Using NetFlow Filtering or Sampling to Select the Network Traffic to Track

First Published: June 19, 2006
Last Updated: December 17, 2010

This module contains information about and instructions for selecting the network traffic to track through the use of NetFlow filtering or sampling. The NetFlow Input Filtering and Random Sampled NetFlow features, described in this module, allow you to collect data from specific subsets of traffic.

- The NetFlow Input Filters feature provides NetFlow data for a specific subset of traffic by letting you create filters to select flows for NetFlow processing. For example, you can select flows from a specific group of hosts.
- The Random Sampled NetFlow feature provides NetFlow data for a subset of traffic in a Cisco router by processing only one randomly selected packet out of *n* sequential packets (*n* is a user-configurable parameter).

NetFlow is a Cisco IOS application that provides statistics on packets that flow through the router. It is emerging as a primary network accounting and security technology.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Using NetFlow Filtering or Sampling to Select Network Traffic to Track" section on page 21.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Contents

Prerequisites for Using NetFlow Filtering or Sampling to Select Network Traffic to Track

Prerequisites for NetFlow Input Filters

Before you can configure the NetFlow Input Filters feature, you must:

- Configure the router for IP routing.
- Configure Cisco Express Forwarding (CEF) switching or distributed Cisco Express Forwarding (dCEF) switching on the router and on the interfaces that you want to enable NetFlow Input Filters on (fast switching is not supported).
- Create traffic classes and define NetFlow sampler maps.

Note The NetFlow Input Filters feature is supported in the Version 5 and Version 9 NetFlow export formats.

Prerequisites for Random Sampled NetFlow

Before you can configure the Random Sampled NetFlow feature, you must:

- Configure the router for IP routing.
- Configure Cisco Express Forwarding (CEF) switching or distributed CEF (dCEF) switching on the router and on the interfaces that you want to enable Random Sampled NetFlow on (fast switching is not supported).
- Configure NetFlow Version 5 or Version 9 data export if you want to export NetFlow data (otherwise, NetFlow data is visible in the cache, but is not exported).
- Configure NetFlow Version 9 if you want to use sampler option templates or view NetFlow sampler IDs.

Restrictions for Using NetFlow Filtering or Sampling to Select Network Traffic to Track

Restrictions for NetFlow Input Filters

On Cisco 7500 platforms, the NetFlow Input Filters feature is supported only in distributed mode.

Restrictions for Random Sampled NetFlow

If full NetFlow is enabled on an interface, it takes precedence over Random Sampled NetFlow (which will thus have no effect). This means that you should disable full NetFlow on an interface before enabling Random Sampled NetFlow on that interface.

Enabling Random Sampled NetFlow on a physical interface does not automatically enable Random Sampled NetFlow on subinterfaces; you must explicitly configure it on subinterfaces. Also, disabling Random Sampled NetFlow on a physical interface (or a subinterface) does not enable full NetFlow. This restriction prevents the transition to full NetFlow from overwhelming the physical interface (or subinterface). If you want full NetFlow, you must explicitly enable it.

If you enable Random Sampled NetFlow with Version 5 data export, sampler option templates are not exported, and sampler IDs are exported in the least significant three bits of the last byte of the Version 5 record pad field. Use NetFlow Version 9 if you want to use sampler option templates or view NetFlow sampler IDs.

Information About Using NetFlow Filtering or Sampling to Select Network Traffic to Track

Roadmap: Using NetFlow Filtering or Sampling to Select the Network Traffic to Track

Table 1 provides a roadmap that includes links to associated information and configuration instruction for selecting traffic of interest.

Table 1 Roadmap: Selecting the Network Traffic to Track Using Sampling and Filtering

Traffic of Interest	Links to Associated Information and Configuration Instructions
A specific subset of NetFlow traffic for the purpose of class-based traffic analysis and monitoring (including on-network or off-network traffic)	Associated information: • Filtering and Sampling of NetFlow Traffic, page 4 • NetFlow Input Filters: Flow Classification, page 6 • Prerequisites for NetFlow Input Filters, page 2 • Restrictions for NetFlow Input Filters, page 3 Configuration instructions: • Configuring NetFlow Input Filters to Reduce the Impact of NetFlow Data Export, page 7
Statistical sampling of network traffic for traffic engineering or capacity planning purposes	Associated information: • Filtering and Sampling of NetFlow Traffic, page 4 • Random Sampled NetFlow: Sampling Mode, page 6 • Prerequisites for Random Sampled NetFlow, page 2 • Restrictions for Random Sampled NetFlow, page 3 Configuration instructions: • Configuring NetFlow Input Filters to Reduce the Impact of NetFlow Data Export, page 7

Filtering and Sampling of NetFlow Traffic

NetFlow provides highly granular per-flow traffic statistics in a Cisco router. A flow is a unidirectional stream of packets that arrive at the router on the same subinterface, have the same source and destination IP addresses, Layer 4 protocol, TCP/UDP source and destination ports, and the same type of service (ToS) byte in the IP headers. The router accumulates NetFlow statistics in a NetFlow cache and can export them to an external device (such as the Cisco Networking Services (CNS) NetFlow Collection Engine) for further processing.

Full NetFlow accounts for all traffic entering the subinterface on which it is enabled. But in some cases, you might gather NetFlow data on only a subset of this traffic. The Random Sampled NetFlow feature and the NetFlow Input Filters feature each provide ways to limit incoming traffic to only traffic of interest for NetFlow processing. Random Sampled NetFlow provides NetFlow data for a subset of traffic in a Cisco router by processing only one randomly selected packet out of *n* sequential packets. The NetFlow Input Filters feature provides the capability to gather NetFlow data on only a specific user-defined subset of traffic.

Note Random Sampled NetFlow is more statistically accurate than Sampled NetFlow. NetFlow's ability to sample packets was first provided by a feature named Sampled NetFlow. The methodology that the Sampled NetFlow feature uses is *deterministic* sampling, which selects every *n*th packet for NetFlow processing on a per-interface basis. For example, if you set the sampling rate to 1 out of 100 packets, then Sampled NetFlow samples the 1st, 101st, 201st, 301st, and so on packets. Sampled NetFlow does not allow random sampling and thus can make statistics inaccurate when traffic arrives in fixed patterns.

Note The Random Sampled NetFlow algorithms are applied after input filtering.

Table 2 compares the NetFlow Input Filters feature and the NetFlow Random Sampled feature.

Table 2 Comparison of the NetFlow Input Filters Feature and the Random Sampled NetFlow Feature

Comparison Category	NetFlow Input Filters Feature	Random Sampled NetFlow Feature
Brief description	This feature enables you to gather NetFlow data on only a specific subset of traffic. You do this by creating filters to select flows for NetFlow processing. For example, you can select flows from a specific group of hosts. This feature also lets you select various sampling rates for selected flows.	This feature provides NetFlow data for a subset of traffic in a Cisco router by processing only one randomly selected packet out of *n* sequential packets (*n* is a user-configurable parameter). Packets are sampled as they arrive (before any NetFlow cache entries are made for those packets).
Main uses	You can use this feature for class-based traffic analysis and monitoring on-network or off-network traffic.	You can use this feature for traffic engineering, capacity planning, and applications where full NetFlow is not needed for an accurate view of network traffic.
Export format support	This feature is supported in the Version 5 and Version 9 NetFlow export formats.	This feature is supported in the Version 5 and Version 9 NetFlow export formats.
Cisco IOS release support	12.3(4)T.	12.3(2)T, 12.2(18)S, and 12.0(26)S.
Subinterface support	You can configure NetFlow Input Filters per subinterface as well as per physical interface. You can select more than one filter per subinterface and have all of the filters run simultaneously.	You can configure the Random Sampled NetFlow feature per subinterface as well as per physical interface. Traffic is collected only on the subinterfaces on which Random Sampled NetFlow is configured. As with full NetFlow, enabling Random Sampled NetFlow on a physical interface does not enable Random Sampled NetFlow on subinterfaces automatically—you must explicitly configure it on the subinterfaces.
Memory impact	This feature requires no additional memory. It allows you to use a smaller NetFlow cache than full NetFlow, because it significantly reduces the number of flows. This feature requires an insignificant amount of memory for each configured NetFlow sampler.	This feature allows a smaller NetFlow cache than full NetFlow, because it significantly reduces the number of flows. This feature requires an insignificant amount of memory for each configured NetFlow sampler.
Performance impact	Accounting of classified traffic saves router resources by reducing the number of flows being processed and exported. The amount of bandwidth saved depends on the usage and the class-map criteria. However, performance might degrade depending on the number and complexity of class maps configured in a policy.	Statistical traffic sampling substantially reduces consumption of router resources (especially CPU resources) while providing valuable NetFlow data. This feature substantially reduces the impact of NetFlow data export on interface traffic. For example, a sampling rate of 1 out of 100 packets reduces the export of NetFlow data by about 50 percent.

NetFlow Input Filters: Flow Classification

For the NetFlow Input Filters feature, classification of packets can be based on any of the following: IP source and destination addresses, Layer 4 protocol and port numbers, incoming interface, MAC address, IP Precedence, DSCP value, Layer 2 information (such as Frame-Relay DE bits or Ethernet 802.1p bits), and Network-Based Application Recognition (NBAR) information. The packets are classified (filtered) on the above criteria, and flow accounting is applied to them on subinterfaces.

The filtering mechanism uses the Modular QoS Command-Line Interface (MQC) to classify flows. You can create multiple filters with matching samplers on a per-subinterface basis. For example, you can subdivide subinterface traffic into multiple classes based on type of service (ToS) values or destination prefixes (or both). For each class, you can also configure sampling at a different rate, using higher rates for higher-priority classes of traffic and lower rates for lower-priority ones.

MQC has many policies (actions) such as bandwidth rate and queuing management. These policies are applied only if a packet matches a criterion in a class map that is applied to the subinterface. A class map contains a set of match clauses and instructions on how to evaluate the clauses and acts as a filter for the policies, which are applied only if a packet's content satisfies the match clause. The NetFlow Input Filters feature adds NetFlow accounting to the MQC infrastructure, which means that flow accounting is done on a packet only if it satisfies the match clauses.

Two types of filter are available:

- ACL-based flow-mask filters
- Fields of filter (source IP address, destination IP address, source application port, destination application port, port protocol, ToS bits, and TCP flags)

Random Sampled NetFlow: Sampling Mode

Sampling mode makes use of an algorithm that selects a subset of traffic for NetFlow processing. In the random sampling mode that the Random Sampled NetFlow feature uses, incoming packets are randomly selected so that one out of each *n* sequential packets is selected *on average* for NetFlow processing. For example, if you set the sampling rate to 1 out of 100 packets, then NetFlow might sample the 5th packet and then the 120th, 199th, 302nd, and so on. This sample configuration provides NetFlow data on 1 percent of total traffic. The *n* value is a parameter from 1 to 65535 packets that you can configure.

Random Sampled NetFlow: The NetFlow Sampler

A NetFlow sampler map defines a set of properties (such as the sampling rate and NetFlow sampler name) for NetFlow sampling. Each NetFlow sampler map can be applied to one or many subinterfaces as well as physical interfaces. You can define up to eight NetFlow sampler maps.

For example, you can create a NetFlow sampler map named mysampler1 with the following properties: random sampling mode and a sampling rate of 1 out of 100 packets. This NetFlow sampler map can be applied to any number of subinterfaces, each of which would refer to mysampler1 to perform NetFlow sampling. Traffic from these subinterfaces is merged (from a sampling point of view). This introduces even more "randomness" than random per-subinterface NetFlow sampling does, but statistically it provides the same sampling rate of 1 out of 100 packets for each participating subinterface.

The sampling in random sampled NetFlow is done by NetFlow samplers. A NetFlow sampler is defined as an instance of a NetFlow sampler map that has been applied to a physical interface or subinterface. If full NetFlow is configured on a physical interface, it overrides random sampled NetFlow on all subinterfaces of this physical interface.

How to Configure NetFlow Filtering or Sampling

Note You need to configure input filtering before you apply the random sampled NetFlow algorithms.

Configuring NetFlow Input Filters to Reduce the Impact of NetFlow Data Export

Perform the following tasks to configure NetFlow input filters. Configuring NetFlow input filters reduces the impact of NetFlow data export.

- Creating a Class Map for a Policy Map for NetFlow Input Filtering, page 7 (required)
- Creating a Sampler Map for a Policy Map for NetFlow Input Filtering, page 9 (required)
- Creating a Class-Based Policy Containing NetFlow Sampling Actions, page 9 (required)
- Applying a Policy Containing NetFlow Sampling Actions to an Interface, page 11 (required)

Creating a Class Map for a Policy Map for NetFlow Input Filtering

Perform the following steps to create a class map for a policy map for NetFlow input filtering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all** | **match-any**]
4. **match access-group** *access-group*
5. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`enable`
	
	Example:
	`Router> enable`
		Enables privileged EXEC mode.
		• Enter your password if prompted.

Step 2	`configure terminal`
	
	Example:
	`Router# configure terminal`
		Enters global configuration mode.

	Command or Action	Purpose
Step 3	**class-map** *class-map-name* [**match-all** | **match-any**] **Example:** `Router(config)# class-map my_high_importance_class`	Creates a class map to be used for matching packets to a specified class. • The *class-map-name* argument is the name of the class for the class map. The name can be a maximum of 40 alphanumeric characters. The class name is used for both the class map and for configuring policy for the class in the policy map. • The **match-all | match-any** keywords determine how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (**match-all)** or only one of the match criteria (**match-any**) to be considered a member of the class. Entering the **class-map** command enables class-map configuration mode, in which you can enter one of the match commands to configure the match criteria for this class.
Step 4	**match access-group** *access-group* **Example:** `Router(config-cmap)# match access-group 101`	Configures the match criteria for a class map on the basis of the specified access control list (ACL). • The *access-group* argument is a numbered ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class. An ACL number can be a number from 1 to 2699.
Step 5	**end** **Example:** `Router(config-cmap)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Creating a Sampler Map for a Policy Map for NetFlow Input Filtering

Perform the following steps to create a sampler map for a policy map for NetFlow input filtering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow-sampler-map** *sampler-map-name*
4. **mode random** *one-out-of* **packet-interval**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**flow-sampler-map** *sampler-map-name* **Example:** `Router(config)# flow-sampler-map my_high_sampling`	Defines a statistical sampling NetFlow export flow sampler map. • The *sampler-map-name* argument is the name of the flow sampler map to be defined. Entering the **flow-sampler-map** command enables the flow sampler configuration mode.
Step 4	**mode random** *one-out-of* **packet-interval** **Example:** `Router(config-sampler-map)# mode random one-out-of 100`	Specifies a statistical sampling NetFlow export random sampling mode and a packet interval. • The **random** keyword specifies that sampling uses the random sampling mode. • The *one-out-of* **packet-interval** argument-keyword pair specifies the packet interval (one out of every *n* packets) from which to sample. For *n*, you can specify from 1 to 65535 (packets).
Step 5	**end** **Example:** `Router(config-sampler-map)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Creating a Class-Based Policy Containing NetFlow Sampling Actions

Perform thefollowing steps to create a class-based policy that contains NetFlow sampling actions.

You can assign only one NetFlow input filters sampler to a class. Assigning a subsequent NetFlow input filters sampler to a class overwrites the previous sampler. Removing a NetFlow sampler map also removes the NetFlow input filters sampler from the corresponding policy map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **netflow-sampler** *map-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**policy-map** *policy-map-name* **Example:** `Router(config)# policy-map mypolicymap`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. • The *policy-map-name* argument is the name of the policy map. The name can be a maximum of 40 alphanumeric characters. Entering the **policy-map** command enables quality of service (QoS) policy-map configuration mode, in which you can configure or modify the class policies for that policy map.
Step 4	**class** {*class-name* | **class-default**} **Example:** `Router(config-pmap)# class my_high_importance_class`	Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy. • The *class-name* argument is the name of the class for which you want to configure or modify policy. • The **class-default** keyword specifies the default class so that you can configure or modify its policy. Entering the **class** command enables QoS policy-map class configuration mode.

	Command or Action	Purpose
Step 5	**netflow-sampler** *sampler-map-name* **Example:** `Router(config-pmap-c)# netflow-sampler high_sampling`	Enables a NetFlow input filter sampler. • The *sampler-map-name* argument is the name of the NetFlow sampler map to apply to the class. You can assign only one NetFlow input filter sampler to a class. Assigning another NetFlow input filter sampler to a class overwrites the previous one.
Step 6	**end** **Example:** `Router(config-pmap-c)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Applying a Policy Containing NetFlow Sampling Actions to an Interface

Perform the following steps to apply a policy containing NetFlow sampling actions to an interface.

After you define a service policy with the **policy-map** command, you use the **service-policy** command in interface configuration mode to attach it to one or more interfaces, thus specifying the service policy for those interfaces. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached. You can apply the service policy only in the input direction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **service-policy** {**input** | **output**} *policy-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Router(config)# interface POS 1/0`	Specifies the interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**service-policy** {**input** | **output**} *policy-map-name* **Example:** `Router(config-if)# service-policy input mypolicymap`	Attaches a policy map to an input interface or virtual circuit (VC), or an output interface or VC, to be used as the service policy for that interface or VC. • The **input** keyword attaches the specified policy map to the input interface or input VC. • The **output** keyword attaches the specified policy map to the output interface or output VC. • The *policy-map-name* is the name of a service policy map (created through use of the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug flow-sampler class-based** command to display debugging output for NetFlow input filters.

Configuring Random Sampled NetFlow to Reduce the Impact of NetFlow Data Export

Perform the following tasks to configure and verify the configuration for the Random Sampled NetFlow feature:

- Defining a NetFlow Sampler Map, page 12 (required)
- Applying a NetFlow Sampler Map to an Interface, page 13 (required)
- Verifying the Configuration of Random Sampled NetFlow, page 14 (optional)

Defining a NetFlow Sampler Map

Perform the following task to define a NetFlow sampler map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow-sampler-map** *sampler-map-name*
4. **mode random one-out-of** *sampling-rate*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**flow-sampler-map** *sampler-map-name* **Example:** `Router(config)# flow-sampler-map mysampler1`	Defines a NetFlow sampler map and enters flow sampler map configuration mode. • The *sampler-map-name* argument is the name of the NetFlow sampler map to be defined.
Step 4	**mode random one-out-of** *sampling-rate* **Example:** `Router(config-sampler)# mode random one-out-of 100`	Enables random mode and specifies a sampling rate for the NetFlow sampler. • The **random** keyword specifies that sampling uses the random mode. • The **one-out-of** *sampling-rate* keyword-argument pair specifies the sampling rate (one out of every *n* packets) from which to sample. For *n*, you can specify from 1 to 65535 (packets).
Step 5	**end** **Example:** `Router(config-sampler)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Applying a NetFlow Sampler Map to an Interface

Perform the following task to apply a NetFlow sampler map to an interface.

You can apply a NetFlow sampler map to a physical interface (or a subinterface) to create a NetFlow sampler.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **flow-sampler** *sampler-map-name*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type interface-number* **Example:** `Router(config)# ethernet 1/0.2`	Specifies the interface and enters interface configuration mode.
Step 4	**flow-sampler** *sampler-map-name* **Example:** `Router(config-if)# flow-sampler mysampler1`	Applies a NetFlow sampler map to the interface to create the NetFlow sampler. • The *sampler-map-name* argument is the name of the NetFlow sampler map to apply to the interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits the current configuration mode and returns to privileged EXEC mode.

Verifying the Configuration of Random Sampled NetFlow

Perform the following tasks to verify the configuration of the Random Sampled NetFlow feature.

SUMMARY STEPS

1. **show flow-sampler**
2. **show ip cache verbose flow**
3. **show ip flow export template**

DETAILED STEPS

Step 1 **show flow-sampler**

Use this command to display attributes (including mode, sampling rate, and number of sampled packets) of one or all Random Sampled NetFlow samplers to verify the sampler configuration. For example:

```
Router# show flow-sampler

Sampler : mysampler1, id : 1, packets matched : 10, mode : random sampling mode
  sampling interval is : 100

 Sampler : myflowsampler2, id : 2, packets matched : 5, mode : random sampling mode
  sampling interval is : 200
```

To verify attributes for a particular NetFlow sampler, use the **show flow-sampler** *sampler-map-name* command. For example, enter the following for a NetFlow sampler named mysampler1:

```
Router# show flow-sampler mysampler1

Sampler : mysampler1, id : 1, packets matched : 0, mode : random sampling mode
  sampling interval is : 100
```

Step 2 show ip cache verbose flow

Use this command to display additional NetFlow fields in the header when Random Sampled NetFlow is configured. For example:

```
Router# show ip cache verbose flow
...
SrcIf          SrcIPaddress    DstIf          DstIPaddress    Pr TOS Flgs  Pkts
Port Msk AS                    Port Msk AS    NextHop              B/Pk  Active

BGP: BGP NextHop
Et1/0          8.8.8.8         Et0/0*         9.9.9.9         01 00  10       3
0000 /8  302                   0800 /8  300   3.3.3.3               100     0.1
BGP: 2.2.2.2         Sampler: 1  Class: 1  FFlags: 01
```

This example shows the NetFlow output of the **show ip cache verbose flow** command in which the sampler, class-id, and general flags are set. What is displayed for a flow depends on what flags are set in the flow. If the flow was captured by a sampler, the output shows the sampler ID. If the flow was marked by MQC, the display includes the class ID. If any general flags are set, the output includes the flags.

NetFlow flags (FFlags) that might appear in the **show ip cache verbose flow** command output are:

- FFlags: 01 (#define FLOW_FLAGS_OUTPUT 0x0001)—Egress flow
- FFlags: 02 (#define FLOW_FLAGS_DROP 0x0002)—Dropped flow (for example, dropped by an ACL)
- FFlags: 04 (#define FLOW_FLAGS_MPLS 0x0004)—MPLS flow
- FFlags: 08 (#define FLOW_FLAGS_IPV6 0x0008)—IPv6 flow
- FFlags: 10 (#define FLOW_FLAGS_RSVD 0x0010)—Reserved

IPv6 and RSVD FFlags are seldom used. If FFlags is zero, the line is omitted from the output. If multiple flags are defined (logical ORed together), then both sets of flags are displayed in hexadecimal format.

Step 3 show ip flow export template

Use this command to display the statistics for the NetFlow data export (such as template timeout and refresh rate) for the template-specific configurations. For example:

```
Router# show ip flow export template

Template Options Flag = 0
   Total number of Templates added = 0
   Total active Templates = 0
   Flow Templates active = 0
   Flow Templates added = 0
   Option Templates active = 0
   Option Templates added = 0
   Template ager polls = 0
   Option Template ager polls = 0
Main cache version 9 export is enabled
 Template export information
   Template timeout = 30
   Template refresh rate = 20
 Option export information
```

```
Option timeout = 30
Option refresh rate = 20
```

Troubleshooting Tips

Use the **debug flow-sampler** command to display debugging output for the Random Sampled NetFlow feature.

Configuration Examples for Configuring NetFlow Filtering and Sampling

Example: Configuring NetFlow Input Filters to Reduce the Impact of NetFlow Data Export:

Example: Creating a Class Map for a Policy Map for NetFlow Input Filtering

The following example shows how to create a class map for a policy map for NetFlow input filtering. In the example, class maps named my_high_importance_class and my_medium_importance_class are created.

```
configure terminal
!
class-map my_high_importance_class
 match access-group 101
 exit
!
class-map my_medium_importance_class
 match access-group 102
 end
```

Example: Creating a Sampler Map for a Policy Map for NetFlow Input Filtering

The following example shows how to create a sampler map for a policy map for NetFlow input filtering. In the following example, sampler maps called my_high_sampling, my_medium sampling, and my_low_samplng are created for use with a policy map for NetFlow input filtering.

```
configure terminal
```

```
!
flow-sampler-map my_high_sampling
 mode random one-out-of 1
 exit
!
flow-sampler-map my_medium_sampling
 mode random one-out-of 100
 exit
!
flow-sampler-map my_low_sampling
 mode random one-out-of 1000
 end
```

Example: Creating a Policy Containing NetFlow Sampling Actions

The following example shows how to create a class-based policy containing three NetFlow sampling actions. In this example, a sampling action named my_high_sampling is applied to a class named my_high_importance_class, a sampling action named my_medium_sampling is applied to a class named my_medium_importance_class, and a sampling action named my_low_sampling is applied to the default class.

```
configure terminal
!
policy-map mypolicymap
 class my_high_importance_class
 netflow sampler my_high_sampling
 exit
!
class my_medium_importance_class
 netflow-sampler my_medium_sampling
 exit
!
class class-default
 netflow-sampler my_low_sampling
end
```

Example: Applying a Policy to an Interface

The following example shows how to apply a policy containing NetFlow sampling actions to an interface. In this example, a policy named mypolicymap is attached to interface POS1/0 and also to interface ATM2/0.

```
configure terminal
!
interface POS1/0
 service-policy input mypolicymap
 exit
!
interface ATM2/0
 service-policy input mypolicymap
 end
```

Example: Configuring Random Sampled NetFlow to Reduce the Impact of NetFlow Data Export

Example: Defining a NetFlow Sampler Map

The following example shows how to define a NetFlow sampler map named mysampler1:

```
configure terminal
!
flow-sampler-map mysampler1
 mode random one-out-of 100
 end
```

Example: Applying a NetFlow Sampler Map to an Interface

The following example shows how to enable CEF switching and apply a NetFlow sampler map named mysampler1 to Ethernet interface 1 to create a NetFlow sampler on that interface:

```
configure terminal
!
ip cef
!
interface ethernet 1/0
 flow-sampler mysampler1
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
NetFlow commands	*Cisco IOS NetFlow Command Reference*
Overview of Cisco IOS NetFlow	*Cisco IOS NetFlow Overview*
List of the features documented in the *Cisco IOS NetFlow Configuration Guide*	*Cisco IOS NetFlow Features Roadmap*
The minimum information about and tasks required for configuring NetFlow and NetFlow Data Export	*Getting Started with Configuring NetFlow and NetFlow Data Export*
Tasks for configuring NetFlow to capture and export network traffic data	*Configuring NetFlow and NetFlow Data Export*
Tasks for configuring MPLS Aware NetFlow	*Configuring MPLS Aware NetFlow*
Tasks for configuring MPLS egress NetFlow accounting	*Configuring MPLS Egress NetFlow Accounting and Analysis*
Tasks for configuring Random Sampled NetFlow	*Using NetFlow Filtering or Sampling to Select the Network Traffic to Track*
Tasks for configuring NetFlow aggregation caches	*Configuring NetFlow Aggregation Caches*
Tasks for configuring NetFlow BGP next hop support	*Configuring NetFlow BGP Next Hop Support for Accounting and Analysis*
Tasks for configuring NetFlow multicast support	*"Configuring NetFlow Multicast Accounting"*

Related Topic	Document Title
Tasks for detecting and analyzing network threats with NetFlow	*Detecting and Analyzing Network Threats With NetFlow*
Tasks for configuring NetFlow Reliable Export With SCTP	*NetFlow Reliable Export With SCTP*
Tasks for configuring NetFlow Layer 2 and Security Monitoring Exports	*NetFlow Layer 2 and Security Monitoring Exports*
Tasks for configuring the SNMP NetFlow MIB	*Configuring SNMP and using the NetFlow MIB to Monitor NetFlow Data*
Tasks for configuring the NetFlow MIB and Top Talkers feature	*Configuring NetFlow Top Talkers using Cisco IOS CLI Commands or SNMP Commands*
Information for installing, starting, and configuring the CNS NetFlow Collection Engine	*Cisco CNS NetFlow Collection Engine Documentation*

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using NetFlow Filtering or Sampling to Select Network Traffic to Track

Table 3 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 3 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 3 Feature Information for Using NetFlow Filtering or Sampling to Select Network Traffic to Track

Feature Name	Releases	Feature Information
NetFlow Input Filters	12.3(4)T, 12.2(25)S 12.2(27)SBC 15.0(1)S	The NetFlow Input Filters feature provides NetFlow data for a specific subset of traffic by letting you create filters to select flows for NetFlow processing. For example, you can select flows from a specific group of hosts. This feature also lets you select various sampling rates for selected flows. The NetFlow Input Filters feature is used, for example, for class-based traffic analysis and monitoring on-network or off-network traffic. The following sections provide information about this feature: • Roadmap: Using NetFlow Filtering or Sampling to Select the Network Traffic to Track, page 3 • Filtering and Sampling of NetFlow Traffic, page 4 • NetFlow Input Filters: Flow Classification, page 6 • Configuring NetFlow Input Filters to Reduce the Impact of NetFlow Data Export, page 7 The following commands were introduced or modified by this feature: **netflow-sampler** and **debug flow-sampler**.

Table 3 Feature Information for Using NetFlow Filtering or Sampling to Select Network Traffic to Track (continued)

Feature Name	Releases	Feature Information
Random Sampled NetFlow	12.3(4)T, 12.2(18)S, 12.0(26)S, 12.2(27)SBC 12.2(33)SRC	Random Sampled NetFlow provides NetFlow data for a subset of traffic in a Cisco router by processing only one randomly selected packet out of *n* sequential packets (*n* is a user-configurable parameter). Packets are sampled as they arrive (before any NetFlow cache entries are made for those packets). Statistical traffic sampling substantially reduces consumption of router resources (especially CPU resources) while providing valuable NetFlow data. The main uses of Random Sampled NetFlow are traffic engineering, capacity planning, and applications where full NetFlow is not needed for an accurate view of network traffic. In Cisco IOS Release 12.2(33)SRC, this feature was enhanced to support IPv6 unicast and IPv4 multicast functionality. The following sections provide information about this feature: • Roadmap: Using NetFlow Filtering or Sampling to Select the Network Traffic to Track, page 3 • Filtering and Sampling of NetFlow Traffic, page 4 • Random Sampled NetFlow: Sampling Mode, page 6 • Random Sampled NetFlow: The NetFlow Sampler, page 6 • Configuring Random Sampled NetFlow to Reduce the Impact of NetFlow Data Export, page 12 The following commands were introduced by this feature: **debug flow-sampler**, **flow-sampler**, **flow-sampler-map**, **mode** (**flow sampler map configuration**), and **show flow-sampler**. The following command was modified by this feature: **ip flow-export**.

Glossary

ACL—Access control list. A roster of users and groups of users kept by a router. The list is used to control access to or from the router for a number of services.

BGP—Border Gateway Protocol. Interdomain routing protocol that replaces Exterior Gateway Protocol (EGP). A BGP system exchanges reachability information with other BGP systems. BGP is defined by RFC 1163.

BGP next hop—IP address of the next hop to be used to reach a certain destination.

CEF—Cisco Express Forwarding. Layer 3 IP switching technology that optimizes network performance and scalability for networks with large and dynamic traffic patterns.

dCEF—Distributed Cisco Express Forwarding. A type of CEF switching in which line cards (such as Versatile Interface Processor (VIP) line cards) maintain identical copies of the forwarding information base (FIB) and adjacency tables. The line cards perform the express forwarding between port adapters; this relieves the Route Switch Processor of involvement in the switching operation.

fast switching—Cisco feature in which a route cache is used to expedite packet switching through a router.

flow—Unidirectional stream of packets between a given source and destination. Source and destination are each defined by a network-layer IP address and transport-layer source and destination port numbers.

MQC—Modular QoS command-line interface. A CLI structure that lets you create traffic polices and attach them to interfaces. A traffic policy contains a traffic class and one or more QoS features. The QoS features in the traffic policy determine how the classified traffic is treated.

NBAR—Network-Based Application Recognition. A classification engine in Cisco IOS software that recognizes a wide variety of applications, including web-based applications and client/server applications that dynamically assign Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port numbers. After the application is recognized, the network can invoke specific services for that application. NBAR is a key part of the Cisco Content Networking architecture and works with QoS features to let you use network bandwidth efficiently.

NetFlow—Cisco IOS security and accounting feature that maintains per-flow information.

NetFlow sampler—A set of properties that are defined in a NetFlow sampler map that has been applied to at least one physical interface or subinterface.

NetFlow sampler map—The definition of a set of properties (such as the sampling rate) for NetFlow sampling.

NetFlow v9—NetFlow export format Version 9. A flexible and extensible means for carrying NetFlow records from a network node to a collector. NetFlow Version 9 has definable record types and is self-describing for easier NetFlow Collection Engine configuration.

ToS—type of service. Second byte in the IP header that indicates the desired quality of service for a specific datagram.