

MÓDULO N°1

BGP: Protocolo de ruteo estándar utilizado para comunicar a los sistemas autónomos que componen internet.

RPKI: es el sistema de certificación para los recursos numéricos que ayuda a garantizar la confiabilidad del ruteo en internet.

Sistemas Autónomos (AS): es un grupo de redes IP que poseen una política de ruteo propia e independiente.

Número de Sistema Autónomo (ASN): Identifica de manera única sus redes dentro de internet. Puede ser de 16 o 32 bits. Se diagraman con nubes o círculos.

Pool central de IP lo tiene "iana" y lo delega a los diferentes "rirs".

Asignación de ASNs

Número de AS/bloque	Asignación
0 y 65535	Reservados
entre 1 y 64495	Internet Pública
entre 64496 y 64511	Documentación para ASN 16 bits – RFC 5737
entre 64512 y 65534	Uso sólo privado
entre 65536 y 65551	Documentación para ASN 32 bits – RFC 5398
entre 65552 y 4294967295	Internet Pública

Rutear: acción de, en base a ciertos criterios de selección, elegir qué caminos seguir para alcanzar un determinado destino.

BGP trabaja en el plano de control.

Los routers de una organización que hablan con otro router de otra organización, se llaman router de borde.

Protocolos de ruteo:

Internos (IGP): IS-IS, RIP, EIGRP, OSPF, entre otros.

Externos (EGP): BGP, esta tabla también llamada LOC-RIB (Local Routing Information Base).

Un router puede manejar todos los protocolos que sean necesarios.

Cada protocolo de ruteo resume sus decisiones en lo que se llama tabla de protocolo.

Cuando se intenta llegar a un determinado destino, el cual está anunciado por dos o más tablas de protocolos, estos compiten entre sí en función a distintas propiedades para determinar cuál es la ruta que prevalece; la ruta elegida pasa a ser parte ahora de la tabla de ruteo.

Existen varias tablas de protocolos dentro de un router, pero una sola tabla de ruteo para cada versión de IP.

Los routers de borde anuncian que redes IP pueden ser alcanzadas a través de ellos. De forma análoga, cuando un router recibe esta información, se dice que aprende.

Cuando un router aprende una ruta, la incorpora en su tabla de BGP. Pero cuando un router anuncia una ruta, no basta con que la ruta esté solo en su tabla de BGP, sino que también debe estar en su tabla de ruteo.

MÓDULO N°2

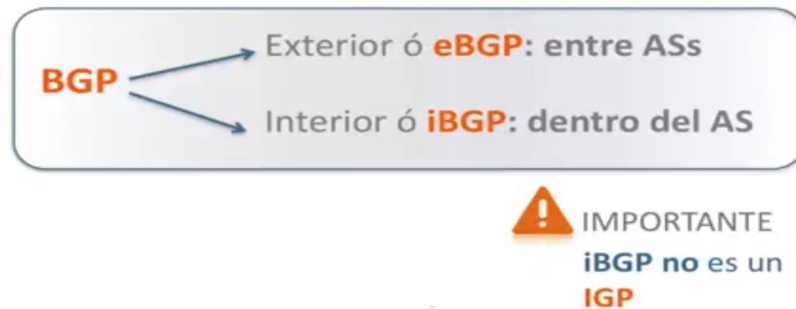
BGP: Border Gateway Protocol o **EGP** Exterior Border Protocol, intercambia información de ruteo y garantiza un camino libre de loop's. Protocolo del tipo "**Path Vector**", tiene en cuenta el camino que hacen determinados paquetes para llegar a destino o considera un vector de Sistemas Autónomos.

RFC's:

- BGP-1: 1989, RFC 1105.
- BGP-4: 1994, RFC 1654 (CIDR y agregados de ruta).
- BGP-4: Hoy, RFC 4271 y RFC 4760 (Multiprotocolo BGP-MBGP). Cuando se implementa en IPv4 y IPv6.

Las "**sesiones**" se establecen entre "**neighbor's**" o "**peer's**" BGP. El puerto utilizado es el TCP 179

Cuando un router aprende rutas, afecta el tráfico saliente y cuando enseña, afecta el tráfico entrante.



BGP es un protocolo diseñado para llevar información de rutas que están fuera de nuestra organización, **NO** para llevar información del ruteo interno.

Buenas prácticas para un ISP.

Información de infraestructura, redes internas -> **IGP**.

Información rutas de clientes -> Dentro del AS **iBGP**.
-> Fuera del AS **eBGP**.

La conexión física:

iBGP: no necesariamente deben estar directamente conectados entre sí los peer's.

eBGP: generalmente están directamente conectados, pero no es un requisito obligatorio.

Tener en cuenta que los “**peer's**” de una potencial “**sesión**” de BGP, deben ser alcanzados por IP desde el otro.

En router cisco, solo se permite un proceso BGP corriendo por router. Lo que significa que un router, sólo puede pertenecer a un AS.

Identificar contra qué neighbor, si está dentro del AS, (pueden estar en cualquier parte del AS) o si está fuera de AS, (por lo general, los router son adyacentes o sea comparten una subred).

Como llevar a cabo una sesión BGP en CISCO:

1- Crear proceso BGP dentro del router: en modo de configuración global,
router bgp <ASN_Local>

2- Identificar el peer contra el cual se quiere levantar la sesión BGP,
neighbor <IP_neighbor> remote-as <ASN_remoto>

-Si se trata de una sesión iBGP ASN_remoto = ASN_local.

-Si se trata de una sesión eBGP ASN_remoto != al ASN_local.

Para verificar lo hecho, se sale del modo de configuración global,
1- show ip bgp summary.

BGP router identifier 198.51.100.1, local AS number 64496
BGP table version is 48347, main routing table version 48347
2558 network entries and 3869 paths using 389968 bytes of memory
527 BGP path attribute entries using 28600 bytes of memory
250 BGP AS-PATH entries using 6304 bytes of memory
1 BGP community entries using 24 bytes of memory
191 BGP route-map cache entries using 3056 bytes of memory
1028 BGP filter-list cache entries using 12336 bytes of memory
BGP activity 14929/71963 prefixes, 42175/38306 paths

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
198.51.100.2	4	64505	38357	31873	48347	0	0	6d08h	1342
203.0.113.2	4	64496	31728	34362	48347	0	0	6d09h	Active

Lo que muestra la salida del comando es un resumen de las sesiones BGP que el router tiene configurado. En este caso, se ve que el router A tiene 2 sesiones BGP configuradas, una con el neighbor 198.51.100.2 y otra con el 203.0.113.2. Una columna importante para visualizar es la que muestra el ASN al que pertenece el neighbor remoto, la tercera columna (recuadrada en azul en la figura). Claramente se puede deducir que la primera sesión es eBGP y la segunda sesión es un BGP interno, pues está levantado contra un neighbor del mismo sistema autónomo. Otra columna muy importante para tener en cuenta sobre la salida de este comando es la última de todas, la denominada State/PrefijosRecibidos (recuadrada en color rojo). En el primer caso, claramente se están recibiendo 1342 prefijos o rutas a través de esa sesión BGP, para que el router A las aprenda, o lo que es lo mismo: las incorpore en su tabla de BGP. En el segundo caso, no muestra la cantidad de prefijos recibidos sino la palabra “Active”.

