Global Information Assurance Certification Paper

Border Gateway Protocol - The Language of the Internet

Yvonne Tracy

April 10, 2002

Introduction

This paper is an examination of Border Gateway Protocol. The intent is to introduce to the reader the concepts involved with routing across the Internet and examine a sample BGP configuration on a Cisco router. The sample will include only the statements that directly apply to the BGP configuration and will not include any interfaces, IP routes, or information pertaining to any other routing protocol, although all of this information would be necessary to have a functioning BGP router. The study of BGP is much larger than this paper will detail. In conclusion there will be a consideration of S-BGP, an option currently underway to increase the level of security on the Internet.

BGP – a necessity, but is it for you?

Any data traversing an internet, or functional collection of networks, must be routed. An integral part of the routing mechanism is the language or protocol the routers speak to each other. There are a number of routing protocols in use, some, such as IGRP (Interior Gateway Routing Protocol) and EIGRP (Enhanced Interior Gateway Routing Protocol) were written by a vendor specifically for use with their equipment. This type of routing protocol is usually referred to as vendor specific.

Other routing protocols are not specifically written for one vendor's equipment. These protocols are usually called standards based protocols, and are often written or authorized by one of the groups designing and maintaining worldwide standards. OSPF (Open Shortest Path First) for example was designed and written by a subgroup of the Internet Engineering Task Force (IETF) because RIP (Routing Information Protocol) was no longer a viable protocol for large networks.

None of these afore-mentioned protocols, however, could scale to route the entire Internet. Vendor specific protocols cannot be used even if they could scale, as the Internet does not use a single vendor's products. RIP, has a maximum hop count of fifteen, or a total of fifteen routers that it can traverse from the source of a packet to its destination. The Internet is worldwide. A mere fifteen hops would not adequately enable communication. An OSPF router must have the capacity to map the entire network for which it routes. No router would be able to contain the topology of the entire world's network. As can be seen, none of these routing protocols were adequate for the Internet, a worldwide connection of networks. In 1984 an Exterior Gateway Protocol (EGP) was developed. It was designed to "exchange net-reachability information between Internet gateways belonging to the same or different autonomous systems" (Mills). Over the years Border Gateway Protocol replaced the former Exterior Gateway Protocol that had originally been proposed for the DARPA (Defense Advanced Research Project Agency) community. The latest version of Border Gateway Protocol was developed in 1995 and is the one currently in use. It is Border Gateway Protocol version 4, usually referred to as BGP-4 or simply BGP.

Many Enterprises, however, do not run BGP. Imagine a house at the end of a dead-end road, you can see that anyone in that house who wants to travel anywhere else, must begin by driving down that one road. That same description can be compared to an Enterprise that has only one connection to the Internet. No matter where in the world they want to send information, it always must start by traveling down that one path to the Internet. That path, because it is always there, can be defined permanently. This is referred to as static routing. BGP is a complex system of routing. If your company has only one connection to the Internet, it is much simpler and, at least in term of man-hours, cheaper to use static and default routes to provide connectivity. When you have multiple routes to the Internet, especially if also using multiple Internet Service Providers, the probability is that you will need to enable BGP in your enterprise.

How BGP works, a simplistic explanation.

As we discussed earlier, most routing protocols route packets between nodes on a network, with the goal of having data reach its destination, preferably by the shortest route possible. This routing can be accomplished via static or predetermined routes, default or last resort routes, or dynamic routes learned via a routing protocol. Border Gateway Protocol is a routing protocol providing dynamic routing. BGP, unlike many of the other routing protocols, does not route primarily within a network but rather routes traffic between networks or Autonomous Systems (AS).

An Autonomous System is defined as “a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS” (Mills), though a more complete definition would add “the administration of an AS appears to other ASs to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through it. From the standpoint of exterior routing, an AS can be viewed as monolithic” (Rekhter, Li). In other words, any AS appears to another AS to be a single system, and any network internal to the AS must be equally reachable from “all border gateways of the AS” (Rekhter, Li). This is as important to the internal network, as it is to the Internet. For example, if your AS were to advertise a route to the Internet before all routers in your own network had learned about the route by way of your internal routing protocol (IGP or Interior Gateway Protocol), your autonomous system could receive traffic that some routers would not yet be able to route. To prevent this from happening, exterior routing must wait until the interior routing “has propagated routing information across your autonomous system” (Cisco). Internet Registries such as ARIN (American Registry for Internet Numbers) assign a unique number to each autonomous system for worldwide identification.

