

Application Note: Securing BGP on Juniper Routers

Version 1.92, 03/30/2005

Stephen Gill
E-mail: gillsr@cymru.com
Published: 06/16/2002

Contents

Introduction	2
Assumptions	3
Topology	4
Routing Options [edit routing-options].....	5
AS Number	5
Static Routes	5
Black Hole Routes	6
Martians Prefixes.....	7
Load Balancing.....	8
BGP Options [edit protocols bgp].....	9
Tracing	9
Logging.....	10
Route Damping.....	10
Private-AS Filtering.....	12
Prefix Limiting	12
Peer Authentication	13
Policy Options [edit policy-options]	14
Inbound Prefixes.....	14
Outbound Prefixes.....	16
Special Prefixes.....	16
Firewall Filtering [edit firewall].....	17
BGP Connections	17
Conclusion	18
References	18

Introduction

BGP is the glue that holds the Internet together. It is a robust protocol whose primary purpose is to distribute routing information between peering points pertaining to every public internetwork in a seamless fashion. Indubitably, inherent in BGP is a great responsibility, but also a great potential for disaster. This misfortune can be caused by inadvertent misconfiguration or deliberate misuse. Consequently, it is crucial that BGP administrators take extra care in fully securing their peering routers. Not hardening BGP configurations could prove very detrimental to one's local network, or even worse, proving harmful to the rest of the Internet community.

It is quite common to encounter the following BGP related problems on the Internet on a regular basis:

- unaggregated prefixes
- invalid prefixes

- excessive prefixes
- duplicate prefixes
- accidental prefixes
- flapping prefixes
- unnecessary prefixes

This paper describes several preventative measures within Juniper routers that can be taken to decrease the problems mentioned above. No configuration is foolproof, but several suggestions are presented that can reduce the security exposure of Juniper peering routers.

Two Juniper security templates have previously been published by the author designed to guide the administrator through the process of securing his or her Juniper router. This document further bridges the Juniper documentation gap and fleshes the “JUNOS Secure BGP Template” into more detailed verbiage. Both security templates are available for download here:

JUNOS Secure Template [1]

<http://www.cymru.com/gillsr/documents/junos-template.pdf>

JUNOS Secure BGP Template [2]

<http://www.cymru.com/gillsr/documents/junos-bgp-template.pdf>

The Juniper router templates were originally based on two Cisco security templates written by Robert Thomas. Both of them can be downloaded from his website at the following URLs:

Cisco Secure Template [4]

<http://www.cymru.com/Documents/secure-ios-template.html>

Cisco Secure BGP Template [5]

<http://www.cymru.com/Documents/secure-bgp-template.html>

Assumptions

A few assumptions worth noting are made by this document. Firstly, we assume the reader has a basic understanding of the BGP protocol and a working knowledge of Juniper Routers. The JUNOS documentation set can be found at <http://www.juniper.net>. Please consult the Juniper documentation for further information on configuring Juniper Routers and BGP.

Secondly we assume that the configurations presented in this document are representative of the test topology below and that individual network

policies may warrant modifications to the recommendations described herein.

Finally, we also assume that the routers in question are running JUNOS version 4.3 or above.

Topology

Before delving into the details of our recommended BGP security measures, we present a topology diagram on which the configuration details are based.

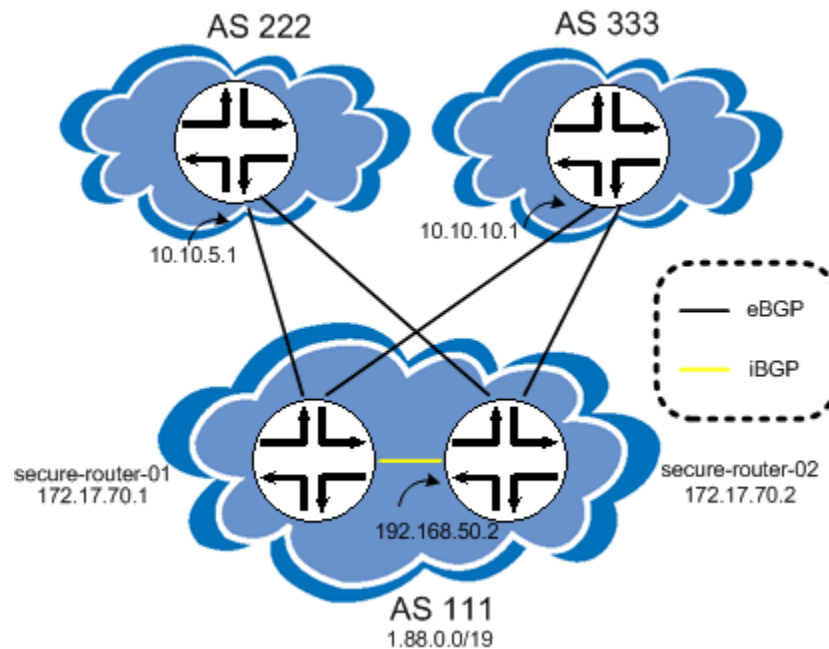


Figure 1 - Juniper BGP Topology

Figure 1 pictures three autonomous systems and four distinct Juniper routers. Two routers in AS 111 each contain one EBGP link to AS 222 and to AS 333 over which they announce the aggregate prefix, 1.88.0.0/29. Furthermore, they maintain an IBGP session between each other. Relevant IP addresses have been included in the diagram for clarity, and are described below:

- 10.10.5.1, AS 222 external IP
- 10.10.10.1, AS 333 external IP
- 172.17.70.1, AS 111 secure-router-01 loopback.
- 172.17.70.2, AS 111 secure-router-02 loopback.
- 192.168.50.2, AS 111 secure-router-02 internal IP.
- 192.168.50.5, internal IP next-hop not shown in topology.
- 192.168.50.8, same as the above.
- 192.168.50.10, same as the above.

The configurations presented in this document were written from the perspective of secure-router-01. They are also listed in function style formatting rather than set style formatting for better readability. Issuing the commands on a router would require appending the word “set” to the beginning of each statement, or using the CLI command “load merge term” for the function style configurations. We leave porting these configurations from the included examples to the neighboring routers as an exercise for the reader.

This document is divided into four major sections, including:

- Routing Options
- BGP Options
- Policy Options
- Firewall Filtering

All sections represent a major level in the JUNOS configuration tree; each is interspersed with basic commands to assist in network verification and troubleshooting.

Routing Options [edit routing-options]

Basic routing settings such as static routes, aggregate routes, martian routes, load balancing options, and the local AS number are configured at the [edit routing-options] level.

AS Number

BGP needs a local AS number to advertise to its peers. The AS number of the local router is configured as follows:

```
[edit routing-options]
/* Our AS Number */
autonomous-system 111;
```

Static Routes

Next, static routes can be used to create aggregates and more specific routes. The aggregate protocol can achieve the same results as a static discard route, but here we chose to stick with a static to announce our prefix of 1.88.0.0/19. To override the discard route, more specific routes are needed in the local routing table.

Loss of connectivity to any of these contributing routes will not carry into external advertisements. This has the added benefit of increasing stability of the global routing table by not changing reachability information to the aggregate prefix. JUNOS will always prefer more specific routes, therefore when packets arrive at the local router they will be forwarded to the appropriate location, assuming it is reachable, rather than being black-holed.

```
[edit routing-options]
static {
  /* This is our aggregate static route */
  route 1.88.0.0/19 discard;
  /* More specific routes used with discard route above.
     Remove if using an IGP to discover internal routes. */
  route 1.88.50.0/24 next-hop 192.168.50.5;
  route 1.88.55.0/24 next-hop 192.168.50.8;
  route 1.88.75.128/25 next-hop 192.168.50.10;
  /* Route to loopback of our iBGP peer */
  route 172.17.70.2/32 next-hop 192.168.50.2;
}
```

Three static routes are used to direct the local traffic towards its final destination. Notice that packets that fall within the scope of the aggregate prefix (/19) but not within the specifics (/24, /24, /25) will be discarded until further specific routes are added to the local routing table.

The last static route listed is used to point to the loopback address of our IBGP peer. This is necessary because we typically peer with loopback addresses for increased IBGP resilience. In our sample topology there is only one path to our IBGP peer, but in the real world there might be several paths to an internal peer. An IGP such as OSPF or IS-IS would usually be used to learn the best path to an internal peer. Peering with a loopback address allows the IGP path to change without affecting the BGP peering session assuming another route exists to the destination.

DNS resolution is not usually necessary on a router. As such, it can be disabled thusly:

```
[edit routing-options]
options {
  /* Turn off DNS resolution */
  no-resolve;
}
```

Black Hole Routes

Similar to our discard aggregate, below we list a series of routes that should never be seen on the Internet. Whether packets are traveling outbound or inbound, traffic destined to these networks will be dropped if they reach the local router. All of the routes are divided into /8 prefixes for administrative purposes since IANA allocations may change over time. For an aggregated list, please consult the bogon list at: <http://www.cymru.com/Documents/bogon-list.html>.

If your network does not contain a 0/0 default route and contains the entire Internet routing table, the static discard routes below are not necessary. Instead, the martian addresses should be replaced with the “orlonger” keyword which will disallow these networks from entering the routing table.

Static discard routes are used to remove all ambiguity when a 0/0 route exists.

```
[edit routing-options static]
/* Discard routes for traffic destined to these networks */
/* Use: http://www.cymru.com/gillsr/documents/junos-discard-routes.txt
/*
```

Though we use RFC 1918 addressing in our topology, we still list those prefix ranges in this section and under the martian addresses. It is expected that the IP addresses listed in this document should be replaced with actual public IP addresses specific to your network. Networks that do in fact use RFC 1918 addressing should not include those ranges under the black hole routes above or the martian prefixes to follow.

Martians Prefixes

Martians differ slightly from discard routes because they have the added functionality of restricting certain announcements from entering the routing table. The standard set of Juniper Martian addresses include:

```
0/8
127/8
128.0/16
191.255/16
192.0.0/24
223.255.255/24
240/4
```

A more extensive and complete list of martian prefixes is used in our configuration and is listed below. Of special note is the use of the “longer” instead of the “orlonger” keyword. This allows us to create a list of martian addresses while permitting the specific discard routes we created above to remain in the routing table. Again, prefixes are listed on /8 boundaries for administrative purposes since IANA allocations may change over time. For an aggregated list, please consult the bogon list at: <http://www.cymru.com/Documents/bogon-list.html>.

```
[edit routing-options martians]
/* Use: http://www.cymru.com/gillsr/documents/junos-martians.txt
*/
```

Martians can be viewed on the router by issuing the “*show route martians*” command. If the administrator is seeking a more static list of martians, he or she might wish to restrict the list of martians to the following groups:

```
0/8 ; Juniper Martians
127/8
128.0/16
191.255/16
192.0.0/24
```

```
223.255.255/24
240/4
10/8           ; RFC 1918 Addresses
172.16/12
192.168/16
169.254/16    ; Bill Manning Draft [7]
192.0.2.0/24
198.18.0.0/15
224.0.0.0/4   ; Multicast
```

The prefix 192.175.48.0/24 listed in Bill Manning's draft [7] used for RFC-1918 nameservers has not been included because they are used to reduce the load of the root nameservers for those who do not comply with RFC 1918.

Once again, Multicast and RFC1918 address ranges have been include in the list above and may need to be removed if your network configuration requires these.

Load Balancing

When multiple equal cost paths exist to a destination, JUNOS has the capability of performing load balancing to split the load across multiple paths. Routers that have an IP2 ASIC can perform per flow load balancing on up to 16 equal cost paths. Routers that have an IP1 ASIC can perform per packet load balancing also on up to 8 equal cost paths. On older M40 routers that contain the IP1 ASIC load balancing is not generally recommended for interactive traffic.

Per-packet load balancing is not enabled by default; however per-prefix load balancing is. For a given destination prefix, a random next-hop address will be selected and placed in the forwarding table. The greater the number of prefixes, the better the load distribution will be across different next-hops. To install multiple next-hops into the FIB you should enable flow-based per-packet load balancing. This is done by creating a simple load balancing policy and applying it to the forwarding table. The forwarding table should point to a policy that enables load balancing with an export statement like so:

```
[edit routing-options]
/* Export the policy to turn on flow based load balancing */
forwarding-table {
    export load-balancing;
}
```

The load balancing policy is created under the [edit policy-options] hierarchy as follows:

```
[edit policy-options]
/* Policy to configure load balancing. */
policy-statement load-balancing {
```

```

    then {
        load-balance per-packet;
    }
}

```

The *'per-packet'* directive is somewhat of a misnomer because it actually performs flow based load balancing on the IP2 ASIC.

BGP Options [edit protocols bgp]

Most options specific to the BGP protocol itself are configured in the [edit protocols bgp] branch. These options include configurations for protocol tracing, logging, route damping, private-as filtering, prefix limiting, and peer authentication.

Tracing

To assist in troubleshooting BGP it is helpful to keep track of all BGP state transitions and basic events in a log file on the router. The following configuration allows up to 5 rotational log files to be created at 1MB each that keep track of BGP state transitions and normal events.

```

[edit protocols bgp]
traceoptions {
    /* Rotate through 5 files at 1mb each */
    file log-bgp size 1m files 5;
    /* Trace BGP state transitions */
    flag state;
    /* Trace BGP normal events */
    flag normal;
}

```

To view the active log file, simply issue the command *'show log log-bgp'*. Several tracing flags exist that might further assist in discovering helpful BGP information. Following is a list of all the flags available for BGP debugging:

all	Trace everything
aspath	
damping	
general	Trace general events
keepalive	
normal	Trace normal events
open	Trace BGP open packets
packets	Trace all BGP protocol packets
policy	Trace policy processing
route	Trace routing information
state	Trace state transitions
task	Trace routing protocol task processing
timer	Trace routing protocol timer processing
update	Trace BGP update packets

Logging

A knob for logging BGP neighbor changes also exists apart from the traceoptions. To log BGP neighbor changes in the router log file use the following syntax:

```
[edit protocols bgp]
log-updown;
```

Route Damping

Route damping is a mechanism for BGP enabled routers aimed at improving the overall stability of the Internet routing table and offloading routers' CPUs. Unstable routes may have a profound effect on the interdomain routing table; in many cases if the oscillation of a flapping route is small enough, it is considered good practice to withdraw the advertisement until it has stabilized. A well known publication by the RIPE organization known as RIPE-229 [6], provides excellent guidelines and watermarks on which to base these parameters.

The premise of the parameters defined in the RIPE publication is simple: the degree of restrictions placed on prefixes should increase according to length. The only exceptions are the netblocks that pertain to the root name servers. Since DNS resolution is at the heart of how the Internet functions, and since humans are not in the regular practice of memorizing IP addresses, these netblocks should always be announced whether they oscillate or not.

RIPE-229 supercedes RIPE-210, a previous publication that neglected to stay up-to-date with the changes in DNS netblock allocations. Before the "golden networks" in RIPE-229 were introduced, a publication entitled "RIPE-210 Addendum" [3] was released by the author aimed at educating the administrator on how to stay current with these special allocations.

BGP route damping can be enabled globally on a Juniper Router like so:

```
[edit protocols bgp]
damping;
```

If the administrator were to leave BGP damping enabled with the default settings, he or she would be neglecting to perform two important functions, namely graded flap damping, and golden network exclusion. A more robust configuration would take the RIPE-229 publication seriously and would employ a policy to bypass key prefixes while performing measured damping based on prefix length.

JUNOS penalizes on route withdrawal and on readvertisement. One route flap attracts a total penalty of 2000 (1000 + 1000) while an attribute

change attracts a penalty of 500. The penalty is known as the figure of merit. The “damping” policy below should be used whenever possible:

```
[edit policy-options]
policy-statement damping {
/* Do NOT dampen DNS root-servers */
  term 1 {
    from {
      prefix-list root-servers.net;
    }
    then {
      damping damp-none;
      /* jump to next policy called */
      next policy;
    }
  }
/* Dampen according to prefix length. */
  term 2 {
    from {
      /* Smallest penalty for /21 and smaller */
      route-filter 0.0.0.0/0 upto /21 damping damp-short;
      /* Medium penalty for /22 to /23 */
      route-filter 0.0.0.0/0 upto /23 damping damp-medium;
      /* Highest penalty for /24 and larger */
      route-filter 0.0.0.0/0 orlonger damping damp-long;
    }
    then {
      next policy;
    }
  }
}
```

Prefix lists such as “root-servers.net” are configured under the [edit policy-options] hierarchy and are discussed later in this paper. In the “damping” policy above, root server netblocks are not damped. Prefixes up to size /21 receive the shortest damping times in accordance with the “damp-short” specifications below. Prefixes in the range of /22 to /23 receive a medium weighted policy while prefixes of size /24 and longer receive the strictest damping parameters. The individual damping configuration settings are detailed as follows:

```
[edit policy-options]
/* Min: 30 min, Max: 60 min, dampen at 3 flaps */
damping damp-long {
  half-life 30;
  reuse 1640;
  suppress 6000;
  max-suppress 60;
}
/* Min: 15 min, Max: 45 min, dampen at 3 flaps */
damping damp-medium {
  half-life 15;
  reuse 1500;
  suppress 6000;
```

```

        max-suppress 45;
    }
    /* Min: 10 min, Max: 30 min, dampen at 3 flaps */
    damping damp-short {
        half-life 10;
        reuse 3000;
        suppress 6000;
        max-suppress 30;
    }
    /* Do not dampen. Referenced for DNS root-servers */
    damping damp-none {
        disable;
    }
}

```

The half-life is an exponential decay algorithm that applies to the figure of merit value. The reuse threshold specifies at what value the route will be used again while the suppress value specifies at what point a route should start to be suppressed. Finally, the max-suppress statement delineates the maximum amount of time that a route can be suppressed.

Once the policy is fully configured, it should be applied to external BGP peers as an import policy. The import policy chain to enable BGP damping for EBGP peers is listed in the [edit policy-options] section later in this paper.

To view route damping statistics, one can use the “*show route damping [decayed | suppressed | history]*” command. A quick summary of the number of routes that are damped per peer can be seen in the “*show bgp summary*” command output.

Private-AS Filtering

JUNOS does not remove private AS numbers from AS Path advertisements by default. Private AS numbers subsist in the range 64512-65535 and are reserved for local use only. Private AS numbers should not be seen on the public Internet. To keep private AS numbers from leaking out, use the following syntax:

```

[edit protocols bgp]
/* Keep private AS numbers 64512-65535 from leaking out */
remove-private;

```

Prefix Limiting

The size of the Internet routing table varies daily and is constantly increasing. This rate of change and slow growth can be used to the administrator’s advantage by hard coding the number of prefixes that are expected from a BGP peer. If the number of routes received exceeds a known threshold, it is quite likely that there has been a misconfiguration on the other end allowing more than the standard prefixes to be advertised.

Prefix limiting is the ability to place an upper bound on the number of prefixes received from a neighbor in order to guard against accidents caused by peers. Peers that exceed this threshold are logged and shut down. The two primary benefits of prefix limiting include memory conservation and protection from invalid route propagation.

If this feature is to be used, the administrator must take into account the current size of the routing table and keep a close watch on these watermarks. One way of staying up-to-date is to receive the bgp-stats report by subscribing to majordomo@lists.apnic.net with the statement "subscribe bgp-stats" in the message body.

Prefix limits can be applied at the neighbor level or group level. Here we configure them globally like so:

```
[edit protocols bgp]
family inet {
  any {
    prefix-limit {
      /* Tear down connection when routes reach maximum. */
      maximum 130000;
      /* Issue warning messages at teardown percent */
      teardown 90;
    }
  }
}
```

Notice that we have asked the router to begin issuing warning messages when the number of prefixes received is at 90% of the threshold. When the number of prefixes reaches 130000 the connection will be torn down, and will stay down awaiting manual intervention.

Peer Authentication

BGP peer authentication adds an additional layer of protection against external forgery by requiring a deeper trust relationship among peers through the HMAC-MD5 algorithm. It is strongly recommended that routing protocols employ the use of authentication whenever possible.

HMAC-MD5 computes a hash from a local secret key and the data being transmitted. The same hash is computed on both sides with the transmitted data and compared against what is received for message validation. Messages received that do not match the local hash are discarded.

The example below shows the configuration of a group-level authentication key for all peers in AS 111 – in this case there is only one, 172.17.70.2. BGP authentication can also be configured globally or at the individual neighbor levels. The relevant BGP configuration statement has been printed in bold below:

```
[edit protocols bgp]
/* iBGP peer-group with AS 111. Peer-groups save typing and CPU
cycles when multiple neighbors exist with same policy */
group iBGP_111 {
  /* No longer needed in newer versions if JUNOS. */
  type internal;
  description "iBGP with AS 111";
  /* Set my address to that of lo0 */
  local-address 172.17.70.1;
  authentication-key bgpwith111;
  /* Set next-hop-self for eBGP routes sent to our iBGP peer */
  export next-hop-self;
  /* The following is assumed if not entered */
  peer-as 111;
  /* Loopback address of our internal peer */
  neighbor 172.17.70.2;
}
```

Policy Options [edit policy-options]

Policy statements are created within the [edit policy-options] hierarchy. They are used to control routing behavior such as filtering and setting specific attributes on routes.

Inbound Prefixes

To control what we will do with all routes received from our BGP peers we create several policies, each with their own specific purpose. First we need a policy that will reject all bogon routes. Since we have already created a list of martians, this list only consists of the multicast range 224/4. Notice that we did not include 224.0.0.0/24 in the martian list because some IGPs such as OSPF use multicast to communicate routing information.

```
[edit policy-options]
policy-statement nobogons {
  from route-filter 224.0.0.0/4 orlonger reject;
}
```

We also create a filter to reject all advertisements that contain a private AS number anywhere in the AS Path. Do not use this filter for peering sessions that require the use of private AS numbers.

```
[edit policy-options]
policy-statement noprivatesasns {
  from as-path private;
  then reject;
}
as-path private 64512-65535;
```

Next, we create a policy that drops all prefixes larger than a /27. Individual BGP policies may vary on the size of prefixes accepted into BGP. Here we have chosen to be fairly lax:

```

policy-statement nosmallprefixes {
    from route-filter 0.0.0.0/0 prefix-length-range /27-/32
    reject;
}

```

Finally, we create a policy used for our IBGP peers that sets the next-hop address in the BGP advertisement to the IP address of the local router instead of carrying the EBGP next-hop information into IBGP. Be very careful when using route-reflectors not to blindly apply this policy to IBGP route-reflector peer-groups. Otherwise, unintended routing loops may occur when client routes are reflected to others.

```

/* Set next-hop to self. */
policy-statement next-hop-self {
    then {
        next-hop self;
    }
}

```

Each of the policies can in turn be applied consecutively to our BGP peers in a chain-like fashion. Our policies were designed with this thought in mind by not issuing a “accept” or “reject” statements for the catchall rules. For our EBGP peers we use a chain of four policy statements to filter out all bogons, to remove private ASN and all small prefix advertisements, and to dampen routes. The relevant BGP configuration statement has been highlighted below:

```

[edit protocols bgp]
/* eBGP peer-group with AS 222 */
group eBGP_222 {
    /* No longer needed in newer versions if JUNOS. */
    type external;
    description "eBGP with AS 222";
    authentication-key bgpwith222;
    /* Remove bogons, set damping, and remove small prefixes */
    import [ nobogons nosmallprefixes noprivateasns damping ];
    /* Only announce our netblock */
    export announce;
    peer-as 222;
    multipath;
    neighbor 10.10.10.1;
}

```

To view the prefixes that have been received from a neighbor, or ADJ-RIB-IN, use the “*show route receive-protocol bgp <neighbor>*” command.

One configuration option which the reader might not yet be familiar with is the “multipath” statement used above. Multipath enables the selection of multiple non-multi-hop EBGP or IBGP paths as active paths.

To view the BGP prefixes that have made it into the routing table, or LOC-RIB, use the “*show route protocol bgp*” command.

Outbound Prefixes

To control what we will do with all routes sent to our EBGP peers we create one policy that will only accept our aggregate prefix. Since the topology in this paper is not a transit AS, we do not want to announce reachability to other Internet destinations through our network. Announcing only our prefix to our EBGP peers allows us to be the sole users of our network pipes.

```
[edit policy-options]
/* Match what we configured as our static aggregate netblock */
policy-statement announce {
  term 1 {
    from {
      protocol static;
      route-filter 1.88.0.0/19 exact;
    }
    then accept;
  }
  term 2 {
    then reject;
  }
}
```

The “announce” policy-statement must be applied to peers in AS 222 and AS 333. Since we have previously shown the configuration for peers in AS 222, we include the peer-group for AS 333 below with the export policy highlighted in bold:

```
/* eBGP peer-group with AS 333 */
group eBGP_333 {
  type external;
  description "eBGP with AS 333";
  authentication-key bgpwith333;
  import [ nobogons nosmallprefixes noprivat easns damping ];
  export announce;
  peer-as 333;
  multipath;
  neighbor 10.10.5.1;
}
```

To view the prefixes that will be sent to a neighbor, or ADJ-RIB-OUT, issue the “*show route advertising-protocol bgp <neighbor>*” command.

Special Prefixes

Previously in our BGP damping configuration we referenced a prefix-list named “root-servers.net”. This prefix list defines a list of root servers as of

09/11/01 and should be kept up to date by referencing the RIPE-229 publication. The configuration for the prefix-list is as follows:

```
[edit policy-options]
prefix-list root-servers.net {
  128.8.0.0/16;
  128.9.0.0/16;
  128.63.0.0/16;
  192.5.4.0/23;
  192.33.4.0/24;
  192.36.148.0/24;
  192.112.36.0/24;
  192.203.230.0/24;
  193.0.14.0/24;
  198.32.64.0/24;
  198.41.0.0/24;
  202.12.27.0/24;
}
```

Firewall Filtering [edit firewall]

Firewall filters are created within the [edit firewall] hierarchy. Filters are equivalent to Cisco ACLs and are used to restrict packets that are allowed to travel to and through the router.

BGP Connections

Juniper Routers will not establish peering relationships with unknown BGP peers. However, it is possible to probe TCP port 179 of a Juniper Router to which it will respond with a TCP RST for unknown peers. To keep the router from sending TCP RST packets in response to BGP probes, a firewall filter that only acknowledges known BGP peers should be crafted and applied to the loopback interface of the router.

The following filter named “router-protect” allows BGP connections from all known BGP peers, logs and discards all violations, and counts them in the “manage-discard-bgp” counter. All other traffic is allowed by default.

```
[edit firewall]
filter router-protect {
  /* Drop and log all unexpected BGP connection attempts */
  term 1 {
    from {
      address {
        0.0.0.0/0;
        10.10.5.1/32 except;
        10.10.10.1/32 except;
        172.17.70.1/32 except;
        172.17.70.2/32 except;
      }
      protocol tcp;
      port bgp;
    }
    then {
```

```

        count manage-discard-bgp;
        discard;
    }
}
term 2 {
    then {
        /* Allow all other traffic */
        count manage-accept-other;
        accept;
    }
}
}

```

The firewall filter should be merged with any existing loopback filters and applied to the loopback interface of the router. The appropriate syntax has been highlighted in bold below:

```

[edit interfaces lo0]
unit 0 {
    family inet {
        filter {
            input router-protect;
        }
        address 172.17.70.1/32;
    }
}

```

Counters can be viewed with the “*show firewall*” command, and firewall logs can be viewed with the “*show firewall log*” command.

Conclusion

Given the tremendous importance of BGP and its use on the Internet as the sole distributor of reachability information, it behooves the peering administrator to take the necessary precautions to protect it as much as possible. This paper described several preventative measures to reduce the security exposure of Juniper Routers to BGP accidents, route oscillations, and deliberate abuse.

References

- [1] Gill, Stephen, "JUNOS Secure Template", November 2001.
<http://www.cymru.com/gillsr/documents/junos-template.pdf>
- [2] Gill, Stephen, "JUNOS Secure BGP Template", June 2002.
<http://www.cymru.com/gillsr/documents/junos-bgp-template.pdf>
- [3] Gill, Stephen, "RIPE 210 Addendum", September 2001.
<http://www.cymru.com/gillsr/documents/ripe-210-addendum.pdf>
- [4] Thomas, Rob, "Secure IOS Template", June 2001.

<http://www.cymru.com/Documents/secure-ios-template.html>

[5] Thomas, Rob, "Secure BGP Template", June 2001.
<http://www.cymru.com/Documents/secure-bgp-template.html>

[6] RIPE, "RIPE Routing-WG Recommendations for Coordinated Route-flap Damping Parameters", October 2001.
<http://www.ripe.net/ripe/docs/ripe-229.html>

[7] Manning, Bill, "Special use IPV4 prefixes", May 2002.
<http://www.ietf.org/internet-drafts/draft-manning-dsua-08.txt>

[8] Thomas, Rob, "Bogon List", July 2002.
<http://www.cymru.com/Documents/bogon-list.html>