¿Qué significa que diga “Active” en lugar de los prefijos recibidos?

Por el contrario de lo que pudiera parecer, “Active” en una sesión BGP significa que la sesión está “intentando” establecerse, pero no está establecida. En este caso, debe aparecer un número entero, aunque sea 0, que indica la cantidad de prefijos recibidos.

-2 show ip bgp.

BGP table version is 134358, local router ID is 198.51.100.1

Status codes: s suppressed, d damped, h history, * valid, > best, i – internal, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	192.0.2.0/26	198.51.100.1		0		64496 65511 i
*>	192.0.2.128/26	198.51.100.1		0		64496 64506 i
*>i		198.51.100.22	0	0		64506 e
*	203.0.113.128/25	198.51.100.1		0		64496 64501 i
*>		198.51.100.114	0	0		64501 i
*>	203.0.113.0/25	198.51.100.1		0		64496 ?

Al aplicarlo, se obtendrá un listado del contenido de la tabla de BGP, con las redes ordenadas en forma numérica (recuadro azul).

Las primeras tres columnas listan el estado de cada ruta (recuadro rojo). Un asterisco en la primera columna, indica que la ruta tiene un next-hop válido, el cual se analizará más adelante. Existen más opciones para este campo, las cual son descritas en el encabezado de la salida del comando.

Luego, si la segunda columna tiene un signo mayor, significa que esa ruta fue seleccionada como el mejor camino hacia una red determinada.

Si la tercera columna está en blanco, indica que el router aprendió la ruta de un vecino externo. Una ruta aprendida de un vecino iBGP debería tener una letra i.

La cuarta columna (nuevamente el recuadro azul) lista las rutas que contiene esta tabla de BGP. Cuando no se especifica máscara de subred, se utilizará la máscara classfull (o sea, /8 para las clases A, /16 para las clases B y /24 para las clases C).

Cuando esta columna tiene campos en blanco, significa que se refiere al mismo prefijo que está justo en la fila anterior. Esto es así cada vez que un router recibe, a través de BGP, varios caminos para un mismo prefijo.

Cuando un router anuncia una ruta por BGP, equivale a enseñar a otro router sobre una entrada que el neighbor tiene en su propia tabla de BGP.

Existen diferentes formas de incorporar una ruta a la tabla de BGP:

Las rutas que se incorporan a una tabla de BGP pueden ser:

- Aprendidas de otras sesiones BGP, ya se verá cómo y cuándo se anuncian en cada caso, pero lo importante en este punto es la forma en la que las rutas llegan a formar parte de la tabla de BGP del router.
- Configuradas dentro del proceso BGP el comando redistribute <routing-protocol>, el cual significa que todas las rutas del protocolo indicado en el campo <routing-protocol> serán transferidas dentro de BGP. No es muy recomendable utilizar este comando, sobre todo porque suele perderse el control de los prefijos a anunciar y la administración se vuelve compleja.
- Finalmente, otra forma de incorporar prefijos en la tabla de BGP para que puedan ser anunciados a un neighbor, es a través del comando network, especificando el prefijo a anunciar y su máscara. Es muy importante tener en cuenta que para que el prefijo indicado en el comando network pueda ser anunciado, una ruta coincidente debe estar en la tabla de ruteo.

El prefijo a anunciar deberá estar en la tabla de ruteo para que pueda ser efectivo el comando network. Esto podrá verificarse inspeccionando la ruta a anunciar dentro de la tabla de ruteo, o sea, a través del comando **show ip route**.

```
R1#show ip route
```

```
.....
```

```
192.0.2.0/30 is subnetted, 4 subnets
```

```
C 192.0.2.0 is directly connected, FastEthernet0/1
```

```
O 192.0.2.4 [110/2] via 192.0.2.2, 00:10:04, FastEthernet0/1
```

```
O 192.0.2.8 [110/2] via 192.0.2.13, 00:10:04, FastEthernet0/0
```

```
C 192.0.2.12 is directly connected, FastEthernet0/0
```

Si la ruta no está en la tabla de ruteo, lo indicado en el comando network para esa ruta, será ignorado. Otra consideración importante a tener en cuenta es que cuando un

prefijo se inserta en la tabla de BGP a través del comando network, esto afecta a todas las sesiones BGP que se encuentran configuradas, lo cual debe ser tomado muy en cuenta pues se podría estar anunciando prefijos a neighbors a los que no se les desea anunciar.

Para saber qué rutas se le anuncia a determinado vecino, el comando a ejecutar será: **show ip bgp neighbor <ip_del_neighbor> advertised-routes**. Aquí se ve un ejemplo:

```
Show ip bgp neighbor 192.0.2.51 advertised-routes
BGP table version is 48402, local router ID is 192.0.2.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.0.2.64/26	0.0.0.0		0	32768	i
*> 192.0.2.128/27	198.51.100.3		0		64496 64506 i
*> 192.0.2.160/27	198.51.100.22	0	0		64506 e
*> 192.0.2.192/27	198.51.100.35		100		65550 i
*> 192.0.2.224/27	198.51.100.2		0		65549 i

En forma análoga, para saber cuáles son las rutas que se reciben de un neighbor, se puede utilizar el comando **show ip bgp neighbors <ip_del_neighbor> routes**.

```
Show ip bgp neighbor 192.0.2.51 routes
BGP table version is 48402, local router ID is 192.0.2.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i203.0.113.0/26	198.51.100.1	0	600	0	65549 i
*>i203.0.113.64/26	198.51.100.65	0	600	0	65547 65548 i
*>i203.0.113.128/27	198.51.100.129	0	600	0	65547 i
*>i203.0.113.160/27	198.51.100.161	0	600	0	65548 i
*>i203.0.113.192/27	198.51.100.193	0	600	0	65496 e
*>i203.0.113.224/27	198.51.100.225	0	600	0	65500 ?

MÓDULO N°3

Atributos: son parámetros preestablecidos que viajan junto con más información sobre los prefijos dentro del paquete “UPDATE”. Estos paquetes pueden ser manipulados por el administrador de red para dirigir las decisiones de BGP.