These Autonomous Systems can, as far as Internet routing is concerned, take one of three forms. If an AS has a single connection to the Internet, it is considered to be a stub AS. A stub AS does not need to take part in BGP routing. Traffic bound to the internet has only one path available to it so can be sent directly to the next hop, usually an Internet Provider, by way of a default route. If the AS has more than one connection, it is referred to as multi-homed, and can be either transit or non-transit

(Mobley). An AS is considered multi-homed, even if both connections to the Internet are to a single provider (Valzah). A non-transit AS does not allow traffic that neither originates nor ends inside the AS. Since a non-transit AS does not move traffic through it, it also does not propagate routes learned from another AS (Halabi, McPherson). A transit AS, on the other hand, does propagate routes, and does pass traffic that has both source and destination outside of itself. An ISP (Internet Service Provider) is an excellent example of a multi-homed, transit AS.

BGP has both Internal and External modes. Internal BGP is spoken between two or more routers within a single AS. Although BGP is not, by definition, an internal routing protocol, nonetheless, you may under some conditions want routers within a single AS to communicate with one another via BGP. If, for example, an AS has multiple connections to the Internet, the BGP speaking routers would speak external BGP (eBGP) to the Internet, while speaking internal BGP (iBGP) among themselves. "When a BGP routing update is received from a neighboring AS, it must be relayed directly to all other BGP speakers in the AS" (Baccala). This would insure that the routers presented a uniform appearance of the AS to the Internet.

BGP uses TCP (Transmission Control Protocol) on port 179 as its transport protocol. Since BGP uses an already defined transport, all of the reliability is taken care of by TCP rather than requiring implementation by BGP. BGP routers must perform the standard TCP three-way handshake in order to set-up communication. "The initial data flow is the entire BGP routing table" (Rekhter, Li). Unlike many other routing protocols, BGP does not require updates in specific intervals. Rather, updates are passed between BGP routers only during times of change. During normal operation, only keepalive packets are exchanged between the routers to ensure that the connection is live. If an error condition occurs, a notification is sent, and the connection is closed.

"BGP is really all about choosing the best path for a packet to take. . . it's important to keep in mind that BGP's first goal is finding the best long-term path" (Valzah). However, one must keep in mind that the criterion for determining "the best path" is not only technical policy, but also political policy (Mobley). The politics of the determination is usually connected to cost. As a company connects to the Internet, their choice of Internet Provider(s) is usually determined at least partly by the associated cost in relation to speed of the connection. If a company has two Internet providers, they may prefer their traffic take one of those providers almost exclusively. If, for example, one of the Internet connections is a Gigabit connection, and the other is a back-up connection to simply provide connection at T-1 speeds for emergency, the best choice for traffic would be the Gigabit connection. BGP routes alone will not provide the control necessary to enable a policy of that type.

Simply configuring – or configuring a simple connection.

When configuring BGP, the first consideration is connectivity. Because the routing uses TCP as its transport mechanism, the first criterion of routing with BGP is to have IP connectivity. After physical connectivity and IP connectivity are verified, the next step is to create the TCP connection. This is done by use of the neighbor command

(Mobley). When defining the BGP neighbor(s) to a router, it is important to remember that these neighbors may or may not be physically adjacent to one another. In other words, the BGP neighbor to router A may not be router B or C, but rather router D or even router E. (see illustration #1)

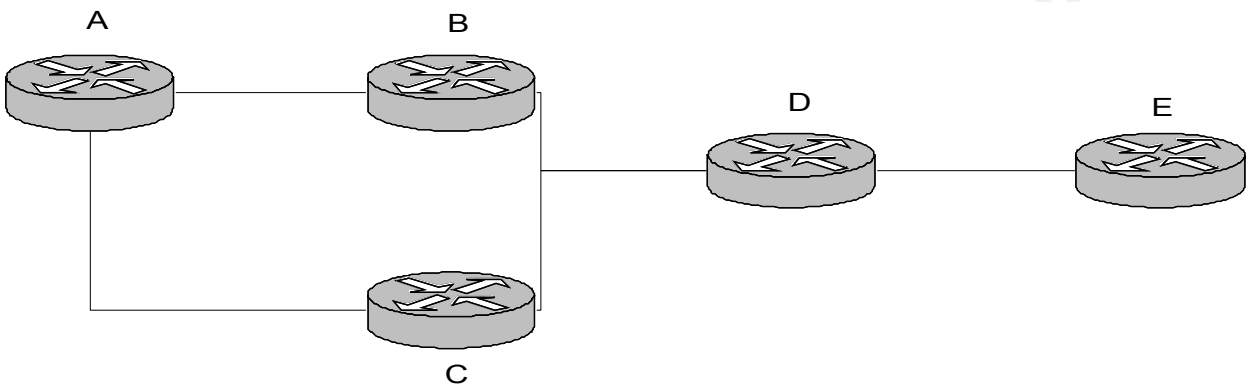

Illustration #1

Although we have discussed the fact that a company may have internal (iBGP) connections and external (eBGP) connections, configuration of both iBGP and eBGP is almost identical. Both are configured by the use of neighbor commands. Let's begin to examine a simple BGP configuration. The configuration we will examine is for a Cisco Router having 2 eBGP neighbors and one iBGP neighbor. Using the above illustration, we will assume that the router we are configuring is Router A. The iBGP neighbor is router B, and the two eBGP neighbors are routers D and E. Router C in this example does not run BGP.

```
router bgp 65104
no synchronization
bgp log-neighbor-changes
network 172.28.0.0 mask 255.255.128.0
network 172.28.128.0 mask 255.255.192.0
network 172.28.192.0 mask 255.255.224.0
network 172.28.224.0 mask 255.255.240.0
network 172.28.240.0 mask 255.255.255.0
network 172.30.0.0
network 172.20.222.0
network 172.21.94.0
network 172.25.40.0 mask 255.255.254.0
network 172.25.42.0
aggregate-address 172.28.0.0 255.255.0.0 summary-only
neighbor 10.250.250.2 remote-as 65104
neighbor 172.20.222.7 remote-as 65454
neighbor 172.25.81.121 remote-as 64560
```

Please understand that the IP addresses and AS numbers referred to in this example, are not intentionally real. We will consider the above configuration in steps.

router BGP 65104 turns on BGP for Autonomous System # 65104 in this router. If this configuration is for an Internet visible router, you MUST use an official AS number which has been assigned to your company. If this autonomous system will not be visible to the Internet, there are private AS numbers available, similar to private IP address space.

no synchronization As we discussed earlier, it is of utmost importance that an autonomous system advertises all of its routes consistently. "If your autonomous system will be passing traffic through it from another autonomous system to a third autonomous system, it is very important that your autonomous system be consistent about the routes that it advertises." Synchronization keeps BGP from advertising a route "until the IGP has propagated routing information across your autonomous system" (Cisco). Some enterprises determine that they do not need to have this feature turned on; therefore no synchronization turns the feature off.

bgp log neighbor changes will log to the defined log file any neighbor changes, which are detected. For the security conscious, this is an important step, as it can help you discover an attack meant to co-opt a peering session.

network xxx.xxx.xxx.xxx mask xxx.xxx.xxx.xxx defines each network that BGP on this router will route. If there are multiple address ranges that will be considered for routing, there must be a network statement that defines each range. If the address range is not a natively masked network (i.e. 11.0.0.0 mask of 255.0.0.0), a mask statement must accompany the network statement to fully define the range to be routed.

aggregate-address 172.28.0.0 255.255.0.0 summary-only is designed to improve your citizenship in the Internet community. As you can see from the network statements for the 172.28.x.x addresses, the network advertised are comparatively small groups of network addresses. The Internet community does not want to be burdened with these small entries in the routing tables, as the tables are already so very large. Therefore, it is a desirable thing to create summary statements if necessary to advertise routes in larger blocks when the specific routes themselves are smaller than 20 bits.

neighbor statements. The next 3 statements define the three BGP neighbors of this router. The first defines Router B, the internal or iBGP relationship. As you can see, this neighbor is defined in a private address space, (10.x.x.x addresses are not assigned to any company, but rather are available for use by anyone, with the understanding that these addresses can not be advertised onto the Internet.) Although this would not be possible for an eBGP relationship, and is not usual even for an iBGP neighbor, it was included this way to show that it could be done. All statements that define a specific peering relationship begin with neighbor xxx.xxx.xxx.xxx identifying the

neighbor in this relationship.

The first statement neighbor 10.250.250.2 remote-as 65104 defines the internal peer, giving the address of the neighbor as well as its autonomous system number. As the autonomous system number of the neighbor is the same as the autonomous system number of Router A's BGP, this is an iBGP peering relationship.

The next two neighbor statements both have different autonomous system numbers than does this router. These two neighbors are both external, or eBGP peers.

If this router is defined as a neighbor in the BGP configuration of each of these 3 neighbor routers, communication should now be verifiable. Because, however, the iBGP neighbor is using private addressing, its address cannot be introduced to the Internet. Internet traffic bound for Router B must, then be sent to a router that both knows about the second router, and knows how to get there. Because Router A fits both of those criteria, we will define it as the next-hop for Router B. To do this we will add a statement, neighbor 10.250.250.2 next-hop-self. This will cause Router A to advertise itself as the correct place to send traffic bound for neighbor 10.250.250.2. Usually, in a fully meshed network, this statement would be superfluous possibly adding unnecessary additional hops.

In addition to the definition of neighbors to create a BGP peering relationship, usually there is also a need to exert some control on the routes that are passed, the routes that are accepted and the traffic that flows into and through this router and autonomous system. Much of this control is exerted by means of route-maps. Route maps "are used on a per-neighbor basis to filter updates and modify various attributes. A route map can be applied to either inbound or outbound updates" (Cisco).

Looking at the example provided above, we can see that Router A's BGP tables has information about a portion of a class B network, 172.28.0.0. Since we are advertising the entire class B out to the Internet via the aggregate-address statement, we can only assume that the IGP must know how to get to the rest of the address space. However, the aggregate-address statement is summarizing the address up to a full class C to all of its neighbors, including the internal (iBGP) neighbor. As this may cause internal routing problems, we now want to advertise specific addresses or unsuppress these small advertisements. To do this we must add and apply a route-map to the neighbor. To apply the route-map, we add another neighbor statement to the iBGP clause.

```
neighbor 10.250.250.2 unsuppress-map AllowSpecifics
```

The route-map is defined in one or more paragraphs. The paragraph includes the name, permit or deny (indicating whether the map is specifically permitting or specifically denying a process), a number of the paragraph, a reference to the items affected, and often a number of commands that will change the characteristics of the affected route. In this specific iBGP relationship, we want to define the route-map, and refer to a list of addresses that we want to not be suppressed for internal route

advertisements.

```
route-map AllowSpecifics permit 10
 match ip address 4

access-list 4 permit 172.28.241.0 0.0.0.255
access-list 4 permit 172.28.242.0 0.0.1.255
access-list 4 permit 172.28.244.0 0.0.3.255
access-list 4 permit 172.28.248.0 0.0.7.255
```

The match command in the route-map paragraph refers to access-list 4, defining the addresses we want specifically advertised to the internal BGP neighbor.

The neighbor definitions for our iBGP peer have now grown to 3, but we may also want to add a definition of this peering. If we do so, the configuration now becomes:

```
router bgp 65104
 no synchronization
 bgp log-neighbor-changes
 network 172.28.0.0 mask 255.255.128.0
 network 172.28.128.0 mask 255.255.192.0
 network 172.28.192.0 mask 255.255.224.0
 network 172.28.224.0 mask 255.255.240.0
 network 172.28.240.0 mask 255.255.255.0
 network 172.30.0.0
 network 172.20.222.0
 network 172.21.94.0
 network 172.25.40.0 mask 255.255.254.0
 network 172.25.42.0
 aggregate-address 172.28.0.0 255.255.0.0 summary-only
 neighbor 10.250.250.2 remote-as 65104
 neighbor 10.250.250.2 description iBGP peering
 neighbor 10.250.250.2 next-hop-self
 neighbor 10.250.250.2 unsuppress-map AllowSpecifics
 neighbor 172.20.222.7 remote-as 65454
 neighbor 172.25.81.121 remote-as 64560

route-map AllowSpecifics permit 10
 match ip address 4

access-list 4 permit 172.28.241.0 0.0.0.255
access-list 4 permit 172.28.242.0 0.0.1.255
access-list 4 permit 172.28.244.0 0.0.3.255
access-list 4 permit 172.28.248.0 0.0.7.255
```

Now we have more fully defined our internal BGP connection, however, the two

external connections are at this point defined only sketchily. With the connection as currently defined, we should communicate, however, we have no control over the connection and what is transmitted over the paths. As we discussed previously, BGP's routing statements are at least partly used to enforce political preferences. Usually, one path will be preferred over another for any of a variety of reasons. Whatever the reason, if a given path is preferred, than there must be some way of enforcing that preference.

The first thing to remember when discussing traffic policy is that inbound traffic and outbound traffic are treated as separate and distinct items. Any policy enforcement applied to outbound traffic will do nothing for inbound, and vice versa. If no policy enforcement is done, than generally traffic will go out whichever way presents the fewest hops to the destination, and traffic will alternatively come in through the pathway that presented to the world the fewest hops to reach your enterprise. If there are no differences in the political drivers between the alternative paths, then this shortest path criteria may be acceptable, usually however, there is a preference for one path under at least some circumstances.

Outbound traffic flow can be determined by a number of factors; however, we will deal with only the most common (which coincidentally are the simplest). Probably the easiest way to ensure that you send all of your traffic to a specific ISP is to have that ISP advertise to you a default route, and then redistribute that default into your IGP. This solution, however, is not amenable to most, as the configuration of the default is then in the power of the provider instead of the Enterprise. If you use Cisco routers, they have provided a proprietary parameter called *weight*. Each peer connection can be given a value called a weight, the higher the weight, the better the path. This weight parameter can be set directly on a neighbor statement, or can be added via a route-map statement, which would provide somewhat more versatility in the configuration. Although the example we are working with does not use a weight statement, if you were to configure one on the neighbor statement, it would be in the form:

neighbor xxx.xxx.xxx.xxx weight 500

The weight statement works well if you have one router that has multiple connections to the Internet. You would then set each connection with a different weight, again, giving the highest weight value to the peer connection you prefer. If, however, your Enterprise has multiple routers connected to the Internet, there must be a method of setting a "weight" that would be shared across the entire Enterprise. This Enterprise-wide "weight" is called local preference. The local preference or the preferred path for the Enterprise to take to the Internet is usually set using one or more route-map statements.

A "community" of routes can also be defined. "Based on the community, you can control which routing information to accept, prefer, or distribute to other neighbors (Cisco)". Three Community variables are considered "well-known" and are used to control route advertisements. These communities explicitly stop either the

advertisement or export of routes to peers (Chandra, Trainer, Li). Communities are used between peers. It's important to realize that the communities are set by the peer which is accepting routes from a peer, so if an ISP supports communities, an Enterprise can set at the ISP the local preference depending on whether or not the Enterprise sends a community variable with the route updates which are being propagated. When this community value is sent to our peer the Enterprise is able to vary the traffic it receives from the peer based on the value sent. Although confusing, the route-map applied in affects outbound traffic, and the route-map applied out affects inbound traffic. So the community sent to our peer as follows with the KC-Out route-map, affects the traffic sent to us by peer 172.20.222.7. The community set by route map Community in the following example is varied by the community value received from peer 172.20.222.7, and affects traffic that we send to that peer.

```
route-map Community permit 10
 match community 103
set local-preference 300
!
route-map Community permit 20
match community 102
set local-preference 200
!
route-map Community permit 30
match community 100
set local-preference 100
```

The format of community variables has changed through the years. Since the original community format was a string of "four octet values . . . ranging from 0x0000000 through 0x0000FFFF and 0xFFFF0000 through 0xFFFFFFFF" (Chandra, Trainer, Li), the communities could be somewhat difficult to read. "In the most recent version of the RFC for BGP, a community is of the form AA:NN, where the first part is the AS number and the second part is a 2 byte number" (Cisco). Cisco's default community format, however, is in the form of NNAA. Therefore, when defining a community on a Cisco router, the additional command *ip bgp-community new-format* is required to change the format of the community variable to the more acceptable form.

```
ip bgp-community new-format
ip community-list 100 permit 65104:100
ip community-list 102 permit 65104:200
ip community-list 103 permit 65104:300
```

Since networks often reflect Enterprise boundaries and change is the nature of business, it is certainly not inconceivable that a change may occur in your or your neighbor's BGP configuration. Unfortunately, a hard change of a BGP peering relationship causes the BGP connection to reset. Resetting BGP connections is fraught with problems for network operations. As a result, something called soft reconfiguration was developed. "Soft reconfiguration allows policies to be configured

and activated without clearing the BGP session" (Cisco). Soft reconfiguration can be either inbound (accepting updates from your peer) or outbound (sending updates to your peer). Because the definition of a "soft" reconfiguration is that the session is not reset, this means the router must "store all of the received updates without modification . . . this can be memory intensive" (Cisco). It may or may not be desirable to accept soft reconfiguration from a peer, depending at least partly on the amount of memory available in your border routers. As inbound soft configurations may not be desirable, they are disallowed by default. If your Enterprise decides to accept soft reconfiguration it must be specifically allowed on a neighbor statement for a peer.

neighbor 172.25.81.121 soft-reconfiguration inbound

Both eBGP peers in our example are allowed to send soft reconfigurations.

We have considered outbound traffic policies, now let's venture a look at controlling inbound traffic. Remembering that maps applied out influence inbound traffic, we will next look at the route-maps that are applied outbound on the configuration for ISP-1. The first of these is a prefix-list applied outbound. The neighbor statement applying the prefix-list *neighbor 172.25.81.121 prefix-list Allowed-BGP out* refers to a named list of IP addresses. This named list will define what prefixes (or address lists) will be advertised by this router.

ip prefix-list Allowed-BGP seq 5 permit 172.30.0.0/16
ip prefix-list Allowed-BGP seq 10 permit 172.26.40.0/23
ip prefix-list Allowed-BGP seq 15 permit 172.24.49.0/24
ip prefix-list Allowed-BGP seq 25 permit 172.26.42.0/24
ip prefix-list Allowed-BGP seq 35 permit 172.28.0.0/16 le 32
ip prefix-list Allowed-BGP seq 40 permit 172.20.222.0/24

As traffic is sent in the direction from which advertisements come, delineating which prefixes we advertise should direct only traffic meant for those addresses to this router.

There is another method of controlling what traffic is sent to an Enterprise called a MED or a Multi-Exit Discriminator. The name is somewhat confusing, as it is configured by an Enterprise to control traffic it receives, however, the name was given from the point of view of the ISP, to control where traffic exits the ISP. A MED is not included in the example shown; however, as it is used by many sites, it is important to discuss it, at least as a simple overview. The MED is exchanged between AS, however; a MED that is received by an AS will not be passed on to another AS. Usually a MED will be used for multiple connections to a single AS, rather than multiple connections to multiple AS, as the point is to control where traffic bound for the Enterprise will exit an ISP. For example if an Enterprise had a connection to ISP-1 in Baltimore, and another connection to ISP-1 in Chicago, they would prefer that traffic destined for the Baltimore site would exit the ISP there, rather than in Chicago. The MED is used to enforce that policy. Many, if not most providers do not allow the use of any manipulation techniques such as a MED to enable an Enterprise to setup a

preference for a best-exit, rather they want to give the traffic to the Enterprise at the closest point to it's entry into their system. This is referred to as *closest-exit* or *hot potato* routing, and is the most common method used by ISPs (Halabi and McPherson).

When connected to the Internet in multiple places, unless your Enterprise is an ISP, you probably don't want to be passing Internet traffic from ISP-1 through your AS to ISP-2, or even to another AS you may be peered with. The example configuration that we have been discussing shows two eBGP connections to different AS. In this particular example, we do want some traffic to be able to transit our AS, but want to define what can transit, and filter out the rest. The neighbor statement applying this filter is

neighbor 172.25.81.121 route-map transit out.

As you can see, the statement is actually applying a route-map out. Again, what is applied out affects inbound traffic.

The route map named transit does two things. First, it matches one of two as-path (either 1 or 2) access-lists each of which has a permit statement affecting outgoing advertisements.

```
ip as-path access-list 1 permit ^$
ip as-path access-list 2 permit 65454$
```

The first is actually a blank (defined by ^$), which will permit our own AS advertisements. The second allows advertisements of routes that end with 65454$. The $ indicates the end of an AS list. As the advertisement is sent our own AS number will be added to the route-list. Second, (and this is always done AFTER the match, our own AS (65104) is pre-pended yet again to the list of AS numbers on the advertisement.

```
route-map transit permit 10
 match as-path 1
 set as-path prepend 65104
!
route-map transit permit 20
 match as-path 2
set as-path prepend 65104
```

Since as we said, our AS number would already be added to the list at the time of advertisement, it may seem problematical to add it again. By pre-pending the advertisement we are sending out, we are in effect adding an extra hop in our advertisement. This makes it appear that we are a little further away through this ISP. Therefore, this ISP should appear slightly less attractive as a route to this Enterprise, and so encourages the use of other routes that are available. This method of controlling routes is easily checked in the Internet by available looking-glass sites. With

the additional statements, our BGP configuration now appears as follows:

```
router bgp 65104
no synchronization
bgp log-neighbor-changes
network 172.28.0.0 mask 255.255.128.0
network 172.28.128.0 mask 255.255.192.0
network 172.28.192.0 mask 255.255.224.0
network 172.28.224.0 mask 255.255.240.0
network 172.28.240.0 mask 255.255.255.0
network 172.30.0.0
network 172.20.222.0
network 172.21.94.0
network 172.25.40.0 mask 255.255.254.0
network 172.25.42.0
aggregate-address 172.28.0.0 255.255.0.0 summary-only
neighbor 10.250.250.2 remote-as 65104
neighbor 10.250.250.2 description iBGP peering
neighbor 10.250.250.2 next-hop-self
neighbor 10.250.250.2 unsuppress-map AllowSpecifics
neighbor 172.20.222.7 remote-as 65454
neighbor 172.20.222.7 description KC peer
neighbor 172.20.222.7 send-community
neighbor 172.20.222.7 route-map Community in
neighbor 172.20.222.7 route-map KC-Out out
neighbor 172.20.222.7 soft-reconfiguration inbound
neighbor 172.25.81.121 remote-as 64560
neighbor 172.25.81.121 description ISP-1
neighbor 172.25.81.121 soft-reconfiguration inbound
neighbor 172.25.81.121 prefix-list Allowed-BGP out
neighbor 172.25.81.121 route-map transit out

ip bgp-community new-format
ip community-list 100 permit 65104:100
ip community-list 102 permit 65104:200
ip community-list 103 permit 65104:300
ip as-path access-list 1 permit ^$
ip as-path access-list 2 permit 65454$

ip prefix-list Allowed-BGP seq 5 permit 172.30.0.0/16
ip prefix-list Allowed-BGP seq 10 permit 172.26.40.0/23
ip prefix-list Allowed-BGP seq 15 permit 172.24.49.0/24
ip prefix-list Allowed-BGP seq 25 permit 172.26.42.0/24
ip prefix-list Allowed-BGP seq 35 permit 172.28.0.0/16 le 32
ip prefix-list Allowed-BGP seq 40 permit 172.20.222.0/24
```

```
ip prefix-list Match-All seq 10 permit 0.0.0.0/0 le 32

access-list 4 permit 172.28.241.0 0.0.0.255
access-list 4 permit 172.28.242.0 0.0.1.255
access-list 4 permit 172.28.244.0 0.0.3.255
access-list 4 permit 172.28.248.0 0.0.7.255

route-map Community permit 10
 match community 103
set local-preference 300
!
route-map Community permit 20
match community 102
set local-preference 200
!
route-map Community permit 30
match community 100
set local-preference 100

route-map AllowSpecifics permit 10
 match ip address 4

route-map KC-Out permit 10
match ip address prefix-list Match-All
set community 65454:300 additive

route-map transit permit 10
 match as-path 1
 set as-path prepend 65104
!
route-map transit permit 20
 match as-path 2
 set as-path prepend 65104
```

But is it secure?

