

Chapter 3

VPN Examples

The following examples illustrate how to configure Border Gateway Protocol (BGP) route target filtering for virtual private networks (VPNs):

- BGP Route Target Filtering for VPNs Overview on page 39
- BGP Route Target Filtering for VPNs on page 42
- Route Origin for VPNs on page 52

BGP Route Target Filtering for VPNs Overview

BGP route target filtering is enabled by configuring the `family route-target` statement at the appropriate BGP hierarchy level. This statement enables the exchange of a new `route-target` address family, which is stored in the `bgp.rtarget.0` routing table.

The following configuration illustrates how you could configure BGP route target filtering for a BGP group titled `to_vpn04`:

```
[edit]
protocols {
  bgp {
    group to_vpn04 {
      type internal;
      local-address 10.255.14.182;
      peer-as 200;
      neighbor 10.255.14.174 {
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
    }
  }
}
```

The following configuration illustrates how you could configure a couple of local VPN routing and forwarding (VRF) routing instances to take advantage of the functionality provided by BGP route target filtering. Based on this configuration, BGP would automatically generate local routes corresponding to the route targets referenced in the VRF import policies (note the targets defined by the `vrf-target` statements).

```
[edit]
routing-instances {
  vpn1 {
    instance-type vrf;
    interface t1-0/1/2.0;
    vrf-target target:200:101;
    protocols {
      ospf {
        export bgp-routes;
        area 0.0.0.0 {
          interface t1-0/1/2.0;
        }
      }
    }
  }
  vpn2 {
    instance-type vrf;
    interface t1-0/1/2.1;
    vrf-target target:200:102;
    protocols {
      ospf {
        export bgp-routes;
        area 0.0.0.0 {
          interface t1-0/1/2.1;
        }
      }
    }
  }
}
```

Issue the `show route table bgp.rtarget.0` show command to verify the BGP route target filtering configuration:

```
user@host> show route table bgp.rtarget.0

bgp.rtarget.0: 4 destinations, 6 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200:200:101/96
    * [RTarget/5] 00:10:00
      Local
200:200:102/96
    * [RTarget/5] 00:10:00
      Local
200:200:103/96
    * [BGP/170] 00:09:48, localpref 100, from 10.255.14.174
      AS path: I
      > t3-0/0/0.0
200:200:104/96
    * [BGP/170] 00:09:48, localpref 100, from 10.255.14.174
      AS path: I
      > t3-0/0/0.0
```

The `show` command display format for route target prefixes is:

AS number:route target extended community/length

The first number represents the autonomous system of the router that sent this advertisement. The remainder of the display follows the JUNOS `show` command convention for extended communities.

The output from the `show route table bgp-rtarget.0` command displays the locally generated and remotely generated routes.

The first two entries correspond to the route targets configured for the two local VRF routing instances (`vpn1` and `vpn2`):

- 200:200:101/96—Community 200:101 in the `vpn1` routing instance
- 200:200:102/96—Community 200:102 in the `vpn2` routing instance

The last two entries are prefixes received from a BGP peer:

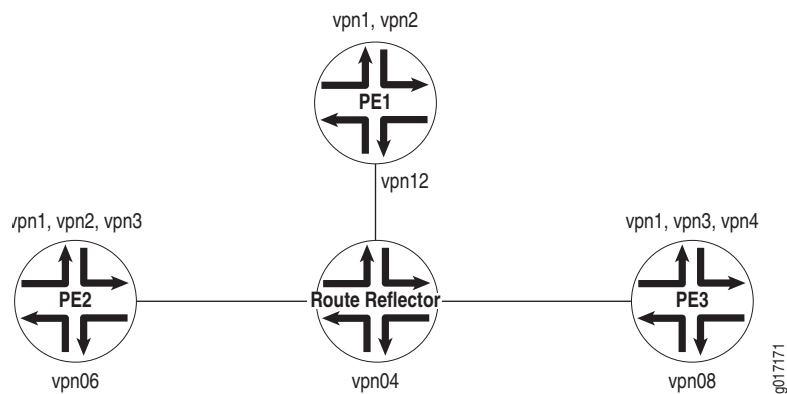
- 200:200:103/96—Tells the local router that routes tagged with this community (200:103) should be advertised to peer 10.255.14.174 through t3-0/0/0.0
- 200:200:104/96—Tells the local router that routes tagged with this community (200:104) should be advertised to peer 10.255.14.174 through t3-0/0/0.0

BGP Route Target Filtering for VPNs

The following sections describe how to configure BGP route target filtering for a group of VPNs based on the topology illustrated in Figure 3:

- Configure BGP Route Target Filtering on Router PE1 on page 42
- Configure BGP Route Target Filtering on Router PE2 on page 44
- Configure BGP Route Target Filtering on the Route Reflector on page 47
- Configure BGP Route Target Filtering on Router PE3 on page 49

Figure 3: BGP Route Target Filtering Enabled for a Group of VPNs



Configure BGP Route Target Filtering on Router PE1

This section describes how to enable BGP route target filtering on router PE1 for this example.

Configure the routing options on router PE1 as follows:

```
[edit]
routing-options {
  route-distinguisher-id 10.255.14.182;
  autonomous-system 200;
}
```

Configure the BGP protocol on router PE1 as follows:

```
[edit]
protocols {
  bgp {
    group to_VPN_D {
      type internal;
      local-address 10.255.14.182;
      peer-as 200;
      neighbor 10.255.14.174 {
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
    }
  }
}
```

Configure the vpn1 routing instance as follows:

```
[edit]
routing-instances {
  vpn1 {
    instance-type vrf;
    interface t1-0/1/2.0;
    vrf-target target:200:101;
    protocols {
      ospf {
        export bgp-routes;
        area 0.0.0.0 {
          interface t1-0/1/2.0;
        }
      }
    }
  }
}
```

Configure the vpn2 routing instance on router PE1 as follows:

```
[edit]
routing-instances {
  vpn2 {
    instance-type vrf;
    interface t1-0/1/2.1;
    vrf-target target:200:102;
    protocols {
      ospf {
        export bgp-routes;
        area 0.0.0.0 {
          interface t1-0/1/2.1;
        }
      }
    }
  }
}
```

Once you have implemented this configuration, you should see the following when you issue a `show route table bgp.rtarget.0` command:

```
user@host> show route table bgp.rtarget.0

bgp.rtarget.0: 4 destinations, 6 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200:200:101/96
                *[RTarget/5] 00:27:42
                  Local
                  [BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                AS path: I
                > via t3-0/0/0.0
200:200:102/96
                *[RTarget/5] 00:27:42
                  Local
                  [BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                AS path: I
                > via t3-0/0/0.0
200:200:103/96
                *[BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                AS path: I
                > via t3-0/0/0.0
200:200:104/96
                *[BGP/170] 00:27:30, localpref 100, from
10.255.14.174
                AS path: I
                > via t3-0/0/0.0
```

Configure BGP Route Target Filtering on Router PE2

This section describes how to enable BGP route target filtering on router PE2 for this example.

Configure the routing options on router PE2 as follows:

```
[edit]
routing-options {
    route-distinguisher-id 10.255.14.176;
    autonomous-system 200;
}
```

Configure the BGP protocol on router PE2 as follows:

```
[edit]
protocols {
  bgp {
    group to_vpn04 {
      type internal;
      local-address 10.255.14.176;
      peer-as 200;
      neighbor 10.255.14.174 {
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
    }
  }
}
```

Configure the vpn1 routing instance on router PE2 as follows:

```
[edit]
routing-instances {
  vpn1 {
    instance-type vrf;
    interface t3-0/0/0.0;
    vrf-target target:200:101;
    protocols {
      bgp {
        group vpn1 {
          type external;
          peer-as 101;
          as-override;
          neighbor 10.49.11.2;
        }
      }
    }
  }
}
```

Configure the `vpn2` routing instance on router PE2 as follows:

```
[edit]
routing-instances {
  vpn2 {
    instance-type vrf;
    interface t3-0/0/0.1;
    vrf-target target:200:102;
    protocols {
      bgp {
        group vpn2 {
          type external;
          peer-as 102;
          as-override;
          neighbor 10.49.21.2;
        }
      }
    }
  }
}
```

Configure the `vpn3` routing instance on router PE2 as follows:

```
[edit]
routing-instances {
  vpn3 {
    instance-type vrf;
    interface t3-0/0/0.2;
    vrf-import vpn3-import;
    vrf-export vpn3-export;
    protocols {
      bgp {
        group vpn3 {
          type external;
          peer-as 103;
          as-override;
          neighbor 10.49.31.2;
        }
      }
    }
  }
}
```

Once you have configured router PE2 in this manner, you should see the following when you issue the `show route table bgp.rtarget.0` command:

```
user@host> show route table bgp.rtarget.0

bgp.rtarget.0: 4 destinations, 7 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200:200:101/96
                *[RTarget/5] 00:28:15
                  Local
                  [BGP/170] 00:28:03, localpref 100, from
10.255.14.174
                  AS path: I
                  > via t1-0/1/0.0
200:200:102/96
                *[RTarget/5] 00:28:15
                  Local
                  [BGP/170] 00:28:03, localpref 100, from
10.255.14.174
                  AS path: I
                  > via t1-0/1/0.0
200:200:103/96
                *[RTarget/5] 00:28:15
                  Local
                  [BGP/170] 00:28:03, localpref 100, from
10.255.14.174
                  AS path: I
                  > via t1-0/1/0.0
200:200:104/96
                *[BGP/170] 00:28:03, localpref 100, from
10.255.14.174
                  AS path: I
                  > via t1-0/1/0.0
```

Configure BGP Route Target Filtering on the Route Reflector

This section illustrates how to enable BGP route target filtering on the route reflector for this example.

Configure the routing options on the route reflector as follows:

```
[edit]
routing-options {
  route-distinguisher-id 10.255.14.174;
  autonomous-system 200;
}
```

Configure the BGP protocol on the route reflector as follows:

```
[edit]
protocols {
  bgp {
    group rr-group {
      type internal;
      local-address 10.255.14.174;
      cluster 10.255.14.174;
      peer-as 200;
      neighbor 10.255.14.182 {
        description to_PE1_vpn12;
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
      neighbor 10.255.14.176 {
        description to_PE2_vpn06;
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
      neighbor 10.255.14.178 {
        description to_PE3_vpn08;
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
    }
  }
}
```

Once you have configured the route reflector in this manner, you should see the following when you issue the `show route table bgp.rtarget.0` command:

```

user@host> show route table bgp.rtarget.0

bgp.rtarget.0: 4 destinations, 8 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200:200:101/96
10.255.14.176      *[BGP/170] 00:29:03, localpref 100, from
                    AS path: I
                    > via t1-0/2/0.0
                    [BGP/170] 00:29:03, localpref 100, from
10.255.14.178      AS path: I
                    > via t3-0/1/1.0
                    [BGP/170] 00:29:03, localpref 100, from
10.255.14.182      AS path: I
                    > via t3-0/1/3.0
200:200:102/96
10.255.14.176      *[BGP/170] 00:29:03, localpref 100, from
                    AS path: I
                    > via t1-0/2/0.0
                    [BGP/170] 00:29:03, localpref 100, from
10.255.14.182      AS path: I
                    > via t3-0/1/3.0
200:200:103/96
10.255.14.176      *[BGP/170] 00:29:03, localpref 100, from
                    AS path: I
                    > via t1-0/2/0.0
                    [BGP/170] 00:29:03, localpref 100, from
10.255.14.178      AS path: I
                    > via t3-0/1/1.0
200:200:104/96
10.255.14.178      *[BGP/170] 00:29:03, localpref 100, from
                    AS path: I
                    > via t3-0/1/1.0

```

Configure BGP Route Target Filtering on Router PE3

The following section describes how to enable BGP route target filtering on router PE3 for this example.

Configure the routing options on router PE3 as follows:

```

[edit]
routing-options {
    route-distinguisher-id 10.255.14.178;
    autonomous-system 200;
}

```

Configure the BGP protocol on router PE3 as follows:

```
[edit]
protocols {
  bgp {
    group to_vpn04 {
      type internal;
      local-address 10.255.14.178;
      peer-as 200;
      neighbor 10.255.14.174 {
        family inet-vpn {
          unicast;
        }
        family route-target;
      }
    }
  }
}
```

Configure the vpn1 routing instance on router PE3 as follows:

```
[edit]
routing-instances {
  vpn1 {
    instance-type vrf;
    interface t3-0/0/0.0;
    vrf-target target:200:101;
    protocols {
      rip {
        group vpn1 {
          export bgp-routes;
          neighbor t3-0/0/0.0;
        }
      }
    }
  }
}
```

Configure the vpn3 routing instance on router PE3 as follows:

```
[edit]
routing-instances {
  vpn3 {
    instance-type vrf;
    interface t3-0/0/0.1;
    vrf-target target:200:103;
    protocols {
      rip {
        group vpn3 {
          export bgp-routes;
          neighbor t3-0/0/0.1;
        }
      }
    }
  }
}
```

Configure the `vpn4` routing instance on router PE3 as follows:

```
[edit]
routing-instances {
  vpn4 {
    instance-type vrf;
    interface t3-0/0/0.2;
    vrf-target target:200:104;
    protocols {
      rip {
        group vpn4 {
          export bgp-routes;
          neighbor t3-0/0/0.2;
        }
      }
    }
  }
}
```

Once you have configured the router PE3 in this manner, you should see the following when you issue the `show route table bgp.rtarget.0` command:

```
user@host> show route table bgp.rtarget.0

bgp.rtarget.0: 4 destinations, 7 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200:200:101/96
* [RTarget/5] 00:29:42
  Local
  [BGP/170] 00:29:30, localpref 100, from
10.255.14.174
  AS path: I
  > via t3-0/0/1.0
200:200:102/96
* [BGP/170] 00:29:29, localpref 100, from
10.255.14.174
  AS path: I
  > via t3-0/0/1.0
200:200:103/96
* [RTarget/5] 00:29:42
  Local
  [BGP/170] 00:29:30, localpref 100, from
10.255.14.174
  AS path: I
  > via t3-0/0/1.0
200:200:104/96
* [RTarget/5] 00:29:42
  Local
  [BGP/170] 00:29:30, localpref 100, from
10.255.14.174
  AS path: I
  > via t3-0/0/1.0
```

Route Origin for VPNs

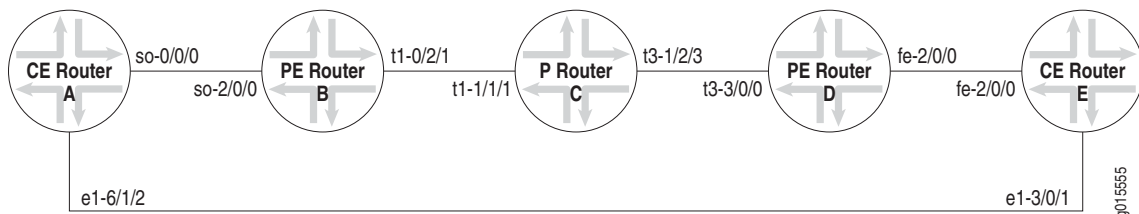
You can use route origin to prevent routes learned from one customer edge (CE) router marked with origin community from being advertised back to it from another CE router in the same autonomous system (AS).

In the example, the route origin is used to prevent routes learned from CE Router A that are marked with origin community from being advertised back to CE Router E by AS 200.

The following sections describe how to configure the route origin for a group of VPNs based on the topology illustrated in Figure 4:

- Configuring the Site of Origin Community on CE Router A on page 52
- Configuring the Community on CE Router A on page 53
- Applying the Policy Statement on CE Router A on page 53
- Configuring the Policy on PE Router D on page 54
- Configuring the Community on PE Router D on page 55
- Applying the Policy on PE Router D on page 55

Figure 4: Network Topology of Site of Origin Example



In this topology, CE Router A and CE Router E are in the same AS (AS200). They use external BGP (EBGP) to exchange routes with their respective provider edge (PE) routers, PE Router B and PE Router D. The two CE Routers have a back connection.

Configuring the Site of Origin Community on CE Router A

The following section describes how to configure CE Router A to advertise routes with a site of origin community to PE Router B for this example.



NOTE: In this example, direct routes are configured to be advertised, but any route can be configured.

Configure a policy to advertise routes with `soo` community on CE Router A as follows:

```
[edit]
policy-options {
  policy-statement export-to-my-isp {
    term a {
      from {
        protocol direct;
      }
      then {
        community add my-soo;
        accept;
      }
    }
  }
}
```

Configuring the Community on CE Router A

Configure the `soo` community on CE Router A as follows:

```
[edit]
policy-options {
  community my-soo {
    members origin:100:1;
  }
}
```

Applying the Policy Statement on CE Router A

Apply the `export-to-my-isp` policy statement as an export policy to the EBGP peering on the CE Router A as follows:

```
[edit]
protocols {
  bgp {
    group my_osp {
      export export-to-my-isp
    }
  }
}
```

When you issue the `show route receive-protocol bgp 10.12.99.2 detail` command, you should see the following routes originated from PE Router B with `soo` community :

```
user@host> show route receive-protocol bgp 10.12.99.2 detail

inet.0: 16 destinations, 16 routes (15 active, 0 holddown, 1 hidden)

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

vpn_blue.inet.0: 8 destinations, 10 routes (8 active, 0 holddown, 0 hidden)
* 10.12.33.0/30 (2 entries, 1 announced)
  Nexthop: 10.12.99.2
  AS path: 100 I
  Communities: origin:100:1

  10.12.99.0/30 (2 entries, 1 announced)
  Nexthop: 10.12.99.2
  AS path: 100 I
  Communities: origin:100:1

* 10.255.71.177/32 (1 entry, 1 announced)
  Nexthop: 10.12.99.2
  AS path: 100 I
  Communities: origin:100:1

* 192.168.64.0/21 (1 entry, 1 announced)
  Nexthop: 10.12.99.2
  AS path: 100 I
  Communities: origin:100:1
iso.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)

mpls.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

inet6.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

__juniper_private1__.inet6.0: 1 destinations, 1 routes (1 active, 0 holddown, 0
hidden)
```

Configuring the Policy on PE Router D

Configure a policy on PE Router D that prevents routes with `soo` community tagged by CE Router A from being advertised to CE Router E as follows:

```
[edit]
policy-options {
  policy-statement soo-ce1-policy {
    term a {
      from {
        community my-soo;
      }
      then {
        reject;
      }
    }
  }
}
```

Configuring the Community on PE Router D

Configure the community on PE Router D as follows:

```
[edit]
policy-options {
  community my-soo {
    members origin:100:1;
  }
}
```

Applying the Policy on PE Router D

To prevent routes learned from CE Router A from being advertised to CE Router E (the two routers can communicate these routes directly), apply the `soo-ce1-policy` policy statement as an export policy to the PE Router D and CE Router E EBGP session `vpn_blue`.

View the EBGP session on PE Router D using the `show routing-instances` command.

```
user@host# show routing-instances
vpn_blue {
  instance-type vrf;
  interface fe-2/0/0.0;
  vrf-target target:100:200;
  protocols {
    bgp {
      group ce2 {
        advertise-peer-as;
        peer-as 100;
        neighbor 10.12.99.6;
      }
    }
  }
}
```

Apply the `soo-ce1-policy` policy statement as an export policy to the PE Router D and CE Router E EBGP session `vpn_blue` as follows:

```
[edit routing-instances]
vpn_blue {
  protocols {
    bgp {
      group ce2{
        export soo-ce1-policy;
      }
    }
  }
}
```