RFC 4271, los atributos se dividen en 4 categorías:

1- Mandatorios bien conocidos (Well-known mandatory): son atributos que tienen que estar si o si, en forma mandatoria, en el paquete UPDATE de BGP. Deben poder ser reconocidos por todas las implementaciones de BGP. Se trata de una forma de asegurar que todas las implementaciones BGP acuerdan un conjunto estándar de atributos.

2- Discrecional bien conocido (Well-known discretionary): son atributos que deben ser reconocidos por todas las implementaciones de BGP, pero pueden o no enviarse en el mensaje UPDATE de BGP.

Además de los atributos bien conocidos, una ruta puede contener uno o más atributos opcionales, lo que significa que las implementaciones BGP

no necesariamente deberán soportarlos. Éstos a su vez se clasifican en transitivos o no transitivos, como puede verse en los dos tipos que siguen:

3- Opcional transitivo (Optional transitive): Si un atributo es transitivo, significa que BGP deberá aceptar y publicar el atributo, aun si este no es reconocido dentro de la implementación.

4- Opcional no transitivo (Optional non-transitive): Si un atributo es opcional no transitivo, significa que el atributo, al ser recibido en el mensaje UPDATE, deberá ignorarse y no pasarse a otros peers BGP. Ahora que se ha explicado el significado de cada categoría, es oportuno ver en qué categoría se encuadra cada uno de los atributos que se utilizan en BGP. Un resumen de ello se muestra en esta tabla:

Attribute Code	Type
1 — ORIGIN	Well-known mandatory
3 — NEXT_HOP	Well-known mandatory
2 — AS_PATH	Well-known mandatory
4 — MULTI_EXIT_DISC	Optional nontransitive
5 — LOCAL_PREF	Well-known discretionary
6 — ATOMIC_AGGREGATE	Well-known discretionary
7 — AGGREGATOR	Well-known discretionary
8 — COMMUNITY	Optional transitive (Cisco)
9 — ORIGINATOR_ID	Optional nontransitive (Cisco)
10 — Cluster List	Optional nontransitive (Cisco)
11 — Destination Preference	(MCI)
12 — Advertiser	(Baynet)
13 — rcid_path	(Baynet)
255 — Reserved	—

Del Foro: “Es muy probable que estés en lo cierto. He estado investigando, y veo que la bibliografía difiere, pero también he encontrado lo que tu dices:

ATOMIC_AGGREGATE **Well-Known Discretionary**

AGGREGATOR **Optional Transitive**

Creo que es importante que lo tengamos en cuenta... gracias por el dato!”

En particular, en este curso se verán los atributos: Origin, Next-hop, AS_PATH, MED (ó Multi_EXIT_DISCriminator), LOCAL_PREF (ó Local Preference), y también se verá WEIGHT.

Origin:

Este atributo informa a los sistemas autónomos como fue introducido el prefijo de red dentro de la tabla de BGP. El atributo admite tres valores: IGP, EGP, e incomplete, representado este último con un “?”. La letra “i” significa que la ruta fue originada en un IGP y luego fue anunciada a través del comando “network”. Cuando se encuentra una letra “e”, significa que la ruta fue originada en un EGP, lo que es equivalente a que pasó de BGP a BGP. Cuando el atributo muestra un “?”, significa que el origen es desconocido. En general esto es consecuencia de la utilización del comando redistribute con algún otro protocolo


dentro de BGP. BGP utiliza este atributo para discriminar entre diferentes rutas a un mismo destino y que el orden de preferencia es "i < e < ?"

>show ip bgp

BGP table version is 134358, local router ID is 198.51.100.1

Status codes: **s** suppressed, **d** damped, **h** history, ***** valid, **>** best, **i** – internal, **S** Stale

Origin codes: **i** - IGP, **e** - EGP, **?** – incomplete



Network	Next Hop	Metric	LocPrf	Weight	Path	
*> 192.0.2.0/26	198.51.100.1			0	64496 65511	i
*> 192.0.2.128/26	198.51.100.1			0	64496 64506	i
*>i	198.51.100.22	0	0		64506	e
* 203.0.113.128/25	198.51.100.1			0	64496 64501	i
*>	198.51.100.114	0	0		64501	i
*> 203.0.113.0/25	198.51.100.1			0	64496	?

Next-Hop:

Indica cuál es el próximo salto para alcanzar el destino. Es muy importante tener en cuenta que indica la IP del próximo salto, pero no la ruta completa. Además, es importante resaltar también que el próximo salto no necesariamente corresponde a un router directamente conectado. Así, el next-hop variará de acuerdo a cómo fue incorporada la ruta. A diferencia de los IGP, que el next-hop suele ser la IP del router que anunció la ruta, en BGP la situación es distinta. Ahora se verá eso: Si se trata de una sesión eBGP, el next-hop es la IP del neighbor que anunció la ruta hacia el sistema autónomo (es decir, nuestro vecino externo). Si se trata de una sesión iBGP hay que diferenciar dos casos:

- Si la ruta fue originada dentro del sistema autónomo, el next-hop será la IP del router que origina la ruta.
- Si la ruta fue introducida al sistema autónomo por eBGP, el next-hop se mantiene inalterado y es la IP del peer de la sesión eBGP. Si al visualizar el next-hop se advierte la IP 0.0.0.0, significa que la ruta se originó localmente en ese router.

>show ip bgp

BGP table version is 134358, local router ID is 198.51.100.1

Status codes: **s** suppressed, **d** damped, **h** history, ***** valid, **>** best, **i** – internal, **S** Stale

Origin codes: **i** - IGP, **e** - EGP, **?** - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.0.2.0/26	198.51.100.1		0	64496	65511 i
*> 192.0.2.128/26	198.51.100.1		0	64496	64506 i
*>i	198.51.100.22	0	0	64506	e
* 203.0.113.128/25	198.51.100.1		0	64496	64501 i
*>	198.51.100.114	0	0	64501	i
*> 203.0.113.0/25	198.51.100.1		0	64496	?

En medios multiacceso, tales como redes ethernet, el next-hop se mantiene inalterado y es la IP de la interfaz del router que originó la ruta. Esto permite que si un router re-anuncia rutas aprendidas de otro, como en el caso de un route-server, el next-hop siga siendo la IP del router que anunció la ruta originalmente. Esto es particularmente útil en un IXP, ya que de esa forma se evita que el tráfico pase por el servidor de rutas y éste sólo procesará la información de BGP, pero no estará involucrado en el forwarding de los paquetes.

AS_PATH:

Este atributo guarda como información el número de cada uno de los sistemas autónomos que atravesó el anuncio de una ruta hasta llegar al AS local. Dicho de otra forma, es la secuencia de ASs que se deberán atravesar para llegar al AS del destino.

Este atributo suele ser clave para la elección del mejor camino. Otra particularidad a tener en cuenta es que un AS_PATH en blanco está indicando que la ruta fue originada en el AS local.

>show ip bgp

BGP table version is 134358, local router ID is 198.51.100.1

Status codes: s suppressed, d damped, h history, * valid, > best, i – internal, S Stale

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.0.2.0/26	198.51.100.1		0		64496 65511 i
*> 192.0.2.128/26	198.51.100.1		0		64496 64506 i
*>i	198.51.100.22	0	0		64506 e
* 203.0.113.128/25	198.51.100.1		0		64496 64501 i
*>	198.51.100.114	0	0		64501 i
*> 203.0.113.0/25	198.51.100.1		0		64496 ?

Note que la forma de interpretar el camino que ha realizado la ruta es “de atrás para adelante”, o sea, el primer sistema autónomo que se ve es el último que la ruta atravesó antes de llegar al AS local, el cual no será mostrado.

MED:

Multi Exit Discriminator. MED es un atributo que se utiliza para indicar a los vecinos eBGP cuál es la preferencia acerca del tráfico entrante a nuestro sistema autónomo. Algo así como indicar cuál es la puerta de entrada de nuestro AS que será la preferida en caso de tener varias sesiones eBGP. Como se ve, MED es utilizado en los casos en los que existe una conexión “Multihomed”, o sea cuando una organización, tiene conexión con más de un sistema autónomo.

Cuando una misma ruta es recibida de dos sesiones eBGP, se preferirá aquella que tenga el valor de MED más bajo. Este atributo no es de los que más se consideran en el proceso de selección de rutas, pues antes que evaluar el MED hay otros atributos que resultan más relevantes, y eso se verá más adelante.

Por otro lado, existen implementaciones más eficientes a la del MED, como ser el uso de COMMUNITY, aunque ese tema escapa al alcance de este curso. La forma de visualizar

el valor de este atributo en la salida del comando “show ip bgp” es identificando la columna “Metric”.

LOCAL_Preference:

Este atributo, cuanto mayor es su valor, mayor es el grado de preferencia entre rutas a un mismo destino. La particularidad que tiene es que se trata de un atributo local al sistema autónomo, lo que significa que se propaga por iBGP, pero no por eBGP. Otra característica es que asume por default el valor 100, y que para configurarle otro valor se hace a través del comando “route-map”.

Weight:

Este atributo es similar al Local_Preference, donde el valor más alto es tenido en cuenta a la hora de decidir, pero lo diferencia que su valor no se transfiere dentro del sistema autónomo, sino que es sólo local al router que lo define. Este atributo es muy importante porque tiene alta precedencia a la hora de elegir una ruta por el algoritmo de decisión de BGP. Weight resulta muy útil cuando se necesita discriminar rutas de proveedores conectados a un mismo router. Al igual que Local_preference, también puede ser configurado a través de route-map.

Importante: este atributo era inicialmente propietario de routers Cisco, pero ya otros fabricantes han incorporado atributos similares.

La salida del comando show ip bgp mostrará los valores del Weight en la columna correspondiente:

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.0.2.0/26	198.51.100.1	95		300	64496 65511 i
*> 192.0.2.128/26	198.51.100.1	95		0	64496 64506 i
*>i	198.51.100.22	95		100	64506 e
* 203.0.113.128/25	198.51.100.1	95		0	64496 64501 i
*>	198.51.100.114		100	0	64501 i
*> 203.0.113.0/25	198.51.100.1	0	100	0	64496 64497 64498 i

Importante: De los atributos mostrados en estos ejemplos, el más utilizado es Local_preference, ya que es un atributo parte del estándar de BGP y por lo tanto está en todas las implementaciones. Además, está entre los primeros a ser evaluados en la selección del mejor camino, que se verá más adelante, por lo que su utilización brinda una posibilidad simple de alterar la preferencia de rutas en BGP. Por esta razón, es una práctica común en muchos ISPs dar a sus clientes información acerca de cómo manejan el Local-Preference en sus redes.

MÓDULO N°4

Incorporar rutas a la tabla BGP se lo puede hacer mediante:

- 1- Aprendidas por BGP.
- 2- Redistribute.
- 3- Comando Network. (La ruta tiene que estar en la tabla de ruteo.)

Si no hay ruta estática ni ruta aprendida por un IGP, que coincida con la ruta que se quiere anunciar con el comando network?. En esos casos, habrá que acudir a algún método que permita generar una ruta para agregar a la tabla de ruteo.

La forma de resolver esta situación es mediante la incorporación de la siguiente sentencia basada en el ejemplo ya mostrado:

```
ip route 203.0.113.128 255.255.255.128 null0  
router bgp 64496  
network 203.0.113.128 mask 255.255.255.128
```

Se trata de incorporar en la tabla de ruteo una ruta estática a “null 0”, es decir, una ruta que no apunta a un next-hop sino que descarta los paquetes. Este tipo de ruta se denomina “pull up”, y sólo será utilizada si no existe una ruta más específica para la tabla de ruteo. Por lo general, lo que se hará es insertar este tipo de rutas para prefijos sumariados: de esa forma, si existen prefijos más específicos en la tabla de ruteo, serán utilizados. Si no existe ningún prefijo específico para alguna porción de la ruta sumariada, se utilizará la ruta a “null 0” y los paquetes serán descartados.

Después de lo que hemos visto, resulta natural que el modo de anunciar la ruta default sea a través del comando “**network 0.0.0.0**”. Esto claramente puede hacerse y es correcto, pero habrá que tener en cuenta dos cosas. Una es que se estará anunciando el default a todos los neighbor, y la otra es que deberá existir la misma ruta, sí o sí, en la tabla de ruteo para que pueda ser anunciada. Otra opción es usar el comando “**default-information originate**”. Al igual que en el caso anterior, la ruta default será anunciada a todos los neighbors, pero este comando tiene la particularidad que anunciará la ruta default sin necesidad de que la ruta esté en la tabla de ruteo. Finalmente, otra opción posible es utilizar el comando: **neighbor <IP_Neighbor> default-originate**. En este caso, la ruta default solo se anuncia a un neighbor, el indicado en el comando, y al igual que en el comando “**default-information originate**”, tampoco es necesario que la ruta esté previamente en la tabla de ruteo para que pueda ser anunciada.

Hay varias razones para querer controlar el tráfico entrante y saliente del AS, y algunas podrían ser no tan obvias. Por ejemplo,

- Técnicamente es importante controlar la cantidad de rutas que los routers manejan, pues todo incide en la capacidad de procesamiento, uso de la memoria, entre otros.
- Además de esto, un ISP por ejemplo, estará interesado en que las rutas que envía y recibe de un cliente sean las acordadas, por cuestiones económicas y contractuales. De la misma manera, existen costos asociados a utilizar un sistema autónomo como AS de tránsito.
- Todo esto sin mencionar que la seguridad que se debe brindar a los clientes con las rutas que les anunciamos es crucial para dar un buen servicio.