Unfortunately, as you can see in all of the above discussion regarding configuration of BGP, there is not a great deal of security involved in the implementation of BGP. The connection between the routers is a TCP session, and there is not a viable method at this time of ensuring authenticity between the various routers in the Internet. It certainly seems that a spoofed IP address could be inserted into the TCP connection between the routers, to introduce false routes. As the Internet grows exponentially and the world becomes ever more dependent upon it, the lack of security in the routing protocol, the heart of the Internet is becoming of greater concern. .In order to truly improve BGP security one must include verification of IP address ownership, verification of AS ownership, authentication and authorization for a given router to "speak" for a given AS, route and address advertisement authorization, and route

withdrawal authorization (BBN). BBN Technologies has been working on an extension to BGP-4 called S-BGP that addresses these additional security requirements.

Implementation of this new secured extension requires enabling changes in many places. The Internet Registries, the router vendors and the ISP's must all devote time, energy and money in order to facilitate implementation. "The approach adopted to securing BGP route distribution involves two Public Key Infrastructures (PKIs), a new transitive path attribute containing "attestations", and the use of IPsec" (Lynn, Mikkelson, Seo). The PKI certificates would authenticate ownership of IP addresses, AS numbers, BGP routers, and verify the authority of a BGP router to speak for a given AS. The Internet registries would provide PKI assignment, just as AS numbers and IP addresses are now assigned (BBN).

BGP would have a new attribute, the "attestation path attribute" to "carry digital signatures that protect the prefixes . . . the AS path, and optionally other transitive path attributes as a BGP update is propagated between AS" (Lynn, Mikkelson, Seo). IPsec, would then provide the data integrity for the TCP connection itself. "It also protects the connection from replay attacks" (Lynn, Mikkelson, Seo). An Internet Draft describing PKI extensions of identification of autonomous system ids and IP addresses has been written as of February, 2002, and submitted for comment.

But what really is needed to implement the secured BGP? ISP's or some other forum would need to step up to answering the need for certificate repositories. Router vendors need to implement S-BGP in their software. ISP's and other BGP speakers need to purchase and deploy the new technology. It all takes money and time. Hopefully, both the money and time will be spent before some enterprising hacker decides that more fun can be had destroying the Internet than using it.

References

BBN Technologies, Secure BGP Project (S-BGP), http://www.net-tech.bbn.com/sbgp/sbgp-index.html.

Baccala, Brent, Connected, An Internet Encyclopedia, http://www.freesoft.org/CIE/Topics/88.htm

Chandra, R., Trainer, P., Li, S., RFC 1997 "BGP Communities Attribute":, http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1997.html

Cisco IOS Configuration Guide. http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/np1_c/1cprt1/1cbgp.htm#4809

Halabi, Sam, and McPherson, Danny. Internet Routing Architectures (Cisco Press, 2000)

Lynn, Charles; Mikkelson, Joanne; Seo, Karen. Internet Draft, Secure BGP, http://www.net-tech.bbn.com/sbgp/draft-clynn-s-bgp-protocol-00.txt

Mobley, Larry. BGP, The Border Gateway Protocol, Information Management Systems, 2000.

Mills, D. L., RFC 904, "Exterior Gateway Protocol Formal Specification," http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc0904.html,

Rekhter, Y.; Li, T., RFC 1771, "A Border Gateway Protocol 4 (BGP-4)," http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1771.html

Van Valzah, Robert A. Reliable Internet Connectivity with BGP, http://www.bgpbook.com/

Last Updated: July 24th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Community SANS Trenton SEC401	Trenton, NJ	Aug 21, 2017 - Aug 26, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor