Implementing BGP Flowspec

Flowspec specifies procedures for the distribution of flow specification rules via BGP and defines procedure to encode flow specification rules as Border Gateway Protocol Network Layer Reachability Information (BGP NLRI) which can be used in any application. It also defines application for the purpose of packet filtering in order to mitigate (distributed) denial of service attacks.

Note For more information about BGP Flowspec and complete descriptions of the BGP Flowspec commands listed in this module, see the *BGP Flowspec Commands* chapter in the *Routing Command Reference for Cisco CRS Routers*.

Feature History for Implementing BGP Flowspec

Release 5.2.0	This feature was introduced.

• BGP Flow Specification, on page 1

BGP Flow Specification

The BGP flow specification (flowspec) feature allows you to rapidly deploy and propagate filtering and policing functionality among a large number of BGP peer routers to mitigate the effects of a distributed denial-of-service (DDoS) attack over your network.

In traditional methods for DDoS mitigation, such as RTBH (remotely triggered blackhole), a BGP route is injected advertising the website address under attack with a special community. This special community on the border routers sets the next hop to a special next hop to discard/null, thus preventing traffic from suspect sources into your network. While this offers good protection, it makes the Server completely unreachable.

BGP flowspec, on the other hand, allows for a more granular approach and lets you effectively construct instructions to match a particular flow with source, destination, L4 parameters and packet specifics such as length, fragment and so on. Flowspec allows for a dynamic installation of an action at the border routers to either:

- Drop the traffic
- Inject it in a different VRF for analysis or

- Allow it, but police it at a specific defined rate

Thus, instead of sending a route with a special community that the border routers must associate with a next hop to drop in their route policy language, BGP flowspec sends a specific flow format to the border routers instructing them to create a sort of ACL with class-map and policy-map to implement the rule you want advertised. In order to accomplish this, BGP flowspec adds a new NLRI (network layer reachability information) to the BGP protocol. Information About Implementing BGP Flowspec , on page 3 provides details on flow specifications, supported matching criteria and traffic filtering action.

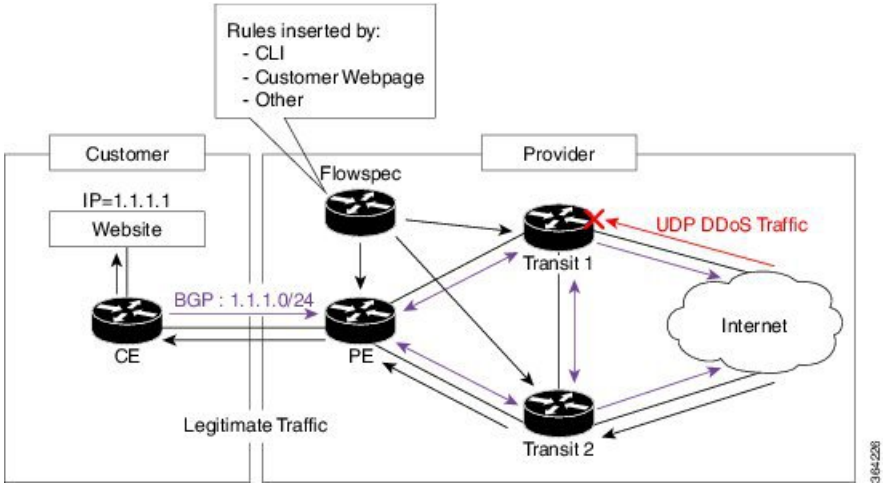
Limitations

These limitations apply for BGP flowspec:

- Flowspec is not supported on Cisco CRS-1 line cards (MSC-A, MSC-B, FP-40).
- Flowspec is not supported on subscriber and satellite interfaces.
- A maximum of five multi-value range can be specified in a flowspec rule.
- A mix of address families is not allowed in flowspec rules.
- In multiple match scenario, only the first matching flowspec rule will be applied.
- There are 1500 TCAM entries. If each rule translates into one entry, then a maximum of 1500 flowspec rules are supported per system.
- QoS takes precedence over BGP flowspec.
- You cannot do explicit matching with respect to MPLS.
- This feaure does not function if another configured feature causes all the IPv6 parsing to go to slowpath. For example, this feature does not function when port mirroring in configured.
- The feature is supported on Cisco ASR 9000 Third Generation Ethernet Line Cards. Other line cards in the router do not suport this feature.
- If there is a bundle in the router, and if one of the line card interfaces in the bundle belongs to a line card that is not Cisco ASR 9000 Third Generation Ethernet Line Card, you may observe inconsistent behaviour.
- You cannot enable this feature on satellite interfaces.
- This feature supports TCP, UDP, and ICMPv6.

BGP Flowspec Conceptual Architecture

In this illustration, a Flowspec router (controller) is configured on the Provider Edge with flows (match criteria and actions). The Flowspec router advertises these flows to the other edge routers and the AS (that is, Transit 1, Transit 2 and PE). These transit routers then install the flows into the hardware. Once the flow is installed into the hardware, the transit routers are able to do a lookup to see if incoming traffic matches the defined flows and take suitable action. The action in this scenario is to 'drop' the DDoS traffic at the edge of the network itself and deliver only clean and legitimate traffic to the Customer Edge.



The ensuing section provides an example of the CLI configuration of how flowspec works. First, on the Flowspec router you define the match-action criteria to take on the incoming traffic. This comprises the PBR portion of the configuration. The **service-policy type** defines the actual PBR policy and contains the combination of match and action criteria which must be added to the flowspec. In this example, the policy is added under address-family IPv4, and hence it is propagated as an IPv4 flowspec rule.

```
Flowspec router CLI example:

class-map type traffic match-all cm1
     match source-address ipv4 100.0.0.0/24

policy-map type pbr pm1
    class type traffic cm1
      drop


flowspec
    address-family ipv4
       service-policy type pbr pm0

Transient router CLI:

flowspec
    address-family ipv4
       service-policy type pbr pm1
```

For detailed procedural information and commands used for configuring Flowspec, see Configuring BGP Flowspec with ePBR, on page 10.