Existen dos grandes tipos de filtros en BGP: los filtros basados en direcciones IP y los filtros basados en el path. Cuando se refiere a los filtros basados en direcciones IP, hay dos subgrupos. Aquellos denominados “**distribute-list**”, los cuales se implementan con access-list. Y por otro lado, están aquellos denominados “**prefix-lists**”, cuya implementación es más reciente que la de distribute-list.

Dentro de los filtros basados en el path, están los denominados “**filter-list**”, los cuales analizan y aplican los filtros de acuerdo a la información que contiene el atributo AS_Path.

Importante: se debe tener en cuenta que los filtros pueden ser aplicados a la entrada o a la salida de una sesión BGP. Si se aplican a la entrada, entonces estarán afectando lo que se aprende del neighbor. En cambio si se aplican a la salida, los prefijos afectados serán los que se anuncian al vecino.

El primer paso es crear el prefix-list que se aplicará luego al vecino. Para ello, y aunque no se verán filtros implementados con access-list, será útil repasarlas, y ver las diferencias con el prefix-list.

Como recordarán, las access-list tiene esta forma:

access-list <nro_access-list> permit|deny ip <prefijo> <máscara de wildcard>

Las listas de acceso se construyen con múltiples sentencias como la indicada, donde cada sentencia especifica qué se quiere filtrar o dejar pasar.

Access-list (ACL)	Prefix-list
<code>access-list <nro_access-list> permit deny ip <prefijo> <máscara de wildcard></code>	<code>ip prefix-list <nombre_prefix-list> <nro_seq> permit deny <red/prefijo></code>
<code>access-list 101 deny ip 10.0.0.0 0.255.255.255</code> <code>access-list 101 deny ip 203.0.113.0 0.0.0.255</code> <code>access-list 101 permit ip 192.0.2.0 0.0.0.255</code>	<code>ip prefix-list Entrada seq 5 deny 10.0.0.0/8 le 32</code> <code>ip prefix-list Entrada seq 10 deny 203.0.113.0/24 le 32</code> <code>ip prefix-list Entrada seq 15 permit 0.0.0.0/0 le 32</code>

Por ejemplo, en la lista de acceso 101 que se muestra a continuación, se especifican dos sentencias, las cuales están indicando que se permitirá el anuncio o el aprendizaje de los prefijos 10.0.0.0/8, 203.0.113.0/24 y 192.0.2.0/24, dependiendo de si la ACL se aplica a la entrada o a la salida de una sesión BGP:

Estas sentencias se van ejecutando en forma secuencial, tal como fue creada la ACL, con el inconveniente de que si se quiere agregar o quitar alguna de ellas, se debe borrar y volver a crear toda la ACL con la modificación. Los prefix-list en cambio tienen la siguiente forma:

ip prefix-list <nombre_prefix-list> <nro_seq> permit|deny <red/prefijo>

Las ventajas son varias: entre ellas, que pueden ser representados por un nombre, lo cual los vuelve más legibles e intuitivos. Además, la posibilidad de identificar a cada sentencia con un número de secuencia permite que realizar una modificación en el prefix-list sea más fácil pues sólo se remueve la sentencia a modificar, identificada con su número de secuencia. Pero la ventaja más significativa es que las rutas que se quieren permitir o denegar, se expresan con el formato de red y prefijo de subred, lo cual los hace más amigables en su lectura. Un detalle que no es menor es que estas implementaciones permiten que, a través de unos modificadores llamados “**ge**”, los cuales significan “mayor o igual”, y “**le**”, que significan “menor o igual”, las longitudes de los prefijos sean variables.

Un ejemplo de prefix-list es el que se ve en la columna de la derecha del cuadro, donde el nombre elegido para identificar al prefix-list es “Entrada”, pues se especificará en ese prefix-list todo lo que no se quiere permitir que entre a nuestro AS.

En este caso se estarán denegando todos los prefijos de entrada de la red 10.0.0.0 de longitud entre 8 y 32, es decir, denegamos toda la red privada 10.0.0.0; También se filtran los prefijos de la red 203.0.113.0/24 menores o iguales a 32. Por último, se permite todo el resto de los prefijos. Esta última línea es necesaria ya que de lo contrario existe un “deny” implícito al final del prefix-list.

Para aplicar un filtro de rutas en BGP basado en prefix-list deberá configurarse de la siguiente manera:

neighbor <ip-address|peer-group> prefix-list <nombre_prefix-list> in|out

Simplemente se indicará a quién se aplica el filtro, cuál es el prefix-list que se aplica, y si se aplicará sobre las rutas que vienen como información de ese neighbor, en cuyo caso será in, o en lo que se le enseña al neighbor, que será out.

Importante: Notar también que se puede aplicar el filtro a un determinado neighbor o a un grupo de neighbors que fueron declarados previamente en un “peer-group”, de la siguiente manera:

neighbor <nombre_peer-group> peer-group

neighbor <IP_neighbor> peer-group <nombre_peer-group>

Repitiendo esta última sentencia por cada neighbor que se desee hacer pertenecer al peer-group.

Hasta ahora los filtros que se vieron filtraban rutas indicando los prefijos o las direcciones IP. En este nuevo caso los filtros actuarán según información del AS_Path, o dicho de otra forma, del camino que realizan los prefijos. Esto es así porque muchas veces es más simple filtrar por la procedencia de las rutas, que filtrar ruta a ruta. Para implementar filtros basados en AS_path, se considerarán dos pasos bien diferenciados.

- **Dos pasos:**

1. Crear sentencia con expresión regular.

```
ip as-path access-list <nro_filtro> permit|deny <regex>
```

2. Aplicar el filtro

```
neighbor <IP_neighbor> filter-list <nro_filtro> in|out
```

Por un lado (paso 1), se deberá crear una expresión regular que denote un AS_Path. Notar que hay un número de filtro que deberá mantenerse para indicar que se trata de sentencias dentro del mismo filtro. Al final del comando se deberá indicar la expresión regular que denota lo que queremos controlar con el filtro.

El paso siguiente (paso 2), es aplicar el filtro con la expresión regular, a la entrada o salida de una sesión BGP, donde se debe especificar el neighbor al cual se le aplica el filtro, el número de filtro que se le está aplicando (según lo que se creó previamente), y si se aplica sobre las rutas que se aprenden de un neighbor, o las que se anuncian a ese neighbor, en cuyo caso habrá que especificar con la palabra “out” en vez de “in” como en el primer caso.

Para poder leer con facilidad qué quiere decir cada filtro de este tipo, hay que comprender las expresiones regulares. Algunas de ellas, las más comunes son las que se muestran en la siguiente tabla.

Caracter	Función
^	empieza con
\$	termina con
.	cualquier caracter
_	cualquier delimitador (espacio, comienzo, fin, coma)
[0-9]	rango del 0 al 9
[123]	1, 2 ó 3
()	asocia
	ó
*	cero o más veces
?	cero o una vez
+	una o más veces
\#	llama a la expresión ubicada en la posición # del regexp

Los **route-maps** son sentencias que se construyen en forma similar a un lenguaje de programación. Cada sentencia se identifica con un número de secuencia, lo cual facilita sus modificaciones, pero lo más importante es que el número de secuencia implica el orden en el que cada sentencia será ejecutada, en forma ascendente.

La ejecución del route-map avanza hasta que una de las sentencias da verdadero, y se ejecuta. Se verá además que existe una palabra clave dentro del route-map, la palabra “match”, la cual será el disparador para permitir o denegar una acción.

Si la palabra “match”, por su naturaleza de condicionante, no está dentro de ninguna sentencia del route-map, entonces todas las rutas resultan con criterio verdadero y el comando “set” se aplicará a todas ellas. Lo mismo ocurre si tampoco existe en él una lista de acceso para la sentencia “match”. Como consideración final, si existen en el route-map varias sentencias “match”, todas deberán ser verdaderas para que el mismo resulte verdadero y se pueda tomar una acción. Al igual que ocurre con las listas de acceso, una denegación está implícitamente incluida al final del route-map.

MÓDULO N°5

La confiabilidad sobre la información de ruteo en Internet

Internet ha funcionado desde siempre gracias a una gran cadena de confianza que se establece entre los operadores de las redes para que el ruteo pueda llevarse a cabo lo mejor posible. Dicho de otra manera, la confianza en Internet radica en que cada organización anuncie sólo sus propios prefijos, o los prefijos de las organizaciones a las que le da tránsito. Sin embargo, eso no está garantizado en BGP, sino que se basa en el buen trabajo que hagan los operadores de las diferentes redes.

Para intentar hacer más confiable la información que se recibe de los neighbors, una organización puede tomar medidas como ser: valerse de algún IRR para recibir y anunciar rutas, proteger a los dispositivos routers para que no puedan ser accedidos en forma indebida, implementar la autenticación entre peers, entre otras medidas. Otra medida que complementa y ayuda al trabajo de los operadores de redes es la implementación de filtros basados en el RFC1918, el cual describe a las redes que son de usos privados y/o reservados, y que no deben ser anunciados a Internet, como por ejemplo las redes de uso sólo en LAN, las redes para multicast, los bloques para documentación, entre otras.

Cuando algún dispositivo en el routing de Internet anuncia un prefijo que no le corresponde anunciar por no estar autorizado a hacerlo, ya sea porque no es propio o porque no es parte de los sistemas autónomos a los que le da tránsito, se produce lo que se denomina “Secuestro de rutas”, o “route hijacking”.

Secuestro de rutas

Una situación distinta se produce cuando el camino se ve afectado no porque un AS anuncie un prefijo que no tiene, sino porque haga un “mal anuncio”, como podría ocurrir en un AS de tránsito. Muchas veces a estos eventos se le llama “**route leak**”, y aunque no

hay un claro consenso sobre eso, es interesante que se vea el RFC7908, pues puede ayudar a ver las diferentes posturas respecto a la definición del término.

Existen otras acciones que atentan contra el buen ruteo en Internet. Por ejemplo, aquellas que se manifiestan como ataques con el camino o path. Una técnica para realizar este tipo de ataques es la de insertar en el mensaje UPDATE, no solo el propio AS sino más números de sistemas autónomos. Otro tipo de ataque al camino consiste en originar una ruta para determinado prefijo, pero, hacerlo utilizando un ASN que no es el que corresponde que origine rutas para ese prefijo.

Diferentes medidas de protección

Una de ellas sería indicarle a sus upstream u organizaciones con las que mantiene acuerdos de peering, cuáles son los bloques IP que se les va a anunciar, como una suerte de acuerdo previo que permita implementar filtros a medida. Esto puede hacerse de una manera informal, cuando los operadores se conocen, mediante un e-mail o un llamado telefónico. Pero cuando estas organizaciones se interconectan con otras y ya no hay tal posibilidad, ¿cómo verificar que los anuncios que se reciben de un neighbor son correctos?. Otra forma es que se dé a conocer entre los peers la base de datos de recursos de Internet que se utilizará para verificar la información de pertenencia de los recursos, tal como se mencionó en el tema anterior, y que puede hacerse a través del servicio de Internet Routing Registry o IRR. En estas bases de datos las organizaciones listan, no sólo los recursos que poseen, sino también la política de ruteo que implementan: cuáles son sus peers, qué anuncian a cada uno y qué se acepta recibir de ellos, entre otros datos. Con toda esta información se puede automatizar el filtrado y configuración de políticas de entrada en los routers. Una de las bases de datos IRR más conocidas es RADB; a su vez, los RIRs suelen implementar IRRs. Al día de hoy, LACNIC no cuenta con una base de datos de este tipo. Además de esto, complementan la medida que, independientemente de la información suministrada por el peer, los operadores de redes utilicen información de los servicios de “whois” con el fin de verificar el derecho de uso del recurso.

No obstante, ninguna de estas fuentes provee información firmada, ni mecanismos que garanticen la autenticación de derecho de uso. Por tal motivo se dice que la integridad de Internet depende de la confianza entre quienes la operan, la confianza que se pueda generar entre los administradores de redes de todos los sistemas autónomos que componen la gran red de redes. Algunos operadores reemplazan esta ausencia de mecanismos de firma de la información por cartas escritas por parte de la organización, llamadas LoA, firmadas de puño y letra. Esto puede ser fácilmente falsificado y no resulta de gran utilidad en la automatización. No obstante, existe una propuesta de solución que poco a poco va ganando terreno entre las implementaciones de seguridad en redes, que se denomina RPKI.

Certificación de recursos

RPKI, por sus siglas en inglés de Resource Public Key Infrastructure, propone un mecanismo para validar que el sistema autónomo que origina una ruta sea el que está autorizado a hacerlo.

Importante: RPKI sólo previene problemas que devienen del origen de la ruta, no de otros ataques.

RPKI, tal como se mencionó, permite que se valide el derecho de una organización a utilizar un recurso como ser IPv4, IPv6 o ASN. Básicamente lo que hace es combinar la jerarquía del modelo de RIRs con el uso de certificados digitales basados en el estándar X.509, los cuales tienen una extensión para soportar IPs y ASNs (para el caso, ver RFC 3779). Punto importante a destacar: RPKI es un estándar de IETF, según los RFC 6480 - 6492.

Generar una PKI sobre los recursos, donde la raíz de la cadena de confianza son los RIRs (por ejemplo como LACNIC en nuestra región) y mediante certificados digitales se construye una cadena de firmas hasta certificar los recursos de una organización.

Una vez que la organización tiene sus recursos firmados, podrá crear unos objetos llamados ROA (del inglés Route Origin Authorizations), firmados digitalmente, que proveen una autorización explícita por parte del poseedor de un prefijo permitiendo a un ASN originar rutas de ese rango. En fin, usando certificados podemos crear objetos que identifiquen el origen de un prefijo. Los ROAs contienen información sobre el AS de origen permitido para un conjunto de prefijos. Son firmados usando los certificados generados por RPKI, y luego los ROAs firmados son copiados al repositorio.

¿Qué información tienen o declaran esos ROAs? Básicamente un conjunto de prefijos, con su longitud y máximo permitido (/20 a /24 por ejemplo) y el ASN de origen de estos (también el período de validez del ROA).

Con esto, tenemos una fuente de información firmada digitalmente contra la cual podemos contrastar los UPDATEs que recibimos por BGP: si el ASN de origen de un prefijo en BGP coincide con un ROA generado por el poseedor de ese prefijo, entonces ese anuncio será correcto, o sea "válido". De lo contrario, será inválido. Puede suceder también que no exista un ROA que cubra ese prefijo, en ese caso tendremos un tercer estado que será incompleto o "not found" (este sería el caso de los prefijos de organizaciones que no han implementado RPKI).

Con todo lo mencionado, se está en condiciones de definir cómo está compuesta la solución de RPKI, y es de la siguiente manera:

- PKI (Public Key Infrastructure) de recursos, los cuales estarían integrados por las direcciones IP y ASN asignados, más los certificados de pertenencia otorgados por el RIR (y otra información de PKI).
- Objetos firmados digitalmente para soportar seguridad del enrutamiento, o sea los ROAs.
- Un repositorio distribuido que almacena los objetos PKI y los objetos de enrutamiento firmados. Este repositorio es público y puede ser copiado localmente por organizaciones que quieran utilizarlo (o sea, como si fuera un cache).

La solución obtiene su cierre cuando se lleva a cabo la "Validación".

La validación es el proceso automatizado que llevan adelante los ISPs u organizaciones que quieren contrastar los anuncios de BGP, o sea los UPDATE, contra la base de RPKI, o sea, los repositorios, y de esa forma verificar que los anuncios tienen una procedencia autorizada. Durante el proceso de validación, los ISPs interesados utilizarán:

- Las propiedades del cifrado de clave pública (o sea, los certificados)
- Las propiedades de los prefijos.

En resumen, ¿qué debería hacer una organización que desea implementar RPKI para validar el origen de sus rutas?

En primer lugar, debería generar certificados digitales sobre sus recursos IPv4, IPv6 y ASN.

Luego generar los ROAs correspondientes a los anuncios de BGP que va a realizar. Esta acción es la que más cuidado merece, pues un ROA mal generado, que no contemple todos los prefijos podría invalidar a alguno de estos.

Una vez que la organización hizo estos dos pasos, ya cuenta con una certificación digital para los anuncios BGP sobre sus recursos.

Luego, las organizaciones que deseen utilizar la validación sobre los recursos de quien ya implementó RPKI podrá hacerlo, y con ello asegurarse la procedencia de las rutas, pero esto es un proceso independiente a la organización que firma sus propios recursos.

MÓDULO N°6

Selección del mejor camino

Sigue lo que se llama un algoritmo de decisión. BGP irá evaluando sentencias del algoritmo una a una, en forma secuencial y respetando el orden, y cuando una de sentencias pueda marcar la diferencia entre las rutas, la decisión sobre la mejor ruta estará tomada y el algoritmo se detendrá.

Lo primero que evaluará BGP es naturalmente si el next-hop indicado en la ruta es accesible a través de IP, pues si no lo es, la ruta automáticamente se descarta.

Si se tienen varias rutas a un mismo destino y todas son con next-hop alcanzables, entonces BGP evaluará si alguna de las rutas fue aprendida por iBGP, la opción de “synchronization” está habilitada, y la ruta no está en la tabla de alguno de los protocolos de ruteo interno. En esos casos la ruta es descartada. Vale la pena aclarar que hoy en día, y desde hace bastante tiempo ya, la opción de synchronization viene deshabilitada por default en la mayoría de los sistemas operativos de los router, por lo que se ha decidido no incluirlo en este curso. No obstante, merece atención sobre todo debido a que tiene alta precedencia en el algoritmo de decisión, por lo que se aconseja su lectura en bibliografías de referencia.

Si las rutas no son descartadas en el paso anterior, entonces BGP continuará con el proceso de evaluación.

Si estuviera configurado el atributo Weight, seguiría por éste y elegirá la ruta con mayor peso. Si el atributo Weight no logra decidir, entonces BGP evaluará nuevamente y elegirá la ruta de mayor Local_Preference. Si el Local_Preference es el mismo, entonces BGP preferirá la ruta que haya sido originada en el router, ya sea por el comando network o con redistribution.

Si ninguno de los pasos anteriores se cumple, BGP evaluará cuál es el AS_Path más corto. Notar que recién aquí BGP tiene en cuenta la longitud del AS_Path, que da una idea de la distancia a la que se encuentra el destino (los AS por los que hay que pasar). En muchos casos este es el criterio por el cual se definirá la mejor ruta, si no se ha hecho uso de Weight o Local_Preference.

Si aún así no puede decidirse la mejor ruta, entonces se evaluará el código de origen y se elegirá el de menor valor, teniendo en cuenta que IGP tiene menor valor que EGP, y a su vez éste es preferido antes que una ruta de origen desconocido o “incomplete”. Si el origen de las rutas es el mismo, ha llegado la hora de que BGP evalúe el atributo MED. Es

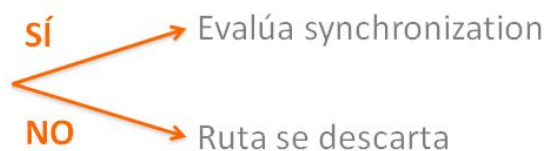
importante aclarar que para que este atributo sea evaluado, los neighbors desde donde se aprendieron las rutas deben pertenecer al mismo sistema autónomo, a no ser que se especifique lo contrario en la configuración de BGP con el comando “bgp always-compare-med”. El menor valor de MED es preferido. Llegada esta instancia de decisión, se preferirán las rutas que se aprendan por eBGP, a las aprendidas por iBGP.

Si aún no se ha podido tomar una decisión, BGP preferirá la ruta cuyo Next-Hop tenga la menor métrica en el IGP. Es difícil que se tenga que llegar a este punto de decisión porque en general en cualquiera de las instancias anteriores se puede decidir cuál es la mejor ruta. No obstante, si se llegara hasta acá, BGP decidiría en base a la ruta que tuviera menor “router-ID”, que es la mayor IP de las interfaces de un router.

Gráficamente se puede sintetizar de la siguiente manera:

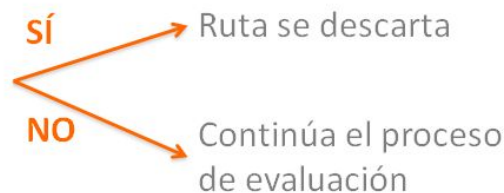
1. Evaluación del NH

El NH es alcanzable?



2. Evaluación de synchronization

- iBGP
- Synch habilitado
- No hay entrada en la tabla de ruteo



3. Evaluación del Weight

Weight
 \neq



4. Evaluación del Local_Pref

Local_Pref
 \neq



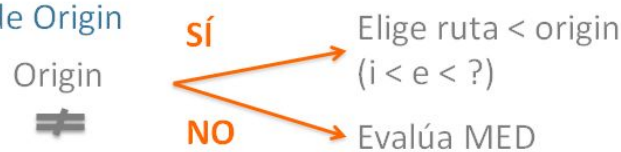
5. Origin de la ruta



6. Evaluación del AS_Path



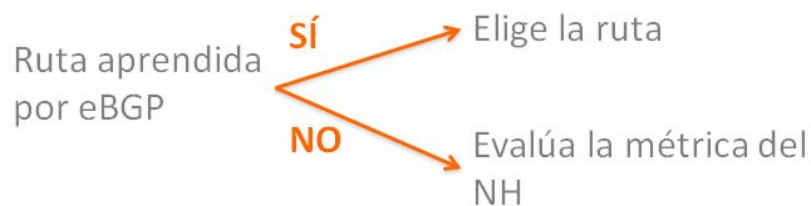
7. Evaluación de Origin



8. Evaluación del MED



9. Evaluación de cómo fue aprendida la ruta



10. Evaluación de la métrica del NH



En primer lugar, es muy importante considerar los roles de los IGP y el del BGP. La recomendación es que siempre se utilice un IGP para llevar información de las rutas de infraestructura de la organización, y que se utilice BGP para las rutas de Internet y las de clientes, discriminando entre iBGP y eBGP de la siguiente manera:

- eBGP naturalmente para las sesiones con neighbors pertenecientes a otros AS.
- iBGP utilizarlo para transportar:
 - Los prefijos de Internet a través del backbone.
 - Los prefijos de los clientes.

Es importante también tener en cuenta la sumarización de prefijos y no desagregar más de lo necesario. Hacia afuera de nuestro sistema autónomo sólo se debería mostrar la información de prefijos sumarizada, no desagregando las redes a menos que haya alguna razón realmente válida.

Otra buena práctica consiste en utilizar direcciones /32 para las interfaces de loopbacks y levantar las sesiones de iBGP con ellas.

El uso de “Peer Groups”, para agrupar vecinos con las mismas características de ruteo, por ejemplo: clientes, proveedores, tránsito, etc.

Para mitigar el robo de prefijos, algo que se puede hacer fácilmente es utilizar passwords en la sesiones de BGP.

Entre lo que se recomienda como buena práctica NO hacer, está:

- Redistribuir BGP dentro de un IGP.
- Redistribuir prefijos IGP dentro del BGP.
- Usar IGP para transportar prefijos de clientes o redes externas.

Finalmente, se verán qué prefijos no se debería aceptar recibir de ningún neighbor, ni anunciarlos:

- Prefijos definidos en el RFC1918. Ver cuáles son en la URL:
<https://tools.ietf.org/html/rfc1918>
- Los prefijos propios de la organización (para evitar loops).
- La ruta por defecto, a no ser que se requiera.
- Prefijos mayores a /24 (o sea, bloques de redes pequeñas como /25, /26, etc.).

A continuación, se muestra un ejemplo de un filtro implementado con prefix-list para ser implementado a la entrada de una sesión BGP:

```
ip prefix-list in-filter deny 0.0.0.0/0 ! Block default
ip prefix-list in-filter deny 0.0.0.0/8 le 32
ip prefix-list in-filter deny 10.0.0.0/8 le 32
ip prefix-list in-filter deny 101.10.0.0/19 le 32 ! Block local prefix
ip prefix-list in-filter deny 127.0.0.0/8 le 32
ip prefix-list in-filter deny 169.254.0.0/16 le 32
ip prefix-list in-filter deny 172.16.0.0/12 le 32
ip prefix-list in-filter deny 192.0.2.0/24 le 32
ip prefix-list in-filter deny 192.168.0.0/16 le 32
ip prefix-list in-filter deny 224.0.0.0/3 le 32 ! Block multicast
ip prefix-list in-filter deny 0.0.0.0/0 ge 25 ! Block prefixes >/24
ip prefix-list in-filter permit 0.0.0.0/0 le 32
```

También, en el caso de ISP es importante aplicar las reglas de la BCP38 (RFC 2827), a fin de no permitir spoofing de direcciones. Para ver una introducción al tema referirse a: www.bcp38.info Como nota final en este tema, para quienes estén interesados en contribuir a la mejora del ruteo global, pueden ver en <https://www.routingmanifesto.org> el documento MANRS: Mutually Agreed Norms for Routing Security, que es una iniciativa para despertar conciencia y promover acciones en los ISPs y otras organizaciones para un ruteo más seguro y estable en Internet.

Resumen de lo visto

Llegando al tercer tema del sexto módulo, se resumirán los principales conceptos que se han visto en este curso de BGP.

BGP es un protocolo de ruteo externo o EGP que se utiliza para la comunicación de sistemas autónomos en Internet.

Fue concebido para llevar información de las rutas externas a nuestro AS, ya sea dentro o fuera del mismo (iBGP o eBGP).

Trabaja aprendiendo y anunciando rutas a través de sesiones con routers que se denominan “neighbors”, que se deben configurar explícitamente.

Para saber cuál es la mejor ruta utiliza un algoritmo de decisión basado en atributos.

Una administración controlada de los prefijos que se anuncian y reciben a través de BGP supone la implementación de políticas en las sesiones, tanto en la entrada como en la salida de las mismas.

Para eso se vieron los filtros por IP y por AS_Path, los route-maps y una gran cantidad de atributos.

BGP se despliega en toda la Internet gracias a una gran cadena de confianza entre los administradores de las redes. No obstante, diferentes formas de ataques pueden producir caminos indeseados del tráfico de Internet, como el secuestro de rutas o los route leaks.

Para mitigar esto existen diferentes medidas que se pueden tomar, pero la principal de ellas, por aportar la garantía del origen de las rutas, es RPKI.

Finalmente existe documentación y RFCs que ayudan a llevar adelante implementaciones de BGP con buenas prácticas y recomendaciones para que nuestra organización también sea parte la comunidad que vela por el buen crecimiento de Internet.