0(config)#access-list 100 permit tcp 10.0.100.1 0.0.0.254 host 172.16.0.2 eq www

router0(config)# access-list 100 deny ip any any

router0(config)#interfaz fastethernet 0/0

router0(config-if)# ip access-group 100 in

Ejemplo 6. Se ha establecido una política de seguridad en la red representada en la figura 4.1 de modo que la red 10.0.100.0/24 quede dividida en dos partes según los valores de sus IPs. Se quieren establecer estas dos condiciones referidas al acceso a los servidores de la red 172.16.1.0/24:

a) La mitad inferior del rango de IPs (de la 10.0.100.1 a la 10.0.100.127) podrá acceder a todos los servidores de la red 172.16.1.0/24.

b) La mitad superior de de la red 10.0.100.0/24 tendrá negado el acceso a los servidores con IPs en el rango que abarca la mitad inferior de los valores IP de la red 172.16.1.0/24

Fíjese que ahora hemos de tener en cuenta tanto direcciones de origen como direcciones de destino, por lo tanto es imperativo hacer uso de listas de acceso extendidas.

El ejemplo que nos ocupa tiene varias soluciones posibles. Una táctica que puede utilizar cualquier administrador, y que puede ser la más cómoda, es permitir tráfico. Por defecto lo que no esté permitido está negado. De hecho ese es el criterio para establecer un deny implícito al final de toda lista de acceso.

Con el criterio descrito en el párrafo anterior, la condición b podemos leerla en positivo. Denegar el acceso a la mitad inferior de los servidores es equivalente a permitirlo a la mitad superior, puesto que cualquier otro tráfico es negado por defecto.

router0(config)#access-list 100 permit ip 10.0.100.0 0.0.0.127 172.16.1.0 0.0.0.255

router0(config)#access-list 100 permit ip 10.0.100.128 0.0.0.127 172.16.1.128 0.0.0.127

router0(config)# access-list 100 deny ip any any

router0(config)#interfaz fastethernet 0/0

router0(config-if)# ip access-group 100 in

Ejemplo 7. Hasta ahora hemos visto cada ejemplo de modo independiente. Vamos a combinar ahora el ejemplo anterior con dos restricciones más.

a) La primera condición es que los host de la red 10.0.100.0/24 sólo puedan acceder al servidor Web a través del puerto 80.

b) La segunda es que no puedan acceder a la red del Administrador (10.0.200.0/24)

La solución es:

router0(config)#access-list 100 permit ip 10.0.100.0 0.0.0.127 172.16.1.0 0.0.0.255

router0(config)#access-list 100 permit ip 10.0.100.128 0.0.0.127 172.16.1.128 0.0.0.127

router0(config)#access-list 100 permit tcp 10.0.100.0 0.0.0.255 host 172.16.0.2 eq 80

router0(config)# access-list 100 deny ip any any

router0(config)#interfaz fastethernet 0/0

router0(config-if)# ip access-group 100 in

Observe que no es necesario denegar el acceso a la red 10.0.200.0/24 pues por defecto no está permitido.

5.- PROPUESTA DE EJERCICIOS.

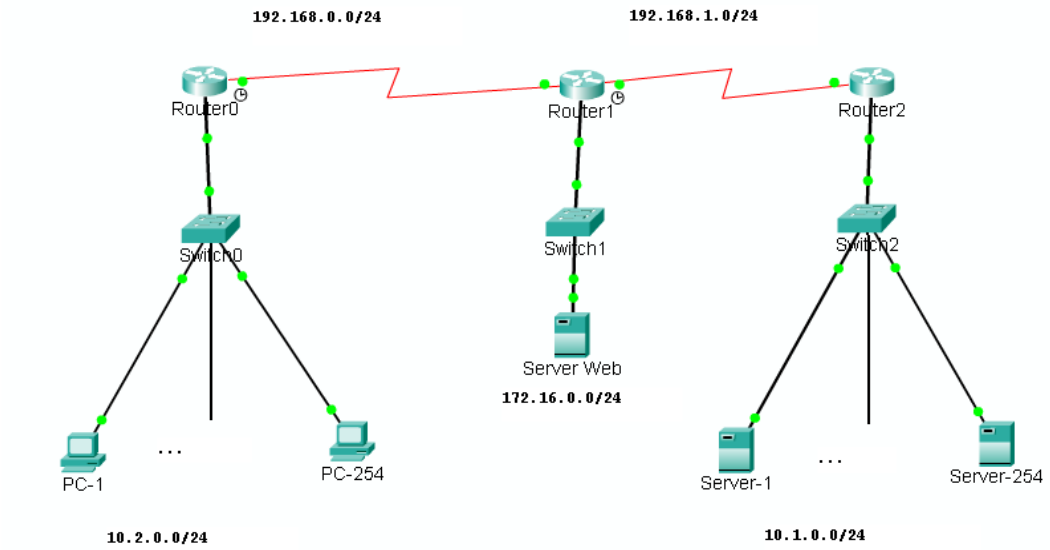
5.1.- Segmente la dirección de red 14.0.0.0/8 en 15 subredes y utilice una máscara Wildcard para seleccionar la primera mitad de las subredes obtenidas.

5.2.- Seleccione las direcciones de host impares de la red 195.170.93.0/24

5.3.- Seleccione con una máscara Wildcard las redes de clase B que se encuentran entre la 150.32.0.0/16 y la 150.63.0.0/16

5.4.- Segmente la dirección de red 172.16.0.0/16 en 5 subredes y seleccione la primera mitad de los host de la tercera subred.

5.5.- Dada la red mostrada en la figura siguiente:



Se pide configurar ACL para esteblecer las siguientes restricciones:

a) Sólo tendrán acceso al servidor Web los equipos de la red 10.2.0.0/24

b) El servidor con IP 10.1.0.10 sólo será accesible desde el host 10.2.0.2

c) El servidor con IP 10.1.0.11 no podrá ser accesible para los hosts con IPs cuyo último octeto tenga un valor par.

6.- PACKET TRACER Y ACL.

Le aconsejamos practicar con el Packet Tracer los ejemplos y ejercicios propuestos en este documento.

Los archivos con las soluciones a los ejemplos dados en esta guía se adjuntan en los archivos:

Archivo	*Ejemplo*
ACL estándar 1 2 3.pkt	Ejemplos 1, 2 y 3 de la sección 4.1
ACL extendida 4.pkt	Ejemplo 4 de la sección 4.2
ACL extendida 5.pkt	Ejemplo 5 de la sección 4.2
ACL extendida 6.pkt	Ejemplo 6 de la sección 4.2
ACL extendida 7.pkt	Ejemplo 7 de la sección 4.2

NOTA: Se ha trabajado con la version 3.2

En estos archivos se ha configurado el protocolo de enrutamiento RIP version 2. El direccionamiento corresponde con el de la figura 4.1.

Además de configurar las ACL de los ejemplos, se han creado paquetes en el modo de simulación para probar que funcionan correctamente:

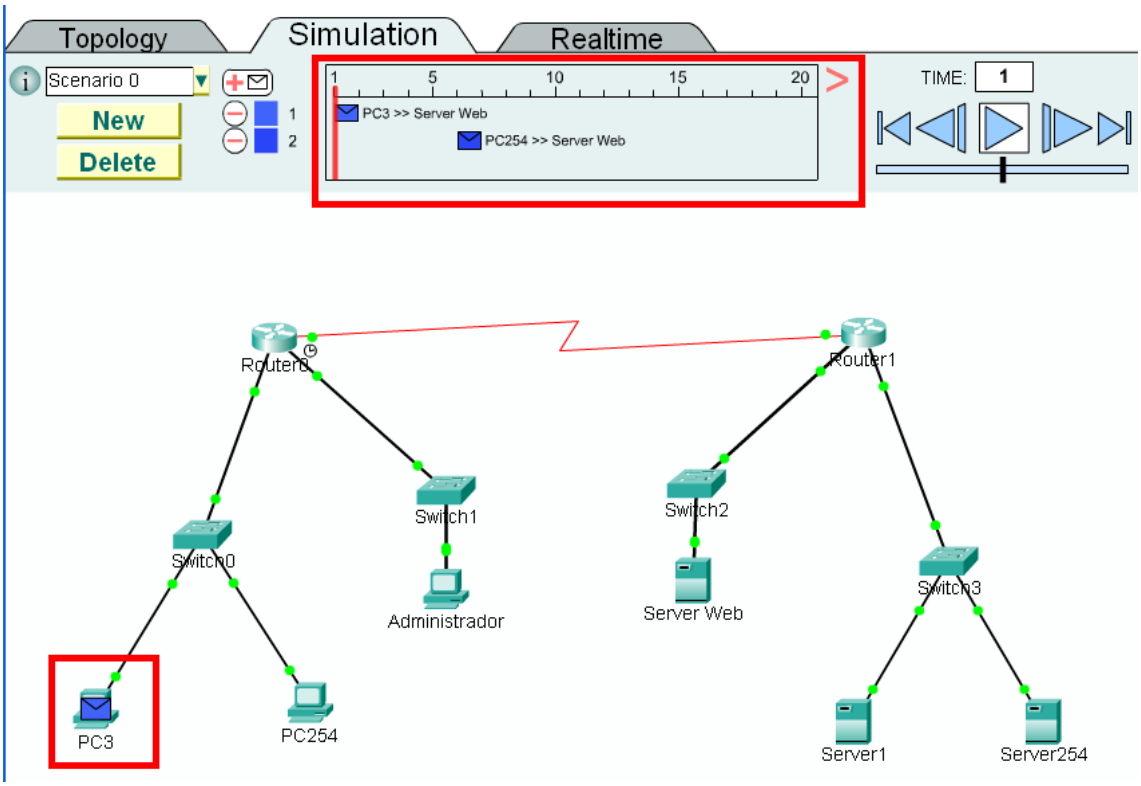


Figura 6.1

En la figura 6.1 vemos el archivo para el ejemplo 5. Hemos recuadrado en rojo en la parte superior el proceso que se simulará. Observe que se han creado dos paquetes para probar la ACL, uno con origen en un host de IP impar y otro de IP par.

Si hace click sobre cualquier paquete, se abrirá una ventana mostrando las características del paquete creado. Por ejemplo, marcando sobre el situado en el PC3:

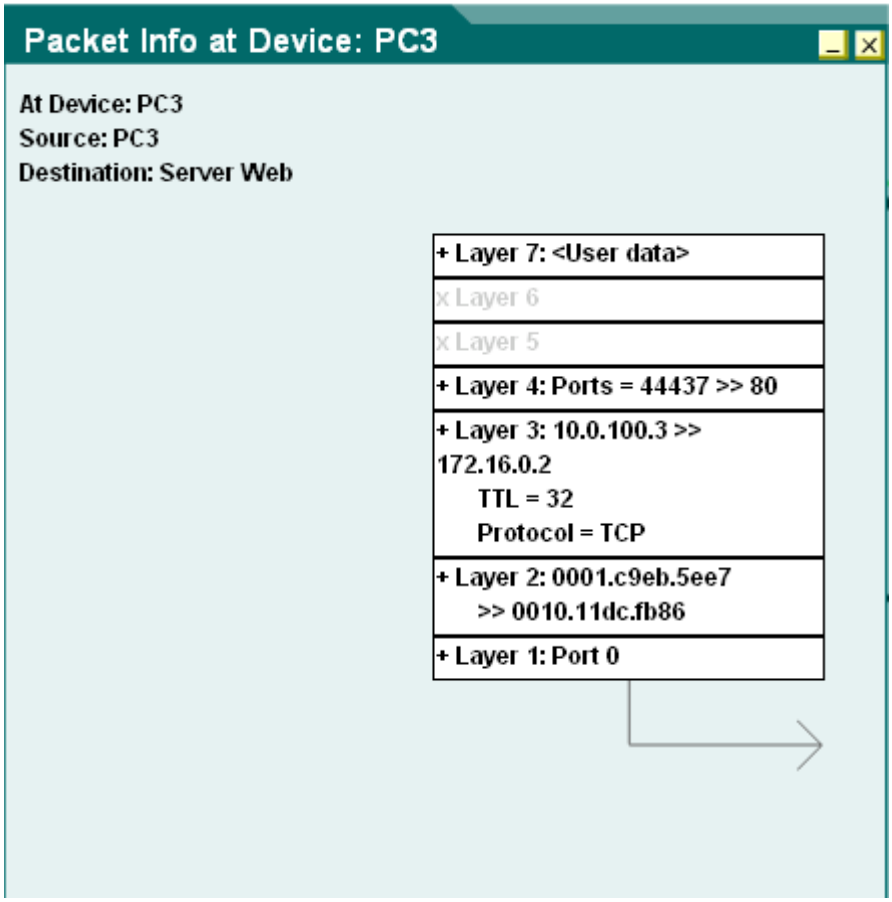


Figura 6.2

Para visualizar la simulación presione play (es un poco lento) o step forward para ir viendo cada salto. De este último modo cada click es un salto del paquete al siguiente dispositivo.