Information About Implementing BGP Flowspec

To implement BGP Flowspec, you need to understand the following concepts:

Flow Specifications

A flow specification is an n-tuple consisting of several matching criteria that can be applied to IP traffic. A given IP packet is said to match the defined flow if it matches all the specified criteria. A given flow may be associated with a set of attributes, depending on the particular application; such attributes may or may not include reachability information (that is, NEXT_HOP).

Every flow-spec route is effectively a rule, consisting of a matching part (encoded in the NLRI field) and an action part (encoded as a BGP extended community). The BGP flowspec rules are converted internally to equivalent C3PL policy representing match and action parameters. The match and action support can vary based on underlying platform hardware capabilities. Supported Matching Criteria and Actions, on page 4 and Traffic Filtering Actions, on page 7 provides information on the supported match (tuple definitions) and action parameters.

Supported Matching Criteria and Actions

A Flow Specification NLRI type may include several components such as destination prefix, source prefix, protocol, ports, and so on. This NLRI is treated as an opaque bit string prefix by BGP. Each bit string identifies a key to a database entry with which a set of attributes can be associated. This NLRI information is encoded using MP_REACH_NLRI and MP_UNREACH_NLRI attributes. Whenever the corresponding application does not require Next-Hop information, this is encoded as a 0-octet length Next Hop in the MP_REACH_NLRI attribute and ignored on receipt. The NLRI field of the MP_REACH_NLRI and MP_UNREACH_NLRI is encoded as a 1- or 2-octet NLRI length field followed by a variable-length NLRI value. The NLRI length is expressed in octets.

The Flow specification NLRI-type consists of several optional sub-components. A specific packet is considered to match the flow specification when it matches the intersection (AND) of all the components present in the specification. The following are the supported component types or tuples that you can define:

Tuple definition possibilities

BGP Flowspec NLRI type	QoS match fields	Description and Syntax Construction	Value input method
Type 1	IPv4 or IPv6 Destination address	Defines the destination prefix to match. Prefixes are encoded in the BGP UPDATE messages as a length in bits followed by enough octets to contain the prefix information. Encoding: <type (1 octet), prefix length (1 octet), prefix> **Syntax:** **match destination-address** {**ipv4** | **ipv6**} *address/mask length*	Prefix length
Type 2	IPv4 or IPv6 Source address	Defines the source prefix to match. Encoding: <type (1 octet), prefix-length (1 octet), prefix> **Syntax:** **match source-address** {**ipv4** | **ipv6**} *address/mask length*	Prefix length

Type 3	IPv4 last next header or IPv6Protocol	Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets. Encoding: <type (1 octet), [op, value]+> **Syntax:** Type 3: **match protocol** {*protocol-value* | *min-value* *−max-value*}	Multi value range
Type 4	IPv4 or IPv6 source or destination port	Defines a list of {operation, value} pairs that matches source or destination TCP/UDP ports. Values are encoded as 1- or 2-byte quantities. Port, source port, and destination port components evaluate to FALSE if the IP protocol field of the packet has a value other than TCP or UDP, if the packet is fragmented and this is not the first fragment, or if the system in unable to locate the transport header. Encoding: <type (1 octet), [op, value]+> **Syntax:** **match source-port** {*source-port-value* | *min-value* *−max-value*} **match destination-port** {*destination-port-value* | *min-value* *−max-value*}	Multi value range
Type 5	IPv4 or IPv6 destination port	Defines a list of {operation, value} pairs used to match the destination port of a TCP or UDP packet. Values are encoded as 1- or 2-byte quantities. Encoding: <type (1 octet), [op, value]+> **Syntax:** **match destination-port** {*destination-port-value* | [*min-value* *− max-value*]}	Multi value range
Type 6	IPv4 or IPv6 Source port	Defines a list of {operation, value} pairs used to match the source port of a TCP or UDP packet. Values are encoded as 1- or 2-byte quantities. Encoding: <type (1 octet), [op, value]+> **Syntax:** **match source-port** {*source-port-value* | [*min-value* *− max-value*]}	Multi value range

Type 7	IPv4 or IPv6 ICMP type	Defines a list of {operation, value} pairs used to match the type field of an ICMP packet. Values are encoded using a single byte. The ICMP type and code specifiers evaluate to FALSE whenever the protocol value is not ICMP. Encoding: <type (1 octet), [op, value]+> **Syntax:** **match** {**ipv4** | **ipv6**} **icmp-type** {*value* | *min-value* *-max-value*}	Single value **Note** Multi value range is not supported.
Type 8	IPv4 or IPv6 ICMP code	Defines a list of {operation, value} pairs used to match the code field of an ICMP packet. Values are encoded using a single byte. Encoding: <type (1 octet), [op, value]+> **Syntax:** **match** {**ipv4** | **ipv6**} **icmp-code** {*value* | *min-value* *-max-value*}	Single value **Note** Multi value range is not supported.
Type 9	IPv4 or IPv6 TCP flags (2 bytes include reserved bits) **Note** Reserved and NS bit not supported	Bitmask values can be encoded as a 1- or 2-byte bitmask. When a single byte is specified, it matches byte 13 of the TCP header, which contains bits 8 through 15 of the 4th 32-bit word. When a 2-byte encoding is used, it matches bytes 12 and 13 of the TCP header with the data offset field having a "don't care" value. As with port specifier, this component evaluates to FALSE for packets that are not TCP packets. This type uses the bitmask operand format, which differs from the numeric operator format in the lower nibble. Encoding: <type (1 octet), [op, bitmask]+> **Syntax:** **match tcp-flag** *value* **bit-mask** *mask_value*	Bit mask
Type 10	IPv4 or IPv6 Packet length	Match on the total IP packet length (excluding Layer 2, but including IP header). Values are encoded using 1- or 2-byte quantities. Encoding: <type (1 octet), [op, value]+> **Syntax:** **matchpacket length** {*packet-length-value* | *min-value* *-max-value*}	Multi value range

Type 11	IPv4 or IPv6 DSCP	Defines a list of {operation, value} pairs used to match the 6-bit DSCP field. Values are encoded using a single byte, where the two most significant bits are zero and the six least significant bits contain the DSCP value. Encoding: <type (1 octet), [op, value]+> **Syntax:** **match dscp** {*dscp-value* | *min-value* **-** *max-value*}	Multi value range
Type 12	IPv4 or IPv6 Fragmentation bits	Identifies a fragment-type as the match criterion for a class map. Encoding: <type (1 octet), [op, bitmask]+> **Syntax:** **match fragment type** [**is-fragment**]	Bit mask

In a given flowspec rule, multiple action combinations can be specified without restrictions. However, address family mixing between matching criterion and actions are not allowed. For example, IPv4 matches cannot be combined with IPv6 actions and vice versa.

Note Redirect IP Nexthop is only supported in default VRF cases.

Traffic Filtering Actions, on page 7 provides information on the actions that can be associated with a flow. Configuring BGP Flowspec with ePBR, on page 10 explains the procedure to configure BGP flowpsec with the required tuple definitions and action sequences.

Traffic Filtering Actions

The default action for a traffic filtering flow specification is to accept IP traffic that matches that particular rule. The following extended community values can be used to specify particular actions:

Type	Extended Community	PBR Action	Description
0x8006	traffic-rate 0 traffic-rate <rate>	Drop Police	The traffic-rate extended community is a non-transitive extended community across the autonomous-system boundary and uses following extended community encoding: The first two octets carry the 2-octet id, which can be assigned from a 2-byte AS number. When a 4-byte AS number is locally present, the 2 least significant bytes of such an AS number can be used. This value is purely informational. The remaining 4 octets carry the rate information in IEEE floating point [IEEE.754.1985] format, units being bytes per second. A traffic-rate of 0 should result on all traffic for the particular flow to be discarded. **Command syntax** police rate < > | drop

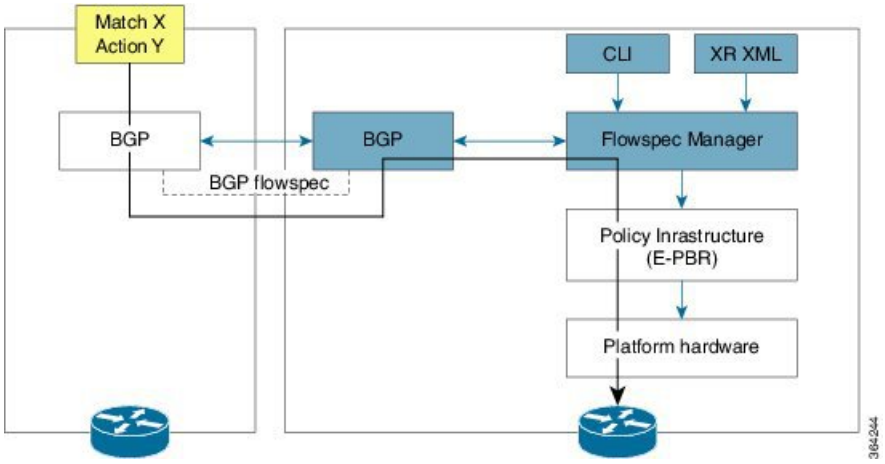
0x8008	redirect-vrf	Redirect VRF	The redirect extended community allows the traffic to be redirected to a VRF routing instance that lists the specified route-target in its import policy. If several local instances match this criteria, the choice between them is a local matter (for example, the instance with the lowest Route Distinguisher value can be elected). This extended community uses the same encoding as the Route Target extended community [RFC4360]. **Command syntax based on route-target** redirect {ipv6} extcommunity rt <route_target_string>
0x8009	traffic-marking	Set DSCP	The traffic marking extended community instructs a system to modify the differentiated service code point (DSCP) bits of a transiting IP packet to the corresponding value. This extended community is encoded as a sequence of 5 zero bytes followed by the DSCP value encoded in the 6 least significant bits of 6th byte. **Command syntax** set dscp <6 bit value> set ipv6 traffic-class <8 bit value>
0x0800	Redirect IP NH	Redirect IPv4 or IPv6 Nexthop	Announces the reachability of one or more flowspec NLRI. When a BGP speaker receives an UPDATE message with the redirect-to- IP extended community it is expected to create a traffic filtering rule for every flow-spec NLRI in the message that has this path as its best path. The filter entry matches the IP packets described in the NLRI field and redirects them or copies them towards the IPv4 or IPv6 address specified in the 'Network Address of Next- Hop' field of the associated MP_REACH_NLRI. **Note** The redirect-to-IP extended community is valid with any other set of flow-spec extended communities except if that set includes a redirect-to-VRF extended community (type 0x8008) and in that case the redirect-to-IP extended community should be ignored. **Command syntax** redirect {ipv6} next-hop <ipv4/v6 address> {ipv4/v6 address}

Configure a Class Map, on page 11 explains how you can configure specific match criteria for a class map.

BGP Flowspec Client-Server (Controller) Model and Configuration with ePBR

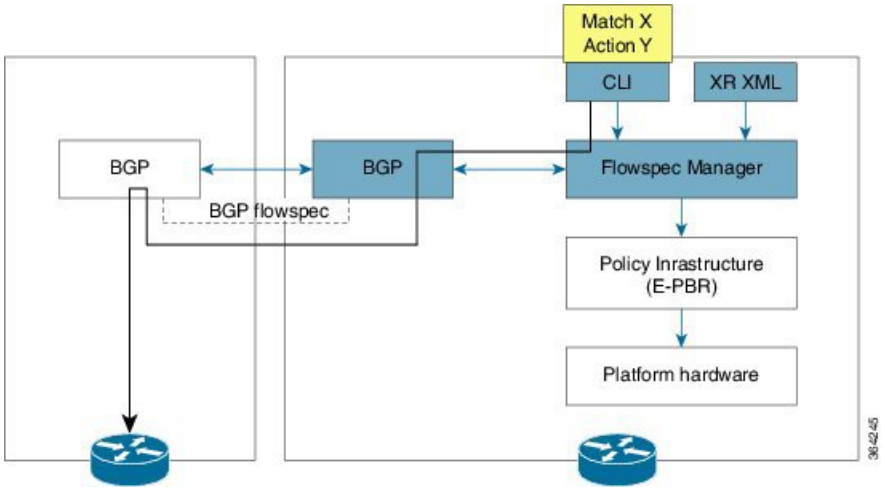
The BGP Flowspec model comprises of a Client and a Server (Controller). The Controller is responsible for sending or injecting the flowspec NRLI entry. The client (acting as a BGP speaker) receives that NRLI and programs the hardware forwarding to act on the instruction from the Controller. An illustration of this model is provided below.

BGP Flowspec Client



Here, the Controller on the left-hand side injects the flowspec NRLI, and the client on the right-hand side receives the information, sends it to the flowspec manager, configures the ePBR (Enhanced Policy-based Routing) infrastructure, which in turn programs the hardware from the underlaying platform in use.

BGP Flowspec Controller



The Controller is configured using CLI to provide that entry for NRLI injection.

BGP Flowspec Configuration

- **BGP-side**: You must enable the new address family for advertisement. This procedure is applicable for both the Client and the Controller. Enable BGP Flowspec, on page 10 explains the procedure.

  Client-side: No specific configuration, except availability of a flowspec-enabled peer.

- **Controller-side**: This includes the policy-map definition and the association to the ePBR configuration consists of two procedures: the class definition, and using that class in ePBR to define the action. The following topics explain the procedure:

  - Configure a Policy Map, on page 13
  - Configure a Class Map, on page 11
  - Link BGP Flowspec to ePBR Policies , on page 15

Configuring BGP Flowspec with ePBR

The following sections explain the procedures for configuring BGP flowspec with ePBR.

Use the following procedures to enable and configure the BGP flowspec feature:

Note To save configuration changes, you must commit changes when the system prompts you.

Enable BGP Flowspec

You must enable the address family for propagating the BGP flowspec policy on both the Client and Server using the following steps:

SUMMARY STEPS

1. **configure**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** | **vpnv4** | **vpnv6** } **flowspec**
4. **exit**
5. **neighbor** *ip-address*
6. **remote-as** *as-number*
7. **address-family** { **ipv4** | **ipv6** } **flowspec**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router bgp** *as-number* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 100`	Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 3	**address-family** { **ipv4**	**ipv6**	**vpnv4**	**vpnv6** } **flowspec** **Example:** `RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 flowspec`	Specifies either the IPv4, IPv6, vpn4 or vpn6 address family and enters address family configuration submode, and initializes the global address family for flowspec policy mapping.
Step 4	**exit** **Example:**	Returns the router to BGP configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-bgp-af)# exit`	
Step 5	**neighbor** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config-bgp)#neighbor 1.1.1.1`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 6	**remote-as** *as-number* **Example:** `RP/0/RP0/CPU0:router(config-bgp-nbr)#remote-as 100`	Assigns a remote autonomous system number to the neighbor.
Step 7	**address-family** { **ipv4** | **ipv6** } **flowspec** **Example:** `RP/0/RP0/CPU0:router(config-bgp)# address-family ipv4 flowspec`	Specifies an address family and enters address family configuration submode, and initializes the global address family for flowspec policy mapping.

Configuring an address family for flowspec policy mapping: Example

```
router bgp 100
 address-family ipv4 flowspec
 ! Initializes the global address family
 address-family ipv6 flowspec
 !
 neighbor 1.1.1.1
  remote-as 100
  address-family ipv4 flowspec
  ! Ties it to a neighbor configuration
  address-family ipv6 flowspec
  !
```

Configure a Class Map

In order to associate the ePBR configuration to BGP flowspec you must perform these sub-steps: define the class and use that class in ePBR to define the action. The steps to define the class include:

SUMMARY STEPS

1. **configure**
2. **class-map [type traffic] [match-all]** *class-map-name*
3. **match** *match-statement*

4. **end-class-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**class-map [type traffic] [match-all]** *class-map-name* **Example:** `RP/0/RP0/CPU0:router(config)# class-map type traffic match all classc1`	Creates a class map to be used for matching packets to the class whose name you specify and enters the class map configuration mode. If you specify **match-any** , one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class. This is the default. If you specify **match-all** , the traffic must match all the match criteria.
Step 3	**match** *match-statement* **Example:** `RP/0/RP0/CPU0:router(config-cmap)# match protocol ipv4 1 60`	Configures the match criteria for a class map on the basis of the statement specified. Any combination of tuples 1-13 match statements can be specified here. The tuple definition possibilities include: • Type 1: **match destination-address** {**ipv4** | **ipv6**} *address/mask length* • Type 2: **match source-address** {**ipv4** | **ipv6**} *address/mask length* • Type 3: **match protocol** {*protocol-value* | *min-value* *−max-value*} **Note** In case of IPv6, it will map to last next-header. • Type 4: Create two class-maps: one with source-port and another with destination-port: • **match source-port** {*source-port-value* | *min-value* *−max-value*} • **match destination-port** {*destination-port-value* | *min-value* *−max-value*} **Note** These are applicable only for TCP and UDP protocols. • Type 5: **match destination-port** {*destination-port-value* | [*min-value* − *max-value*]} • Type 6: **match source-port** {*source-port-value* | [*min-value* − *max-value*]} • Type 7: **match** {**ipv4** | **ipv6**}**icmp-code** {*value* | *min-value* *−max-value*} • Type 8: **match** {**ipv4** | **ipv6**}**icmp-type** {*value* | *min-value* *−max-value*}

	Command or Action	Purpose
		• Type 9: **match tcp-flag** *value* **bit-mask** *mask_value* • Type 10: **matchpacket length** {*packet-length-value* | *min-value* −*max-value*}} • Type 11: **match dscp** {*dscp-value* | *min-value* −*max-value*} • Type 12: **match fragment-type** {**dont-fragment is-fragment first-fragment last-fragment**} • Type 13: **match ipv6 flow-label ipv4 flow-label** {*value* | *min-value* −*max-value*} Supported Matching Criteria and Actions, on page 4 provides conceptual information on these match parameters. *BGP Flowspec Commands* in the *Routing Command Reference for Cisco CRS Routers* guide provides additional details on the various commands used for BGP flowspec configuration.
Step 4	**end-class-map** **Example:** `RP/0/RP0/CPU0:router(config-cmap)# end-class-map`	Ends the class map configuration and returns the router to global configuration mode.

What to do next

Associate the class defined in this procedure to a PBR policy as described in Configure a Policy Map, on page 13.

Configure a Policy Map

This procedure helps you define a policy map and associate it with traffic class you configured previously in Configure a Class Map, on page 11 .

SUMMARY STEPS

1. **configure**
2. **policy-map type pbr** *policy-map*
3. **class** *class-name*
4. **class type traffic** *class-name*
5. *action*
6. **exit**
7. **end-policy-map**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

	Command or Action	Purpose
Step 2	**policy-map type pbr** *policy-map* **Example:** `RP/0/RP0/CPU0:router(config)# policy-map type pbr policyp1`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters the policy map configuration mode.
Step 3	**class** *class-name* **Example:** `RP/0/RP0/CPU0:router(config-pmap)# class class1`	Specifies the name of the class whose policy you want to create or change.
Step 4	**class type traffic** *class-name* **Example:** `RP/0/RP0/CPU0:router(config-pmap)# class type traffic classc1`	Associates a previously configured traffic class with the policy map, and enters control policy-map traffic class configuration mode.
Step 5	*action* **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# set dscp 5`	Define extended community actions as per your requirement. The options include: • Traffic rate: **police rate** *rate* • Redirect VRF: **redirect** { **ipv4ipv6** }**extcommunity rt** *route_target_string* • Traffic Marking: **set** { **dscp** *rate* | **destination-address** {**ipv4** | **ipv6**} *8-bit value*} • Redirect IP NH: **redirect** { **ipv4ipv6** } **nexthop** *ipv4 addressipv6 address* { *ipv4 addressipv6 address*} Traffic Filtering Actions, on page 7 provides conceptual information on these extended community actions.
Step 6	**exit** **Example:** `RP/0/RP0/CPU0:router(config-pmap-c)# exit`	Returns the router to policy map configuration mode.
Step 7	**end-policy-map** **Example:** `RP/0/RP0/CPU0:router(config-cmap)# end-policy-map`	Ends the policy map configuration and returns the router to global configuration mode.

What to do next

Perform VRF and flowspec policy mapping for distribution of flowspec rules using the procedure explained in Link BGP Flowspec to ePBR Policies , on page 15

Link BGP Flowspec to ePBR Policies

For BGP flowspec, an ePBR policy is applied on a per VRF basis, and this policy is applied on all the interfaces that are part of the VRF. If you have already configured a ePBR policy on an interface, it will not be overwritten by the BGP flowspec policy. If you remove the policy from an interface, ePBR infrastructure will automatically apply BGP flowspec policy on it, if one was active at the VRF level.

Note At a time only one ePBR policy can be active on an interface.

SUMMARY STEPS

1. **configure**
2. **flowspec**
3. **local-install interface-all**
4. **address-family ipv4**
5. **local-install interface-all**
6. **service-policy type pbr** *policy-name*
7. **exit**
8. **address-family ipv6**
9. **local-install interface-all**
10. **service-policy type pbr** *policy-name*
11. **vrf** *vrf-name*
12. **address-family ipv4**
13. **local-install interface-all**
14. **service-policy type pbr** *policy-name*
15. **exit**
16. **address-family ipv6**
17. **local-install interface-all**
18. **service-policy type pbr** *policy-name*
19. **commit**
20. **exit**
21. **show flowspec** { **afi-all** | **client** | **ipv4** | **ipv6** | **summary** | **vrf**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**flowspec** **Example:**	Enters the flowspec configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# flowspec`	
Step 3	**local-install interface-all** **Example:** `RP/0/RP0/CPU0:router(config-flowspec)# local-install interface-all`	(Optional) Installs the flowspec policy on all interfaces.
Step 4	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-flowspec)# address-family ipv4`	Specifies either an IPv4 address family and enters address family configuration submode.
Step 5	**local-install interface-all** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-af)# local-install interface-all`	(Optional) Installs the flowspec policy on all interfaces under the subaddress family.
Step 6	**service-policy type pbr** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config-flowspec-af)# service-policy type pbr policys1`	Attaches a policy map to an IPv4 interface to be used as the service policy for that interface.
Step 7	**exit** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-af)# exit`	Returns the router to flowspec configuration mode.
Step 8	**address-family ipv6** **Example:** `RP/0/RP0/CPU0:router(config-flowspec)# address-family ipv6`	Specifies an IPv6 address family and enters address family configuration submode.
Step 9	**local-install interface-all** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-af)# local-install interface-all`	(Optional) Installs the flowspec policy on all interfaces under the subaddress family.

	Command or Action	Purpose
Step 10	**service-policy type pbr** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config-flowspec-af)# service-policy type pbr policys1`	Attaches a policy map to an IPv6 interface to be used as the service policy for that interface.
Step 11	**vrf** *vrf-name* **Example:** `RP/0/RP0/CPU0:router(config-flowspec)# vrf vrf1`	Configures a VRF instance and enters VRF flowspec configuration submode.
Step 12	**address-family ipv4** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf)# address-family ipv4`	Specifies an IPv4 address family and enters address family configuration submode.
Step 13	**local-install interface-all** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf-af)# local-install interface-all`	(Optional) Installs the flowspec policy on all interfaces under the subaddress family.
Step 14	**service-policy type pbr** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf-af)# service-policy type pbr policys1`	Attaches a policy map to an IPv4 interface to be used as the service policy for that interface.
Step 15	**exit** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf-af)# exit`	Returns the router to VRF flowspec configuration submode.
Step 16	**address-family ipv6** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf)# address-family ipv6`	Specifies either an IPv6 address family and enters address family configuration submode.
Step 17	**local-install interface-all** **Example:**	(Optional) Installs the flowspec policy on all interfaces under the subaddress family.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-flowspec-vrf-af)# local-install interface-all`	
Step 18	**service-policy type pbr** *policy-name* **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf-af)# service-policy type pbr policys1`	Attaches a policy map to an IPv6 interface to be used as the service policy for that interface.
Step 19	**commit**	
Step 20	**exit** **Example:** `RP/0/RP0/CPU0:router(config-flowspec-vrf-af)# exit`	Returns the router to flowspec configuration mode.
Step 21	**show flowspec** { **afi-all** | **client** | **ipv4** | **ipv6** | **summary** | **vrf** **Example:** `RP/0/RP0/CPU0:routershow flowspec vrf vrf1 ipv4 summary`	(Optional) Displays flowspec policy applied on an interface.

Verify BGP Flowspec

Use these different **show** commands to verify your flowspec configuration. For instance, you can use the associated flowspec and BGP show commands to check whether flowspec rules are present in your table, how many rules are present, the action that has been taken on the traffic based on the flow specifications you have defined and so on.

SUMMARY STEPS

1. **show processes flowspec_mgr location all**
2. **show flowspec summary**
3. **show flowspec vrf** *vrf_name* \| **all** { **afli-all** \| **ipv4** \| **ipv6** }
4. **show bgp ipv4 flowspec**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show processes flowspec_mgr location all** **Example:**	Specifies whether the flowspec process is running on your system or not. The flowspec manager is responsible for

	Command or Action	Purpose

	`# show processes flowspec_mgr location all`
	`node:  node0_3_CPU0`
	`---`
	`Job Id: 10`
	`PID: 43643169`
	`Executable path:`
	`/disk0/iosxr-fwding-5.2.CSC33695-015.i/bin/flowspec_mgr`
	`Instance #: 1`
	`Version ID: 00.00.0000`
	`Respawn: ON`
	`Respawn count: 331`
	`Max. spawns per minute: 12`
	`Last started: Wed Apr  9 10:42:13 2014`
	`Started on config: cfg/gl/flowspec/`
	`Process group: central-services core: MAINMEM`
	`startup_path: /pkg/startup/flowspec_mgr.startup`
	`Ready: 1.113s`
	`Process cpu time: 0.225 user, 0.023 kernel, 0.248`
	` total`
	`JID  TID CPU Stack pri state  TimeInState`
	`HR:MM:SS:MSEC  NAME`
	`1082  1  0  112K  10 Receive  2:50:23:0508`
	`0:00:00:0241 flowspec_mgr`
	`1082  2  1  112K  10 Sigwaitinfo 2:52:42:0583`
	`  0:00:00:0000 flowspec_mgr`
		creating, distributing and installing the flowspec rules on the hardware.

Step 2	**show flowspec summary**
	Example:
	`# show flowspec summary`
	
	`FlowSpec Manager Summary:`
	`  Tables:  2`
	`  Flows:  1`
	`RP/0/3/CPU0:RA01_R4#`
		Provides a summary of the flowspec rules present on the entire node. In this example, the 2 table indicate that IPv4 and IPv6 has been enabled, and a single flow has been defined across the entire table.

Step 3	**show flowspec vrf** *vrf_name* | **all** { **afli-all** | **ipv4** | **ipv6** }
	Example:
	
	`# show flowspec vrf default ipv4 summary`
	
	`Flowspec VRF+AFI table summary:`
	`VRF: default`
	`  AFI: IPv4`
	`  Total Flows:  1`
	`  Total Service Policies:  1`
	`RP/0/3/CPU0:RA01_R4#`
	`---`
	`# show flowspec vrf default ipv6 summary`
	
	`Flowspec VRF+AFI table summary:`
	`VRF: default`
	`  AFI: IPv6`
	`  Total Flows:  0`
	`  Total Service Policies:  0`
	`RP/0/3/CPU0:RA01_R4#`
	`---`
	`# show flowspec vrf all afi-all summary`
		In order to obtain more granular information on the flowspec, you can filter the show commands based on a particular address-family or by a specific VRF name. In this example, 'vrf default' indicates that the flowspec has been defined on the default table. The 'IPv4 summary' shows the IPv4 flowspec rules present on that default table. As there are no IPv6s configured, the value shows 'zero' for ipv6 summary 'Table Flows' and 'Policies' parameters. 'VRF all' displays information across all the VRFs configured on the table and afli-all displays information for all address families (IPv4 and IPv6).
		
		The **detail** option displays the 'Matched', 'Transmitted, ' and 'Dropped' fields. These can be used to see if the flowspec rule you have defined is in action or not. If there is any traffic that takes this match condition, it indicates if any action has been taken (that is, how many packets were matched and whether these packets have been transmitted or dropped.

	Command or Action	Purpose
	Flowspec VRF+AFI table summary: VRF: default AFI: IPv4 Total Flows: 1 Total Service Policies: 1 VRF: default AFI: IPv6 Total Flows: 0 Total Service Policies: 0 --- # show flowspec vrf default ipv4 Dest:110.1.1.0/24, Source:10.1.1.0/24,DPort:>=120&<=130, SPort:>=25&<=30,DSCP:=30 detail AFI: IPv4 Flow :Dest:110.1.1.0/24,Source:10.1.1.0/24, DPort:>=120&<=130,SPort:>=25&<=30,DSCP:=30 Actions :Traffic-rate: 0 bps (bgp.1) Statistics (packets/bytes) Matched : 0/0 Transmitted : 0/0 Dropped : 0/0	
Step 4	**show bgp ipv4 flowspec** **Example:** # show bgp ipv4 flowspec Dest:110.1.1.0/24,Source:10.1.1.0/24, DPort:>=120&<=130,SPort:>=25&<=30,DSCP:=30/208 BGP routing table entry for Dest:110.1.1.0/24, Source:10.1.1.0/24,Proto:=47,DPort:>=120&<=130,SPort:>=25&<=30,DSCP:=30/208 <snip> Paths: (1 available, best #1) Advertised to update-groups (with more than one peer): 0.3 Path #1: Received by speaker 0 Advertised to update-groups (with more than one peer): 0.3 Local 0.0.0.0 from 0.0.0.0 (3.3.3.3) Origin IGP, localpref 100, valid, redistributed, best, group-best Received Path ID 0, Local Path ID 1, version 42 Extended community: FLOWSPEC Traffic-rate:100,0	Use this command to verify if a flowspec rule configured on the controller router is available on the BGP side. In this example, 'redistributed' indicates that the flowspec rule is not internally originated, but one that has been redistributed from the flowspec process to BGP. The extended community (BGP attribute used to send the match and action criteria to the peer routers) you have configured is also displayed here. In this example, the action defined is to rate limit the traffic.

Preserving Redirect Nexthop

You can explicitly configure redirect nexthop as part of the route specification. Redirect nexthop is encoded as the MP_REACH nexthop in the BGP flowspec NLRI along with the associated extended community. Recipient of such a flowspec route redirects traffic as per FIB lookup for the redirect nexthop, the nexthop can possibly resolve over IP or MPLS tunnel. As the MP_REACH nexthop can be overwritten at a eBGP

boundary, for cases where the nexthop connectivity spans multiple AS's, the nexthop can be preserved through the use of the unchanged knob.

SUMMARY STEPS

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **address-family** { **ipv4** | **ipv6** }
5. **flowspec next-hop unchanged**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router bgp** *as-number* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 100`	Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 3	**neighbor** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 100 neighbor 1.1.1.1`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 4	**address-family** { **ipv4** | **ipv6** } **Example:** `RP/0/RP0/CPU0:router(config-bgp)# router bgp 100 neighbor 1.1.1.1 address-family ipv4`	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode, and initializes the global address family.
Step 5	**flowspec next-hop unchanged** **Example:** `RP/0/RP0/CPU0:router(config-bgp)# router bgp 100 neighbor 1.1.1.1 address-family ipv4 flowspec next-hop unchanged`	Preserves the next-hop for the flowspec unchanged.

Validate BGP Flowspec

BGP Flowspec validation is enabled by default for flowspec SAFI routes for IPv4 or IPv6. VPN routes are not subject to the flow validation. A flow specification NLRI is validated to ensure that any one of the following conditions holds true for the functionality to work:

- The originator of the flow specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification.

- There are no more specific unicast routes, when compared with the flow destination prefix, that have been received from a different neighboring AS than the best-match unicast route, which has been determined in the previous condition.
- The AS_PATH and AS4_PATH attribute of the flow specification are empty.
- The AS_PATH and AS4_PATH attribute of the flow specification does not contain AS_SET and AS_SEQUENCE segments.

Any path which does not meet these conditions, is appropriately marked by BGP and not installed in flowspec manager. Additionally, BGP enforces that the last AS added within the AS_PATH and AS4_PATH attribute of a EBGP learned flow specification NLRI must match the last AS added within the AS_PATH and AS4_PATH attribute of the best-match unicast route for the destination prefix embedded in the flow specification. Also, when the redirect-to-IP extended community is present, by default, BGP enforces the following check when receiving a flow-spec route from an eBGP peer:

If the flow-spec route has an IP next-hop X and includes a redirect-to-IP extended community, then the BGP speaker discards the redirect-to-ip extended community (and not propagate it further with the flow-spec route) if the last AS in the AS_PATH or AS4_PATH attribute of the longest prefix match for X does not match the AS of the eBGP peer.

Disable Flowspec Redirect and Validation, on page 23 explains the procedure to disable BGP flowspec validation.

Disabling BGP Flowspec

This procedure disables BGP flowspec policy on an interface.

SUMMARY STEPS

1. **configure**
2. **interface** *type interface-path-id*
3. **{ ipv4 | ipv6 } flowspec disable**
4. **commit**

DETAILED STEPS

Step 1 **configure**

Step 2 **interface** *type interface-path-id*

Example:

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/1/1/1
```

Configures an interface and enters the interface configuration mode.

Step 3 **{ ipv4 | ipv6 } flowspec disable**

Example:

```
RP/0/RP0/CPU0:router(config-if)# ipv4 flowspec disable
```

Disable flowspec policy on the selected interface.

Step 4 **commit**

Disable flowspec on the interface

The following example shows you how you can disable BGP flowspec on an interface, and apply another PBR policy:

```
Interface GigabitEthernet 0/0/0/0
 flowspec [ipv4/ipv6] disable
int g0/0/0/1
service policy type pbr test_policy
 !
!
```

Disable Flowspec Redirect and Validation

You can disable flowspec validation as a whole for eBGP sessions by means of configuring an explicit knob.

SUMMARY STEPS

1. **configure**
2. **router bgp** *as-number*
3. **neighbor** *ip-address*
4. **address-family** { **ipv4** | **ipv6** }
5. **flowspec validation** { **disable** | **redirect disable** }

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**router bgp** *as-number* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 100`	Specifies the autonomous system number and enters the BGP configuration mode, allowing you to configure the BGP routing process.
Step 3	**neighbor** *ip-address* **Example:** `RP/0/RP0/CPU0:router(config)# router bgp 100 neighbor 1.1.1.1`	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address as a BGP peer.
Step 4	**address-family** { **ipv4**	**ipv6** } **Example:** `RP/0/RP0/CPU0:router(config-bgp)# router bgp 100 neighbor 1.1.1.1 address-family ipv4`	Specifies either the IPv4 or IPv6 address family and enters address family configuration submode, and initializes the global address family.

	Command or Action	Purpose
Step 5	**flowspec validation** { **disable** | **redirect disable** } **Example:** `RP/0/RP0/CPU0:router(config-bgp)# router bgp 100 neighbor 1.1.1.1 address-family ipv4 flowspec validation disable`	You can choose to disable flowspec validation as a whole for all eBGP sessions or disable redirect nexthop validation.

Configuration Examples for Implementing BGP Flowspec

Flowspec Rule Configuration

Flowspec rule configuration example

In this example, two flowspec rules are created for two different VRFs with the goal that all packets to 10.0.1/24 from 192/8 and destination-port {range [137, 139] or 8080, rate limit to 500 bps in blue vrf and drop it in vrf-default. The goal is also to disable flowspec getting enabled on gig 0/0/0/0.

```
class-map type traffic match-all fs_tuple
 match destination-address ipv4 10.0.1.0/24
 match source-address ipv4 192.0.0.0/8
 match destination-port 137-139 8080
 end-class-map
 !
!
policy-map type pbr fs_table_blue
 class type traffic fs_tuple
  police rate 500 bps
  !
 !
 class class-default
 !
 end-policy-map
policy-map type pbr fs_table_default
 class type traffic fs_tuple
  drop
  !
```

```
 !
 class class-default
 !
 end-policy-map
flowspec
 local-install interface-all
 address-family ipv4
  service-policy type pbr fs_table_default
  !
 !
 vrf blue
  address-family ipv4
   service-policy type pbr fs_table_blue local
   !
  !
 !
!
Interface GigabitEthernet 0/0/0/0
 vrf blue
 ipv4 flowspec disable
```

Drop Packet Length

This example shows a drop packet length action configuration:

```
class-map type traffic match-all match-pkt-len
 match packet length 100-150
 end-class-map
!
policy-map type pbr test2
 class type traffic match-pkt-len
  drop
 !
 class type traffic class-default
 !
 end-policy-map
!
```

To configure a traffic class to discard packets belonging to a specific class, you use the drop command in policy-map class configuration mode. In this example, a multi-range packet length value from 100-150 has been defined. If the packet length of the incoming traffic matches this condition, the action is defined to 'drop' this packet.

Redirect traffic and rate-limit: Example

```
class-map type traffic match-all match-src-ipv6-addr
 match source-address ipv6 3110:1::/48
 end-class-map
!
policy-map type pbr test5
 class type traffic match-src-ipv6-addr
  redirect nexthop 3010:10:11::
  police rate 20 mbps
  !
 !
 class type traffic class-default
 !
 end-policy-map
!
```

In this example, an action is defined in the flowspec rule to redirect all the traffic from a particular source P address (3110:1::/48) to a next hop address. Also, for any traffic that comes with this source-address, rate limit the source address to 20 megabits per second.

Redirect Traffic from Global to VRF (vrf1)

This example shows you the configuration for redirecting traffic from a global traffic link to an individual VRF interface.

```
class-map type traffic match-all match-src-ipv6-addr
 match source-address ipv6 3110:1::/48
 end-class-map
!
policy-map type pbr test4
 class type traffic match-src-ipv6-addr
  redirect nexthop route-target 100:1
 !
 class type traffic class-default
 !
 end-policy-map
```

Remark DSCP

This is an example of the set dscp action configuration.

```
class-map type traffic match-all match-dscp-af11
 match dscp 10
 end-class-map
!
policy-map type pbr test6
 class type traffic match-dscp-af11
  set dscp af23
 !
 class type traffic class-default
 !
 end-policy-map
!
```

In this example, the traffic marking extended community (**match dscp**) instructs the system to modify or set the DSCP bits of a transiting IP packet from dscp 10 to dscp af23.

Additional References for BGP Flowspec

The following sections provide references related to implementing BGP Flowspec.

Related Documents

Related Topic	Document Title
BGP flowspec commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Routing Command Reference for Cisco CRS Routers*

Standards

Standards	Title
draft-ietf-idr-flow-spec-v6-05	*Dissemination of Flow Specification Rules for IPv6* ,
draft-ietf-idr-flowspec-redirect-ip-01	BGP Flow-Spec Redirect to IP Action
draft-simpson-idr-flowspec-redirect-02	BGP Flow-Spec Extended Community for Traffic Redirect to IP Next Hop
draft-ietf-idr-bgp-flowspec-oid-02	Revised Validation Procedure for BGP Flow Specifications

RFCs

RFCs	Title
RFC 5575	Dissemination of Flow Specification Rules

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport