NetFlow and sFlow Configuration Guide on Cisco 8000 Series Routers, Cisco IOS XR Releases

Cisco 8000 Series Routers
Updated June 9, 2026

1 NetFlow and sFlow documentation for Cisco 8000 and ASR 9000 series routers

Topics:

Provides a release-agnostic reference for NetFlow and sFlow features on Cisco 8000 and ASR 9000 series routers, updated with the latest information across IOS XR releases.

Provides a release-agnostic reference for NetFlow and sFlow features on Cisco 8000 and ASR 9000 series routers, updated with the latest information across IOS XR releases.

The NetFlow and sFlow documentation for Cisco 8000 and ASR 9000 series routers is maintained as a single, release-agnostic reference. This approach ensures that users always have access to the most current NetFlow and sFlow features and configuration details, regardless of the IOS XR release.

.
.

To begin with NetFlow and sFlow implementation, refer to the Cisco IOS XR Feature, Release, and Platform Matrix. This resource provides insights into NetFlow and sFlow features introduced for the and Series Routers, as well as other IOS XR routing platforms. The goal is to simplify access to documentation and support a comprehensive understanding of NetFlow and sFlow capabilities.

This document for the and Series Routers is designed as a single version that is consistently updated with the latest features and releases. Users can bookmark this reference to access comprehensive NetFlow and sFlow configuration information for the and Series Routers, eliminating the need to consult multiple release-specific documents.

Supported IOS XR releases

Lists the Cisco IOS XR software releases supported for Cisco 8000, ASR 9000, and NCS 5500 platforms, with links to release notes and end-of-life information.

Lists the Cisco IOS XR software releases supported for Cisco 8000, ASR 9000, and NCS 5500 platforms, with links to release notes and end-of-life information.

This section lists the supported Cisco IOS XR software releases for Cisco 8000 Series platforms.

- IOS XR Release 7.11.1
- IOS XR Release 7.10.1
- IOS XR Release 7.9.2
- IOS XR Release 7.9.1
- IOS XR Release 7.8.2
- IOS XR Release 7.8.1
- IOS XR Release 7.7.2
- IOS XR Release 7.7.1
- IOS XR Release 7.5.3
- IOS XR Release 7.5.2
- IOS XR Release 7.5.1
- IOS XR Release 7.3.4
- IOS XR Release 7.3.3
- IOS XR Release 7.3.2
- IOS XR Release 7.3.15
- IOS XR Release 7.3.1

This section lists the supported Cisco IOS XR software releases for ASR 9000 Series platforms.

- IOS XR Release 7.11.1
- IOS XR Release 7.10.1
- IOS XR Release 7.9.2
- IOS XR Release 7.9.1
- IOS XR Release 7.8.2
- IOS XR Release 7.8.1
- IOS XR Release 7.7.2
- IOS XR Release 7.7.1
- IOS XR Release 7.6.2
- IOS XR Release 7.6.1
- IOS XR Release 7.5.3

- IOS XR Release 7.5.2
- IOS XR Release 7.5.1
- IOS XR Release 7.4.2
- IOS XR Release 7.4.1
- IOS XR Release 7.3.4
- IOS XR Release 7.3.2
- IOS XR Release 7.3.1

This section lists the supported Cisco IOS XR software releases for NCS 5500 Series platforms.

- IOS XR Release 7.11.1
- IOS XR Release 7.10.1
- IOS XR Release 7.9.2
- IOS XR Release 7.9.1
- IOS XR Release 7.8.2
- IOS XR Release 7.8.1
- IOS XR Release 7.7.21
- IOS XR Release 7.7.2
- IOS XR Release 7.7.1
- IOS XR Release 7.6.2
- IOS XR Release 7.6.1
- IOS XR Release 7.5.3
- IOS XR Release 7.5.2
- IOS XR Release 7.5.1
- IOS XR Release 7.4.2
- IOS XR Release 7.4.1
- IOS XR Release 7.3.4
- IOS XR Release 7.3.2
- IOS XR Release 7.3.1

Cisco IOS XR releases prior to those listed have reached the End of Extended Software Maintenance. For more information, visit the End-of-Life and End-of-Sale Notices page.

2 Features introduced in Cisco IOS XR products and releases

Topics:

Lists the features introduced and enhanced for NetFlow and sFlow, and maps their availability across Cisco IOS XR platforms and releases.

Lists the features introduced and enhanced for NetFlow and sFlow, and maps their availability across Cisco IOS XR platforms and releases.

This reference provides a summary of features introduced and enhanced for NetFlow and sFlow, and helps map their availability across Cisco IOS XR platforms and releases.

Feature, release, and platform matrix for NetFlow and sFlow

Lists the feature, release, and platform availability for NetFlow and sFlow across Cisco 8000, ASR 9000, and NCS 5500 Series platforms.

Objective and scope.

Use this matrix to locate features of interest and map their availability across Cisco 8000, ASR 9000, and NCS 5500 Series platforms and releases.

3 NetFlow and sFlow protocols

Topics:

Explains the basics of NetFlow and sFlow protocols, their key components, and the benefits of deploying these protocols for network monitoring.

A NetFlow and sFlow protocol is a network monitoring technology that

- provides insights into network traffic and performance,
- enables organizations to understand traffic patterns and detect anomalies, and
- optimizes network performance by collecting and analyzing traffic data.

Key attributes and protocol details for NetFlow and sFlow

Describes the NetFlow and sFlow protocols, their network monitoring capabilities, and the benefits of deploying these protocols for traffic analysis and performance optimization.

NetFlow, developed by Cisco, is a protocol that collects and analyzes network traffic data, allowing organizations to understand traffic patterns, detect anomalies, and optimize network performance.

sFlow is a vendor-neutral protocol for monitoring network traffic. It samples packets at the interface level and provides a broad view of network activity, including detailed information on the types of traffic and the devices generating it.

Lists the feature history and supported platforms for NetFlow and sFlow protocols.

NetFlow and sFlow benefits

Explains the advantages of using NetFlow and sFlow for network monitoring, planning, security, billing, and traffic engineering.

A NetFlow and sFlow benefit is a network feature that

- provides comprehensive insights into network traffic and application usage,
- enables effective network planning, security analysis, billing, and traffic engineering, and
- supports proactive troubleshooting and resource optimization.

Monitoring network applications and use

Network planning

Security analysis

Billing and accounting

Traffic engineering

The data collected through NetFlow or sFlow enables you to view comprehensive, time- and application-based insights into network usage. This data serves as a foundation for strategic network and application resource allocation, offering robust near real-time monitoring capabilities. It can effectively display traffic trends and views based on applications. Additionally, it facilitates proactive identification of issues, streamlined troubleshooting, and swift problem resolution. This information proves invaluable in optimally allocating network resources, as well as identifying and addressing potential security breaches and policy violations.

NetFlow or sFlow can be effectively used to capture data over extended durations, empowering users to monitor and predict network expansion, and plan enhancements such as increased routing devices, ports, or higher-bandwidth interfaces. The data serves as a cornerstone for fine-tuning network planning, including aspects such as peering, backbone upgrades, and routing policy decisions. This approach minimizes overall network operational costs while maximizing performance, capacity, and reliability. The data aids in identifying unwanted WAN traffic, validating bandwidth and Quality of Service (QoS), and facilitating analysis of novel network applications. This wealth of information ultimately contributes to reducing the network operation costs.

NetFlow or sFlow data plays a pivotal role in promptly detecting and categorizing real-time Denial of Service (DoS) attacks, viruses, and worms. Changes in network patterns reveal anomalies that are distinctly highlighted in the NetFlow data. Furthermore, this data is an invaluable resource for network forensic analysis, enabling a comprehensive understanding and reconstruction of security incidents.

NetFlow and sFlow provide insights into the utilization of resources across a network, facilitating detailed accounting reports depicting resource usage across diverse network components.

NetFlow and sFlow can gauge the volume of traffic traversing peering or transit points, and assess whether a peering agreement with other service providers is fair and equitable.

Key components of NetFlow and sFlow

Lists the main NetFlow and sFlow components used to capture, export, collect, analyze, and manage data.

Lists the main NetFlow and sFlow components used to capture, export, collect, analyze, and manage data.

The following NetFlow and sFlow components help you capture, export, collect, analyze, and manage data:

- Exporter: Generates flow records and sends them to a collector.
- Collector: Receives flow records from exporters and stores them for analysis.
- Analyzer: Processes and interprets collected flow data to provide insights and reports.
- Agent: Embedded in network devices to monitor and sample traffic (primarily for sFlow).
- Manager: Provides configuration and management functions for exporters, collectors, and analyzers.

Flow exporter details

Provides information about flow exporters, their functions, and configuration limits.

Flow exporters collect data about network flows and send flow records to a designated collector. This section provides details about their functions and configuration limits.

Flow exporters are device components that collect and export network flow data. The following list describes their main attributes:

- A flow exporter, also known as an exporter map, collects data about network flows and sends flow records to a designated collector.
- The exporter inspects packets, identifies flows, and exports flow-related data. Each exporter map can transmit flow reports to a single destination, and up to eight exporters are allowed per MAP configuration.
- The flow exporter contains network specifications and transport layer attributes for the packets it exports. These packets are sent to the collector using the User Datagram Protocol (UDP). If the source interface does not have an assigned IP address, the packet exporter remains inactive.

Flow monitor

Provides details about flow monitors, their components, and how they facilitate active traffic monitoring on interfaces.

Flow monitors facilitate active traffic monitoring on interfaces by collecting and exporting traffic data using flow record and exporter maps.

A flow monitor is a network monitoring component that:

- facilitates active traffic monitoring on a pre-configured interface,
- generates a flow monitor cache to collect traffic data based on key and non-key fields in the configured record, and
- links to flow record maps and flow exporter maps for comprehensive data collection and export.

You can include extended details in a flow monitor, such as router-specific elements like nexthop, source and destination mask lengths, and extended gateway attributes, including nexthop, communities, local preference, and AS (source AS, source peer AS, and destination AS path) information.

A monitor map contains name references that link to the flow record map and flow exporter map, both of which are committed to an interface. If an exporter map is not applied to the monitor map, the flow records are not exported. In such cases, the aging process adheres to the cache parameters specified in the monitor map.

Flow sampler configuration parameters

Lists the configuration parameters and considerations for flow sampler maps, including supported sampling rates and usage guidelines.

This reference lists the configuration parameters and important considerations for flow sampler maps, including supported sampling rates and usage guidelines for high-speed interfaces.

The sampler map specifies the rate at which packets (one out of n packets) are sampled. The sampler map configuration is typically used for high-speed interfaces to optimize CPU utilization. Set the sampling rate after evaluating network parameters such as traffic rate, number of total flows, cache size, and active and inactive timers.

- The maximum supported sampling rate is 1:1, where every packet is processed.

Consider these points before applying the sampler map:

- Remove any existing Netflow or sFlow configurations before applying a new sampler map on an interface.
- Use the same sampler map configuration on the sub-interfaces and physical interfaces under a port.

4 Monitoring protocols

Topics:

Explains the key differences between NetFlow and sFlow protocols for network traffic monitoring.

A monitoring protocol is a network protocol that

- collects and analyzes data about network traffic flows
- enables administrators to monitor and troubleshoot network performance, and
- supports informed decisions about network management and optimization.

Key differences between NetFlow and sFlow protocols

Describes the characteristics and use cases of NetFlow and sFlow protocols for network traffic monitoring.

NetFlow and sFlow are two widely used monitoring protocols. NetFlow provides detailed flow-based statistics by tracking every flow, while sFlow uses statistical sampling to provide scalable monitoring with lower resource usage. Choosing between NetFlow and sFlow depends on your network's requirements for detail, scalability, and resource consumption.

Examples of monitoring protocol use cases:

- NetFlow is suitable for environments that require granular traffic analysis and security monitoring.
- sFlow is preferred in large-scale networks where scalability and minimal overhead are priorities.

Contrast between NetFlow and sFlow:

Table 1: NetFlow vs. sFlow contrast table

Feature	NetFlow	sFlow
Data collection method	Tracks every flow	Uses statistical sampling
Resource usage	Higher	Lower
Level of detail	Granular	Summary
Scalability	Limited by resource consumption	Highly scalable

Monitoring protocol examples

Examples can include NetFlow for granular traffic analysis and security monitoring, and sFlow for scalable monitoring in large networks with minimal overhead.

Differences between sFlow and NetFlow

Lists the key differences between sFlow and NetFlow based on functionality, monitored flows, processing, scalability, mechanism, and resource usage.

Provides a comparison of sFlow and NetFlow based on functionality, monitored flows, processing, scalability, mechanism, and resource usage.

This reference provides a comparison of NetFlow and sFlow based on several factors.

- NetFlow stores and aggregates IP traffic metadata as flow records, which are exported to a collector after the active timer expires.
- sFlow transfers sampled data promptly to the collector without storing it, resulting in lower resource usage and a stateless mechanism.

Table 2: Differences between sFlow and Netflow

Factor	Netflow	sFlow
Functionality	Stores the exported metadata aggregated related to IP traffic in the form of a Netflow Record Template. This Netflow Record is saved until the active timer expires, at which point it's exported to the collector.	sFlow is transient and therefore the data is not stored, instead, it's promptly transferred to the collector.
Monitored Flows	Monitors the following traffic: • Layer 2 • Layer 3: MPLS, IPv4, IPv6, SRv6	Monitors the following traffic: • Layer 2 • Layer 3: MPLS, IPv4, IPv6, SRv6 • VLANs
Flow Data Processing	Processes flow data and stores it as flow records.	The flow data is sent to the collector for analysis. Unlike NetFlow, sFlow is not cached and contains fewer flow data fields.
Scalability	Scalability is determined by the number of packets sampled as per the NetFlow sampler map.	Scalability is determined by the number of sFlow packets.
Mechanism	Packet aggregation into flows. Netflow is stateful as it stores the Flow record until the active timer expires.	Packets are sampled randomly, while counters are sampled based on time intervals. sFlow is not stateful since the data is not stored; it's directly forwarded to the collector.
Resource usage	Elevated, due to more information provided, resulting in a significant burden on the router.	Typically places lower load on the router.

Comparison of NetFlow and sFlow for network traffic monitoring

Provides a comparison of NetFlow and sFlow protocols to help select the best option for network traffic monitoring based on device type and network performance factors.

This reference lists the key differences between NetFlow and sFlow protocols to help select the best option for network traffic monitoring based on device type and network performance factors.

When selecting a protocol to monitor network traffic, consider the types of devices in your network and key performance factors such as latency and scalability.

- Choose NetFlow to gather extensive data and enhance visibility. NetFlow provides comprehensive insights into traffic flows, which helps address network issues, identify security risks, plan network upgrades, and optimize bandwidth utilization. However, NetFlow may introduce slightly higher latency compared to sFlow.
- Choose sFlow to monitor traffic in networks with limited bandwidth. sFlow imposes less load on network and computing resources than NetFlow, making it suitable for small and medium-sized businesses and networks with less powerful devices. sFlow uses a sampling approach to collect data from a subset of packets, enabling analysis of traffic patterns, identification of irregularities, and optimization of network performance with lower resource demands.

If your network supports both NetFlow and sFlow, you can use both technologies to leverage their respective strengths. By identifying specific use cases, you can select the protocol that best meets your network's traffic monitoring requirements.

Table 3: Comparison of NetFlow and sFlow protocols

Attribute	NetFlow	sFlow	Notes
Data collection method	Flow-based (collects data on all flows)	Packet sampling (collects data from a subset of packets)	NetFlow provides more granular data; sFlow reduces resource usage.
Resource usage	Higher (may increase CPU and memory usage)	Lower (minimal impact on device resources)	sFlow is suitable for devices with limited resources.
Visibility	Comprehensive (detailed traffic analysis)	Statistical (traffic patterns and trends)	NetFlow is preferred for deep analysis; sFlow for overall trends.
Latency impact	May introduce slightly higher latency	Minimal latency impact	sFlow is better for latency-sensitive environments.
Best use case	Large enterprise networks, security monitoring, capacity planning	Small/medium networks, bandwidth-constrained environments	Choose based on network size and monitoring needs.

Egress NetFlow, sFlow, and IPFIX support matrix

Lists egress NetFlow, sFlow, and IPFIX support by platform category, interface type, marketing PID, non-ETM support, and ETM support.

This reference provides support details for egress NetFlow, sFlow, and IPFIX across fixed, modular, and centralized platforms.

Describes the supported platforms, interface types, hardware, and ETM/non-ETM support for egress NetFlow, sFlow, and IPFIX.

- Fixed platforms: Lists support for A100, K100, P100, and P200 with L2 and L3 interface types.
- Modular platforms: Lists support for K100, P100, and P200 with L2 and L3 interface types.
- Centralized platforms: Lists support for K100 with L2 and L3 interface types.

Table 4: Fixed platform support

Platform	Interface Type	Hardware	Non-ETM Support	ETM Support
A100	L3	• 8011-4G24Y4H-I • 8011-32Y8L2H2FH • 8011-24Y8L2FH-I • 8011-24X • 8011-12G12X4Y • 8011-2X2XP4L	Fake Egress sFlow/IPFIX315	Not Supported
A100	L2	• 8011-4G24Y4H-I • 8011-32Y8L2H2FH • 8011-24Y8L2FH-I • 8011-24X • 8011-12G12X4Y • 8011-2X2XP4L	Not Supported	Not Supported
K100	L3	• 8711-48Z-M • 8711-28H8F-M	Fake Egress sFlow/IPFIX315	Real Egress Netflow/sFlow/IPFIX
K100	L2	• 8711-48Z-M • 8711-28H8F-M	Not Supported	Not Supported
P100	L3	• 8212-48FH-M • 8711-32FH-M	Fake Egress sFlow/IPFIX315	Not Supported
P100	L2	• 8212-48FH-M • 8711-32FH-M	Not Supported	Not Supported
P200	L3	• 8223-64E-M • 8223-64E-MO • 8223-64EF-M • 8223-64EF-MO	Not Supported	Not Supported
P200	L2	• 8223-64E-M • 8223-64E-MO • 8223-64EF-M • 8223-64EF-MO	Not Supported	Not Supported

Table 5: Modular platform support

Platform	Interface Type	Hardware	Non-ETM Support	ETM Support
K100	L3	• 88-LC1-48Y8H-EM • 88-LC1-16H16F-EM	-	Real Egress Netflow/sFlow/IPFIX
K100	L2	• 88-LC1-48Y8H-EM • 88-LC1-16H16F-EM	-	Not Supported
P100	L3	• 88-LC1-36EH • 88-LC1-36EH-O • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM	Not Supported	Real Egress Netflow/sFlow/IPFIX
P100	L2	• 88-LC1-36EH • 88-LC1-36EH-O • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM	Not Supported	Not Supported
P200	L3	• 88-LC2-36EF-M	Not Supported	-
P200	L2	• 88-LC2-36EF-M	Not Supported	-

Table 6: Centralized platform support

Platform	Interface Type	Hardware	Non-ETM Support	ETM Support
K100	L3	• 8404-SYS-D • 8404-RSP1-48-EM	Fake Egress sFlow/IPFIX315	Real Egress Netflow/sFlow/IPFIX
K100	L2	• 8404-SYS-D • 8404-RSP1-48-EM	Not Supported	Not Supported

5 NetFlow protocol

Topics:

Explains the NetFlow protocol, its versions, key concepts, and how to configure NetFlow for traffic monitoring and analysis.

A NetFlow protocol is a network traffic monitoring technology that

- collects and analyzes flow data from network devices,
- supports multiple versions including NetFlow v9 and IPFIX, and
- enables administrators to monitor, troubleshoot, and optimize network performance.

NetFlow protocol concepts and configuration guidance

Describes the fundamental principles, variations, benefits, and limitations associated with NetFlow protocol. Provides guidance on configuring NetFlow version 9 or IPFIX for network traffic monitoring and analysis.

NetFlow components

Explains the main components and terms used in NetFlow for network traffic analysis.

A NetFlow component is a network feature that

- collects, processes, and exports flow data from network devices
- enables monitoring and analysis of IP or MPLS packet flows, and
- facilitates storage and examination of flow records for network insight

- Data source: Specific locations within the router, such as physical interfaces and VLANs, where traffic measurements can be taken.
- Flow: A collection of IP or MPLS packets traversing the router during a time period. All packets belonging to a particular flow share common attributes derived from the packet's data.
- Flow record: A set of key and non-key NetFlow field values used to characterize flows in the NetFlow cache. It is generated by examining packet headers and adding a description of packet details in the NetFlow cache.
- Exporter: A component within the router that has NetFlow enabled. An exporter monitors incoming packets, generates flows from them, and transmits information derived from these flows, encapsulated as flow records, to the NetFlow collector.
- Collector: An external device designed to receive flow records from one or multiple exporters. The collector processes the incoming export packets and stores the associated flow record details. Optionally, flow records can undergo aggregation before being stored on the hard disk.
- NetFlow cache: A segment of memory that stores flow entries prior to their exportation to an external collector. This includes two cache types: the normal cache and the permanent cache.
- NetFlow analyser: An external device or application responsible for collecting and scrutinizing flow records to provide valuable insights.
- Collector address: The IP address and UDP port number used by the collector. By default, the designated destination port number is 2055.

NetFlow component attributes and terms

Describes the main attributes and terms associated with NetFlow components for network traffic analysis.

- Data source: Specific locations within the router, such as physical interfaces and VLANs, where traffic measurements can be taken.
- Flow: A collection of IP or MPLS packets traversing the router during a time period. All packets belonging to a particular flow share common attributes derived from the packet's data.
- Flow record: A set of key and non-key NetFlow field values used to characterize flows in the NetFlow cache. It is generated by examining packet headers and adding a description of packet details in the NetFlow cache.
- Exporter: A component within the router that has NetFlow enabled. An exporter monitors incoming packets, generates flows from them, and transmits information derived from these flows, encapsulated as flow records, to the NetFlow collector.
- Collector: An external device designed to receive flow records from one or multiple exporters. The collector processes the incoming export packets and stores the associated flow record details. Optionally, flow records can undergo aggregation before being stored on the hard disk.
- NetFlow cache: A segment of memory that stores flow entries prior to their exportation to an external collector. This includes two cache types: the normal cache and the permanent cache.

- NetFlow analyser: An external device or application responsible for collecting and scrutinizing flow records to provide valuable insights.
- Collector address: The IP address and UDP port number used by the collector. By default, the designated destination port number is 2055.

NetFlow protocol

Explains how the NetFlow protocol enables network monitoring by logging metadata for each flow traversing a router.

A NetFlow protocol is a network protocol that

- logs metadata for each flow entering or leaving a router,
- provides comprehensive insights into network flows, including source and destination IP addresses, ports, and packet counts, and
- supports traffic analysis, capacity planning, and network troubleshooting.

Cross AFI BGP NH information elements

Cross AFI BGP NH information elements are data fields that specify the next hop IP address for different network layer protocols in BGP routing. These elements ensure proper routing across diverse network environments by indicating the appropriate next hop based on the Address Family Identifier (AFI) and its Subsequent Address Family Identifier (SAFI).

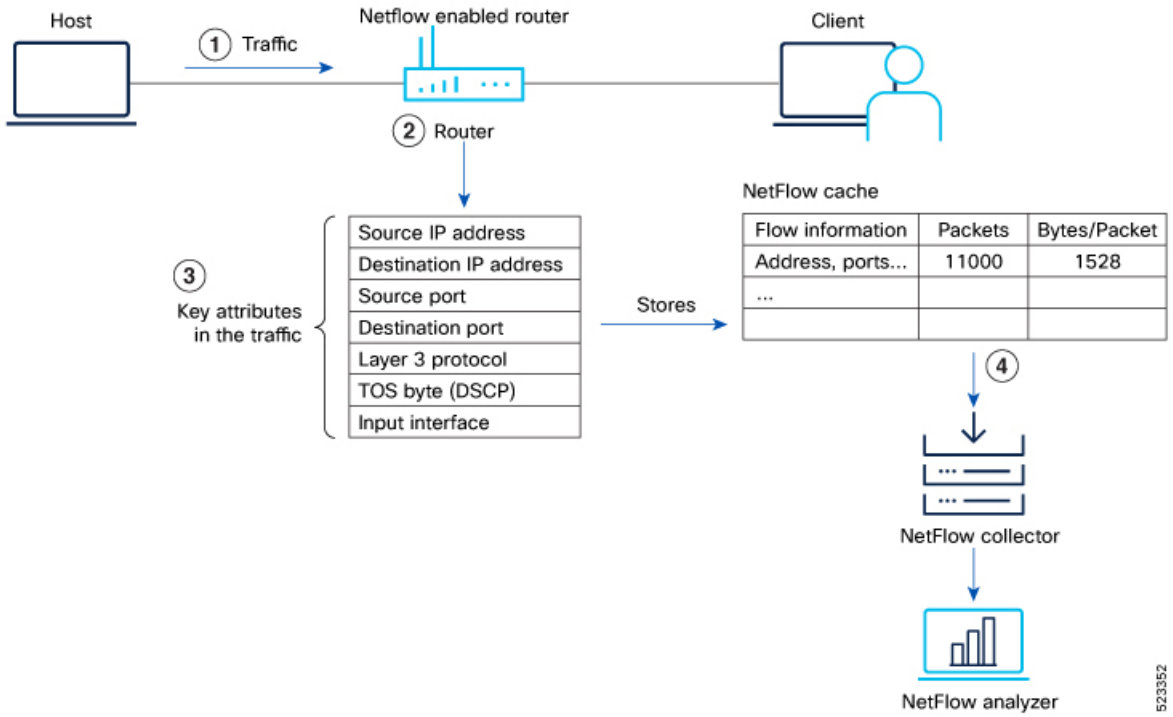
Table 7: Feature History Table

Recording of Packet Flows in NetFlow

Recording of Packet Flows in NetFlow explains the key information for this topic.

The packet in NetFlow is recorded as follows:

Figure 1: Packet Flows in NetFlow



rect 125, 19, 162, 53 Flow Creation
rect 248, 72, 281, 105 Datagram Generation
rect 107, 161, 145, 200 Data Export
rect 618, 219, 652, 254 Analysis and Reporting

In NetFlow, the focus is on recording and collecting full packet flows in the network traffic data. When NetFlow is configured on the router, the router collects flow data by extracting key field attributes from the packet streams, and generates a flow record. This record, along with accounting information, is stored in the database or NetFlow Cache. The extracted records, once sampled, are exported to one or more NetFlow collectors via the UDP transport layer protocol. This exported data has several purpose: enterprise accounting and ISP billing, and so on.

Here's how NetFlow handles the recording of packet flows:

1. Flow Creation: NetFlow creates flow records by monitoring network traffic passing through the router. As a packet stream traverses a router interface, the packets are collected and an internal header is appended. These packets are dispatched to the line card's CPU, which generate a flow record. The router extracts pertinent header details from the packets and creates cache entries. The packets are subject to a policer, which helps protect the internal control plane. With each subsequent arrival of a packet from the same flow, the cache entry is updated. Flow records persist within the line card's cache until they age out due to timer expiration.

   When the expiry of the set timer occurs, the NetFlow is generated. There are timers (two of them) running for flow aging.

   - The active timer signifies the maximum allowable duration for a particular cache entry's existence, even if matched by received sampled packets.
   - The inactive timer represents the duration without receipt of a sampled packet corresponding to a specific cache entry.

2. Datagram Generation: The NetFlow agent generates NetFlow datagrams that contain information about the packets. These datagrams include details such as source and destination IP addresses, port numbers, protocol information, and various flow statistics.

3. Data Export: The NetFlow datagrams are periodically exported from the NetFlow agent to a designated NetFlow collector or analyzer. The export can be done using protocols like UDP or TCP, and the datagrams are typically sent in a structured format like IPFIX or JSON.

A flow record is sent to the NetFlow collector in the following scenarios:

- The flow has been inactive or active for an extended period.
- The user triggers the export of the flow.
- The flow concludes, which is particularly relevant when TCP connections are terminated.

4. Analysis and Reporting: Upon receiving the NetFlow data, the NetFlow collector or analyzer processes and analyzes the information. It aggregates the sampled data to provide statistical insights into network traffic, including top talkers, protocol distribution, traffic patterns, and other metrics.

Key attributes in IP and MPLS packets for NetFlow

Lists the key attributes in IP and MPLS packets that NetFlow monitors.

NetFlow monitors key attributes in IP and MPLS packets to provide detailed visibility into network traffic for analysis and troubleshooting.

NetFlow supports IPv4, IPv6, MPLS, BGP, SRv6, and GTP-U flow types, enabling monitoring of a wide range of packet information. The key attributes in IP and MPLS packets that NetFlow monitors include:

- Source and destination IP addresses
- Source and destination port numbers
- Layer 3 protocol type
- Type of service (ToS) byte
- Input and output interface numbers
- MPLS labels
- BGP next hop
- SRv6 segment identifiers
- GTP-U tunnel identifiers

These attributes allow NetFlow to provide detailed visibility into network traffic for analysis and troubleshooting.

IP traffic monitors

Explains how NetFlow monitors and collects key attributes of IP traffic for both IPv4 and IPv6 networks.

A IP traffic monitor is a network analysis tool that

- collects detailed traffic data for both IPv4 and IPv6 networks
- captures key attributes such as source and destination addresses, protocol types, and bandwidth usage, and
- enables identification of network trends, detection of security threats, and optimization of network performance

Key IP traffic attributes monitored include:

- Source and destination IP addresses
- Source and destination MAC addresses
- Source and destination ports for TCP and UDP
- Differentiated Services Code Point (DSCP)

- Layer 3 protocol
- Type of Service (ToS) byte
- Traffic receiving interface
- Complete IPv4 header fields, including IP-ID and TTL
- Counts for packets and bytes
- Full spectrum of IPv6 header fields, including flow label and option header
- Flow timestamps

Attributes and functions of IP traffic monitors

Describes the main functions and monitored attributes of NetFlow IP traffic monitors for IPv4 and IPv6 networks.

NetFlow IP traffic monitors provide the following capabilities:

- Collect detailed traffic data for both IPv4 and IPv6 networks
- Capture key attributes such as source and destination addresses, protocol types, and bandwidth usage
- Enable identification of network trends, detection of security threats, and optimization of network performance

Key IP traffic attributes monitored include:

- Source and destination IP addresses
- Source and destination MAC addresses
- Source and destination ports for TCP and UDP
- Differentiated Services Code Point (DSCP)
- Layer 3 protocol
- Type of Service (ToS) byte
- Traffic receiving interface
- Complete IPv4 header fields, including IP-ID and TTL
- Counts for packets and bytes
- Full spectrum of IPv6 header fields, including flow label and option header
- Flow timestamps

MPLS traffic monitoring

Explains how NetFlow provides visibility into MPLS traffic by monitoring key attributes such as MPLS labels.

A MPLS traffic monitoring solution is a network visibility tool that

- uses NetFlow to collect detailed traffic data for MPLS traffic
- enables identification of anomalies and detection of cyber threats, and
- supports rapid response to potential security incidents.

MPLS traffic monitoring attributes and benefits

Describes how MPLS traffic monitoring solutions use NetFlow to provide visibility into MPLS traffic, identify anomalies, detect cyber threats, and support rapid response to security incidents.

Key attributes monitored by MPLS traffic monitoring solutions include:

- MPLS labels

BGP traffic monitoring

Explains how NetFlow monitors BGP traffic and identifies key BGP traffic attributes for network analysis.

A BGP traffic monitoring solution is a network analysis tool that

- uses NetFlow to capture and analyze BGP packets in the network,
- provides detailed information on the frequency, direction, and content of BGP traffic, and
- enables identification of potential security threats and monitoring of BGP router behavior.

BGP traffic monitoring attributes and analysis

Describes how NetFlow monitors BGP traffic and identifies key BGP traffic attributes for network analysis.

BGP (Border Gateway Protocol) is a routing protocol that network routers use to exchange routing information and establish optimal data paths across networks. Monitoring BGP communication with NetFlow provides valuable insights into network performance and helps ensure effective routing strategies.

Key BGP traffic attributes monitored include:

- Next-hop address
- Source autonomous system (AS) number
- Destination autonomous system (AS) number
- Source prefix mask
- Destination prefix mask
- BGP next hop
- BGP Policy Accounting traffic index

SRv6 traffic monitoring

Explains how SRv6 traffic monitoring enables performance analysis of SRv6-based core networks using IPFIX and highlights key restrictions.

A SRv6 traffic monitoring is a network feature that

- enables the collection of information elements specific to SRv6 traffic flow in the core
- using IPFIX is supported only on P-nodes; decapsulation nodes are not supported, and
- helps users understand when and how to apply the feature.

Restriction and limitation

Describes the SRv6 traffic monitoring feature, its support for performance monitoring of SRv6-based core networks using IPFIX, and key restrictions.

- SRv6 traffic monitoring using IPFIX is supported only on P-nodes; decapsulation nodes are not supported.

Interface types supported with NetFlow

Lists the interface types that NetFlow supports.

Lists the interface types that NetFlow supports.

The following interface types are supported with NetFlow:

- Physical main interfaces
- L3 interfaces
- L3 subinterfaces
- L2 interfaces
- Bundle interfaces
- Bundle sub-interfaces
- PW-Ether interfaces
- BVI interfaces

NetFlow Guidelines and Limitations

Describes this topic.

Objective and scope.

General

Features not Supported

Flow Exporter

Flow Monitor

Interfaces

IPFIX 315

BGP Attributes - MPLS Record Types on Cisco 8010 Series Routers

- NetFlow is supported in the ingress direction for all routers.
- NetFlow supports export format Version 9 and IPFIX.
- The configuration of the sampler rate as 1 out of 1 is supported.
- You can configure only one active sampler per system.
- Netflow does not cache for MPLS decap and GRE decap nodes when these are configured on sub-interfaces; cache support is available only on main interfaces.

- Netflow record collection is not supported on SRv6 decap node.
- Ingress L2 netflow or IPFIX is not supported.
- Egress NetFlow and Internet Protocol Flow Information Export (IPFIX) are not supported.
- The full Packet Capture (FPC) feature is not supported.
- Destination-based NetFlow accounting is not supported.
- NetFlow filtering using ACL is not supported.
- The Post-QoS Data Monitoring feature is not supported on Q100 ASIC-based line cards.
- Netflow exporter packet does not support TCP.
- A source interface or source address must be configured to enable the exporter. If you do not configure a source interface, the exporter remains in a disabled state. If both a source interface and a source address are configured, the source address takes precedence.
- A valid record type such as IPv4, IPv6, or MPLS must be configured for every flow monitor map
- NetFlow is not supported on Bridge Virtual Interface (BVI).
- The data and flow records for GRE transit traffic do not have the output interface, source, and destination prefix lengths fields set.
- We do not recommend using the management interface to export the NetFlow packets.
- The output interface field is not updated in data and flow records when the traffic is routed through ACL-based forwarding (ABF).
- If IPFIX 315 is enabled on a line card, then all the ports on that line card should have IPFIX315 configured.
- Enable the IPFIX 315 configuration on a sub interface explicitly if the traffic for that sub interface is required to be exported.
- The incoming and outgoing interface information will point to sub-interface if routed via sub-interface.
- For IPFIX 315, the outgoing interface information may not be correct incase of packets that are multicasted or broadcasted on multiple ports.
- On a dual RP centralized system, IPFIX 315 or sFlow has to be enabled on both RP's respectively. Otherwise, the system displays an error message during configuration.
- On edge nodes, only ingress direction is supported.
- The MPLS record types, such as mpls ipv4-fields, mpls ipv6-fields, and mpls ipv4-ipv6-fields, are only supported.
- At the edge node on MPLS or SR-MPLS core with Penultimate Hop Popping (PHP) or Ultimate Hop Popping (UHP) disabled, the packets that are sampled should contain the MPLS label to fetch the BGP attributes.

Starting from Cisco IOS XR Software Release 24.2.1, Guidelines for BGP Attributes - MPLS Record Types on Cisco 8010 Series Routers

- NetFlow can be configured only in the ingress direction.
- Netflow v9, IPFIX, and IPFIX 315 support a maximum of two sampler maps.
- A source interface must always be configured. If you do not configure a source interface, the exporter will remain in a disabled state.
- Only export format Version 9 is supported.

- A valid record map name must always be configured for every flow monitor map.
- NetFlow on sub-interface routed via BVI is not supported.
- Destination-based Netflow accounting is not supported, only IPv4, IPv6 and MPLS record types are supported under monitor-map.
- Output interface field is not updated in data and flow records when the traffic is routed through ACL based forwarding (ABF).
- Output interface field is not updated in data and flow records for the multicast traffic.
- Output interface, source and destination prefix lengths fields are not set in data and flow records for GRE transit traffic.
- In-line modification of NetFlow configuration is not supported.
- For Netflow IPFIX315, configure the command.
- If IPFIX315 is enabled on a line card then all the ports on that line card should have IPFIX315 configured.
- For **hw-module profile qos hqos-enable** , NetFlow does not give the output interface for cases like L2 bridging, xconnect, IPFIX, and so on.
- L4 header port numbers are supported only for TCP and UDP.
- NetFlow does not give the output interface for traffic terminating on GRE tunnel.

Netflow Specific Guidelines and Limitations:

- NetFlow can be configured only in the ingress direction.
- Supports up to 4 sampling Rates (or Intervals) per ethernet line card; there is no such limit for enhanced ethernet line card.
- Up to 4k interfaces/sub-interfaces can be configured with flow monitor per system (4K system limitation).
- Supports up to 8 flow exporters per flow monitor.
- Supports up to 1 million flow entries per line card.
- Supports up to 50,000 flows per second with line card CPU usage up to 50% per ethernet line card.
- Supports up to 100,000 flows per second with line card CPU usage up to 50% per enhanced ethernet line cardd.
- Netflow scale is increased to 200,000pps on enhanced ethernet line cards.
- Supports exporting packet rates up to 50,000 flows per second (100,000 flows per second on enhanced ethernet line cards) with line card CPU usage up to 50%.
- A source interface must always be configured. If you do not configure a source interface, the exporter will remain in a disabled state.
- When the netflow configuration for VPNv4 or VPNv6 is applied in label allocation mode (either per prefix or per CE) then the IPv4 or IPv6 netflow do not capture the BGP attributes such as BGP nh, BGP AS numbers and prefix lengths; these attributes values are set to zero.
- Under VPNv4 and VPNv6 label allocation mode per vrf, BGP attributes, source and destination lengths are captured but AS numbers are not captured.
- Netflow is not supported on BNG subscriber.
- Only export format Version 9 and IPFIX is supported.
- A valid record map name must always be configured for every flow monitor map.
- NetFlow is not supported on Bridge Virtual Interface (BVI).

- NetFlow is not supported on sub-interfaces.
- NetFlow on sub-interface routed via BVI is not supported.
- Destination-based Netflow accounting is not supported, only IPv4, IPv6 and MPLS record types are supported under monitor-map.
- Output interface field is not updated in data and flow records when the traffic is routed through ACL based forwarding (ABF).
- Output interface field is not updated in data and flow records for the multicast traffic.
- Output interface, source and destination prefix lengths fields are not set in data and flow records for GRE transit traffic.
- L4 header port numbers are supported only for TCP and UDP.
- In-line modification of the flow attribute **record** of NetFlow configuration is not supported.
- NetFlow does not give the output interface for traffic terminating on GRE tunnel.
- If full packet capture is disabled, then NetFlow captures only IPv4 and IPv6 packets. To enable packet flow for IPv4, IPv6, and L2VPN psuedo wire packets, enable the **hw-module profile netflow fpc-enable location** command.

Flow Filter Specific Guidelines and Limitations

- NetFlow flow filter is supported on physical interface, physical subinterface, bundle interface, and bundle subinterface.
- NetFlow flow filter is not supported on satellite access interface, ICL interface and clusters.
- Flow filter for MPLS traffic netflow is not supported

IPFIX Specific Guidelines and Limitations

- If IPFIX315 is enabled on a line card then all the ports on that line card should have IPFIX315 configured.
- For **hw-module profile qos hqos-enable** , NetFlow does not give the output interface for cases like L2 bridging, xconnect, IPFIX, and so on.
- IPFIX does not support variable-length information element in the IPFIX template
- IPFIX does not support Stream Control Transmission Protocol (SCTP) as the transport protocol

NetFlow protocol versions

Explains the differences between NetFlow version 9 and version 10 (IPFIX) and describes their key attributes for network monitoring.

A NetFlow protocol version is a network feature that

- provides a comprehensive overview of the distinct versions within the NetFlow monitoring protocol, including NetFlow v9 and NetFlow v10 (IPFIX)
- identifies relevant configuration, support, or operational details, and
- helps users understand when and how to apply the feature.

Comparison of NetFlow version 9 and version 10 (IPFIX)

Describes the differences between NetFlow version 9 and version 10 (IPFIX) and highlights their variations in network monitoring protocols.

Multiple versions of the NetFlow protocol exist. This section provides a comprehensive overview of the distinct versions within the NetFlow monitoring protocol, including NetFlow v9 and NetFlow v10 (IPFIX). It highlights the variations between these protocols.

NetFlow Version 9 protocols

Explains NetFlow Version 9 as a template-based protocol that provides flexible record formats and enables enhancements to NetFlow services.

A NetFlow Version 9 protocol is a network feature that

- uses a template-based approach for flexible record formats,
- enables enhancements to NetFlow services without altering the basic flow-record format, and
- supports extensibility for future NetFlow features.

Attributes and enhancements of NetFlow Version 9 protocols

Describes the key attributes and enhancements provided by NetFlow Version 9 protocols.

NetFlow Version 9 protocols offer the following features:

- uses a template-based approach for flexible record formats,
- enables enhancements to NetFlow services without altering the basic flow-record format, and
- supports extensibility for future NetFlow features.

NetFlow options templates

Explains the role and structure of NetFlow options templates in exporting metadata for NetFlow operations.

A NetFlow options template is a network feature that

- communicates the format of metadata associated with NetFlow operations,
- provides information about the NetFlow process rather than individual IP flows, and
- enables the export of additional tables such as sampler options, interface options, and VRF (Virtual Routing and Forwarding) tables.

Role and structure of NetFlow options templates

Describes the function and exported metadata of NetFlow options templates, including sampler options, interface options, and VRF tables.

NetFlow options templates are template records that communicate the format of metadata associated with NetFlow operations. They provide information about the NetFlow process rather than individual IP flows and enable the export of additional tables such as sampler options, interface options, and VRF (Virtual Routing and Forwarding) tables.

Distinct options templates include the sampler options template and the interface options template. The NetFlow process exports these two tables, as well as the VRF table, to provide comprehensive metadata about the NetFlow operation.

Sampler tables

Explains how sampler tables organize and present information about active samplers and their attributes.

A sampler table is a network feature that:

- organizes information about active samplers,

- provides key attributes such as sampler ID, mode, interval, and name, and
- enables collectors to estimate the sampling rate for individual data flows.

Sampler table attributes and usage

Explains how sampler tables organize and present information about active samplers and their attributes.

The sampler table is part of the Sampler Options Template, which, along with Interface Option Templates, helps organize and present information for collectors. Enabling these options simplifies the process for the collector to determine data flow information.

The sampler table provides the following information for each sampler:

Element ID	Field Name	Value
48	SamplerID	This ID is assigned to the sampler. It is used by the collector to retrieve information about the sampler for a data flow record.
49	SamplerMode	This field indicates the mode in which the sampling has been performed.
50	SamplerRandomInterval	This field indicates the rate at which the sampling is performed.
84	SamplerName	This field indicates the name of the sampler.

Interface tables

Explains interface tables that store information about monitored interfaces and enable collectors to derive interface names linked to data flow.

An interface table is a network feature that

- stores information about interfaces monitored for data flow,
- enables the collector to derive interface names linked to the data flow, and
- contains fields that map SNMP indexes to interface names.

Information stored in interface tables

Provides details about the fields and values contained in interface tables used for monitoring data flow.

The interface table contains the following information:

Field Name	Value
ingressInterface	This field indicates the SNMP index assigned to the interface. By matching this value to the Ingress interface in the data flow record, the collector is able to retrieve the name of the interface.
interfaceDescription	This field indicates the name of the interface.

VRF tables

Explains how VRF tables map VRF IDs to names and enable collectors to determine the required VRF name, including export intervals.

A VRF table is a network feature that

- maps VRF IDs to VRF names
- enables collectors to determine the required VRF name, and
- is exported at intervals specified by the optional timeout keyword, which can be configured manually

VRF table mapping and export intervals

Describes how VRF tables map VRF IDs to names and how collectors use this information to determine the required VRF name. It also outlines the export intervals and configuration options.

The VRF table consists of mappings between VRF IDs and VRF names. Using this information, the collector determines the name of the required VRF.

The VRF table is exported at intervals specified by the optional timeout keyword, which can be configured manually. The default value is 1800 seconds.

The VRF table consists of the following information:

Field Name	Value
ingressVRFID	The identifier of the VRF with the name in the VRF-Name field.
VRF-Name	The VRF name has the VRFID value ingressVRFID. The value "default" indicates that the interface is not assigned explicitly to a VRF.

The data records contain ingressVRFID as an extra field in each record. The values of these fields are used to look up the VRF table to find the VRF names. A value of 0 in these fields indicates that the VRF is unknown.

Configure NetFlow Version 9

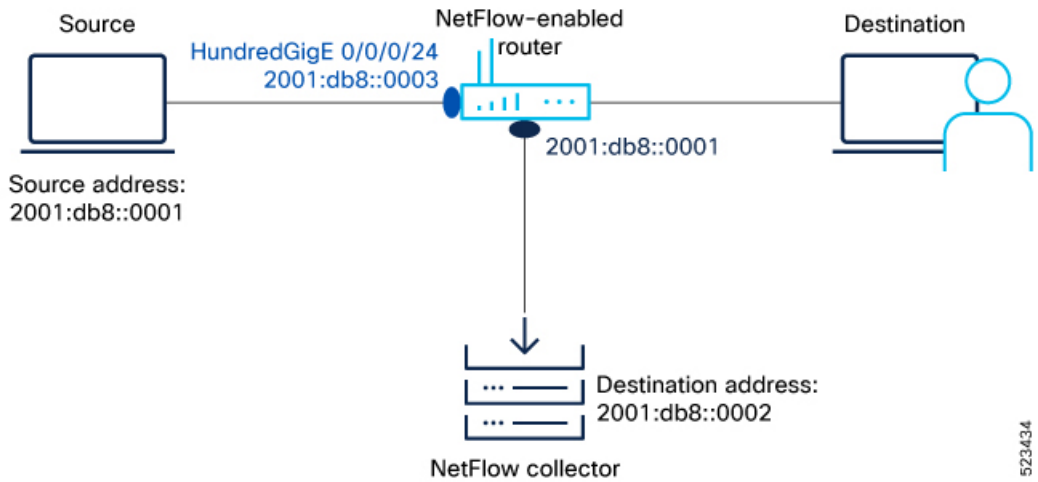
Configure NetFlow Version 9 to monitor and export network traffic data using flow exporter, monitor, and sampler maps.

Configure NetFlow Version 9 to monitor and export network traffic data by setting up flow exporter, monitor, and sampler maps.

- Enable traffic monitoring and export for IPv4, IPv6, MPLS, BGP, SRv6, and GTP-U packets.
- Support up to eight exporters in a single flow monitor map.

This task uses the following topology to configure NetFlow.

Figure 2: NetFlow Version 9 Configuration



To monitor traffic, configure one or more flow exporter maps and associate them to a flow monitor map. Enable NetFlow on the interface in either egress or ingress direction. Optionally, configure a sampler map to set the sampling rate for flow samples.

- Configure exporter maps to specify export destinations.
- Associate exporter maps with flow monitor maps for traffic monitoring.
- Enable NetFlow in ingress or egress direction as required.

Before you begin

Before you begin, gather these details to enable NetFlow on a router:

- Source IP address: `2001:db8::0003`
- NetFlow Collector (destination) IP address: `2001:db8::0002`
- Router interface for NetFlow: `HundredGigE 0/0/0/24`
- NetFlow version for export: `version 9`

Follow these steps to configure NetFlow Version 9 for traffic monitoring and export.

Procedure

1. Configure a Flow Exporter using the flow exporter-map command to specify where and how the packets should be exported.

   Example

   ```
   Router# configure
   Router(config)# flow exporter-map Expo1
   Router(config-fem)# source-address 2001:db8::0003
   Router(config-fem)# destination 2001:db8::0002
   Router(config-fem)# transport udp 1024
   Router(config-fem)# version v9
   Router(config-fem-ver)# options interface-table
   Router(config-fem-ver)# commit
   Router(config-fem-ver)# root
   Router(config)# exit
   ```

   Use the flow exporter-map command to define export parameters such as source address, destination address, transport protocol, and NetFlow version.

   - Configure export options for interface-table and other templates as needed.

If...	Then...
If you need to export NetFlow data to a collector, configure the destination IP address and transport protocol.	Packets are exported to the specified collector using the defined protocol.
If you want to use version 9 for export, set the version parameter accordingly.	NetFlow data is exported in version 9 format.
If you require interface-table options, configure them under the exporter map.	Interface-table information is included in the exported NetFlow data.

Flow exporter is configured and ready for use in NetFlow monitoring.

2. Create a Flow Monitor using the flow monitor-map command to define the type of traffic to be monitored. You can include one or more exporter maps in the monitor map. A single flow monitor map can support up to eight exporters.

   The record type specifies the type of packets that are sampled as the packets pass through the router. MPLS, IPv4, and IPv6 packet sampling is supported.

   - Choose the record type based on the traffic you want to monitor (IPv4, IPv6, MPLS, BGP, SRv6, GTP-U).

If...	Then...
If you want to monitor IPv6 packets, use the record ipv6 command.	IPv6 traffic is monitored and exported.
If you want to monitor MPLS packets, use the record mpls ipv6-fields labels command.	MPLS traffic is monitored and exported.
If you want to monitor BGP packets, use the record ipv6 peer-as command and configure option bgpattr.	BGP traffic is monitored and exported with AS path and community string.

   Examples for recording IP packets, MPLS packets, BGP packets, SRv6, and GTP-U packets:

   -
   ```
   Router# configure
   Router(config)# flow monitor-map fmm-ipv6
   Router(config-fmm)# record ipv6
   Router(config-fmm)# cache entries 500000
   Router(config-fmm)# cache timeout active 60
   Router(config-fmm)# cache timeout inactive 20
   Router(config-fmm)# exporter Expo1
   Router(config-fmm)# commit
   Router(config-fmm)# root
   Router(config)# exit
   ```

   - MPLS Packet Monitoring: Create a flow monitor map to record MPLS packets.

   ```
   Router(config)# flow monitor-map fmm-mpls-ipv6
   Router(config-fmm)# record mpls ipv6-fields labels 3
   Router(config-fmm)# exporter Expo1
   Router(config-fmm)# cache entries 2000000
   Router(config-fmm)# exit
   ```

   - BGP Packet Monitoring: Create a flow monitor map to record BGP packets with a permanent cache.

   ```
   Router(config)# router bgp 50
   Router(config-bgp)# address-family ipv6 unicast
   Router(config-bgp-af)# bgp attribute-download
   Router(config-bgp-af)# root
   Router(config)# flow monitor-map fmm-bgp
   Router(config-fmm)# record ipv6 peer-as
   Router(config-fmm)# exporter Expo1
   Router(config-fmm)# cache entries 2000000
   Router(config-fmm)# exit
   ```

> **Note**
>
> Starting from Cisco IOS XR Software Release 24.2.1, in NetFlow Version 9, to fetch the BGP AS path and community string, configure the option bgpattr command under the flow monitor-map configuration.

- SRv6 Packet Monitoring: Starting from Cisco IOS XR release 7.10.1, create a flow monitor map to record SRv6 packets.

```
Router# configure
Router(config-fem)# flow monitor-map MON
Router(config-fmm)# record ipv6 srv6
Router(config-fmm)# exporter EXP
Router(config-fmm)# cache timeout inactive 5
Router(config-fmm)# sampler-map SAMP
Router(config-fmm)# random 1 out-of 1000
Router(config-fmm)# interface GigabitEthernet0/1/0/0
Router(config-fmm)# ipv6 address 2002:1::1/64
Router(config-fmm)# flow ipv6 monitor M1 sampler SAMP ingress
```

Flow monitor map is configured for the desired packet types and ready for use.

3. Configure a Flow Sampler using the sampler-map command to define the rate at which packet sampling should be performed at the interface where NetFlow is enabled. Use the same sampler map configuration on sub-interfaces and physical interfaces under a port.

Example

```
Router(config)# configure
Router(config)# sampler-map fsm1
Router(config-sm)# random 1 out-of 262144
Router(config)# exit
Router(config)# commit
Router(config)# exit
Router#
```

Set the sampling rate to control the number of packets sampled for NetFlow monitoring.

- Apply the sampler map to all relevant interfaces for consistent sampling.

If...	Then...
If you want to sample 1 out of every 262144 packets, configure random 1 out-of 262144.	Sampling rate is set to 1 out of 262144 packets.
If you apply the sampler map to sub-interfaces and physical interfaces, sampling is consistent across the port.	All interfaces under the port use the same sampling configuration.
If you need to change the sampling rate, modify the random value in the sampler-map configuration.	Sampling rate is updated as per the new configuration.

Flow sampler map is configured and ready for use in NetFlow monitoring.

4. Apply a Flow Monitor Map and a Flow Sampler to a physical interface using the flow command to enable NetFlow on the router. You can enable IPv4, IPv6, or MPLS-aware NetFlow on the interface. Enable NetFlow in the ingress direction to monitor incoming packets.

   Note

   Consider these points before applying the sampler map:

   - Remove any existing NetFlow or sFlow configurations before applying a new Flow sampler on an interface using the no form of the command.
   - Use the same sampler map configuration on sub-interfaces and physical interfaces under a port.

   - Enable NetFlow in ingress direction for monitoring incoming packets.
   - Apply flow monitor and sampler map to the interface as required.

If...	Then...
If you apply the flow monitor and sampler map to the interface, NetFlow monitoring is enabled.	Traffic is monitored and sampled as per configuration.
If you enable NetFlow in ingress direction, incoming packets are monitored.	NetFlow collects data for incoming traffic.
If you remove existing NetFlow or sFlow configurations before applying a new sampler map, configuration conflicts are avoided.	Sampler map is applied without issues.

   Example

   ```
   Router# configure
   Router(config)# interface HundredGigE 0/0/0/24
   Router(config-if)# flow ipv6 monitor fmm-ipv6 sampler fsm1 ingress
   Router(config-if)# commit
   Router(config-if)# root
   Router(config)# exit
   ```

   NetFlow monitoring and sampling are enabled on the interface.

5. View the running configuration to verify the configuration.

   Example

   ```
   Router# show run

   flow exporter-map Expo1
    version v9
     options interface-table
   ```

```
 !
 transport udp 1024
 source-address 2001:db8::3
 destination 2001:db8::2
!
flow monitor-map fmm-ipv6
 record ipv6
 exporter Expo1
 cache entries 500000
 cache timeout active 60
 cache timeout inactive 20
!
sampler-map fsm1
 random 1 out-of 262144
!

interface HundredGigE0/0/0/24
 shutdown
 flow ipv6 monitor fmm-ipv6 sampler fsm1 ingress
!
end
```

Verify that the exporter, monitor, sampler, and interface configurations are present in the running configuration.

- Check for correct source and destination addresses, export version, and sampling rate.

If...	Then...
If the running configuration shows all NetFlow settings, the setup is complete.	NetFlow is ready for traffic monitoring and export.
If any configuration is missing, review and update the settings as needed.	Configuration is corrected and verified.
If you need to verify specific parameters, use show commands for exporter, monitor, and sampler maps.	Detailed configuration is displayed for review.

Running configuration is verified for NetFlow setup.

6. Verify the above configurations using these steps:
   a) Verify the Flow Exporter configuration using the show flow exporter-map command.

Example

```
Router# show flow exporter-map Expo1
Flow Exporter Map : Expo1
------------------------------------------------
Id                  : 1
Packet-Length       : 1468
DestinationIpAddr   : 2001:db8::2
VRFName             : default
SourceIfName        :
SourceIpAddr        : 2001:db8::3
DSCP                : 0
TransportProtocol   : UDP
TransportDestPort   : 1024
Do Not Fragment     : Not Enabled
```

```
Export Version: 9
  Common Template Timeout : 1800 seconds
  Options Template Timeout : 1800 seconds
  Data Template Timeout : 1800 seconds
  Interface-Table Export Timeout : 1800 seconds
  Sampler-Table Export Timeout : 0 seconds
  VRF-Table Export Timeout : 0 seconds
```

b) Verify the Flow Monitor configuration using the show flow monitor-map command.

Example

```
Router# show flow monitor-map fmm-ipv6

Flow Monitor Map : fmm-ipv6
-----------------------------------------------
Id:                 1
RecordMapName:      ipv6
ExportMapName:      Expo1
CacheAgingMode:     Normal
CacheMaxEntries:    500000
CacheActiveTout:    60 seconds
CacheInactiveTout:  20 seconds
CacheUpdateTout:    N/A
CacheRateLimit:     2000
HwCacheExists:      False
HwCacheInactTout:   50
```

c) Verify the sampler map configuration using the show sampler-map command.

Example

```
Router# show sampler-map fsm1

Sampler Map : fsm1
-----------------------------------------------
Id:      1
Mode:    Random (1 out of 262144 Pkts)
Router#
```

Use show commands to verify exporter, monitor, and sampler map configurations.

- Confirm that all NetFlow components are configured and operational.

If...	Then...
If show commands display correct configuration, NetFlow is ready for use.	Traffic monitoring and export are operational.
If any configuration is incorrect, update the settings and verify again.	Configuration is corrected and NetFlow is operational.

If...	Then...
If you need to troubleshoot, review show command outputs for errors or missing parameters.	Issues are identified and resolved.

NetFlow configuration is verified and ready for traffic analysis.

NetFlow Version 9 is configured to monitor and export network traffic data. You can now verify and analyze traffic flows using a NetFlow analyzer.

What to do next

You can now analyze the exported data using a NetFlow analyzer.

- Review traffic statistics and flow records for network analysis.

Verify NetFlow Version 9

Verify the flows captured using the show flow monitor name cache command and analyze cache records for SRv6 L2, QoS, and GTP-U services.

Use this task to confirm that NetFlow Version 9 is capturing and exporting flow data as expected, including specialized records for SRv6 L2, QoS, and GTP-U services.

- Ensure that flow data is accurately captured and exported for advanced services.

NetFlow Version 9 provides detailed flow monitoring and export capabilities, supporting advanced service records such as SRv6 L2, QoS, and GTP-U. This task helps validate that these records are present and correct in the exported cache.

Use this task after configuring NetFlow to verify operational status and exported data integrity.

- Perform this verification as part of routine monitoring or troubleshooting.

Before you begin

Ensure NetFlow Version 9 is configured and operational on the device.

- Access to the device CLI is required.

Follow these steps to verify NetFlow Version 9 flow capture and export.

Procedure

Verify the flows captured using the command.

Example

```
Router# show flow monitor fmm-ipv6 cache summary location 0/0/CPU0
Cache summary for Flow Monitor monitor1:
Cache size: 1000000
Current entries: 295
Flows added: 184409
Flows not added: 0
Ager Polls: 9824
 - Active timeout 183855
```

```
 - Inactive timeout 259
 - Immediate 0
 - TCP FIN flag 0
 - Emergency aged 0
 - Counter wrap aged 0
 - Total 184114
Periodic export:
- Counter wrap 0
- TCP FIN flag 0
Flows exported 184114
```

Review the cache summary to confirm the number of flows added and exported. Analyze the cache records for specialized services as needed.

- Cache summary: In this example, verify the amount of flows added and exported.

If...	Then...
Cache summary shows flows added and exported.	NetFlow is capturing and exporting flows as expected.
SRv6 L2 service records are present.	SRv6 L2 flows are being monitored and exported.
QoS and GTP-U records are present in cache.	Advanced service flows are being captured and exported.

- Review cache summary for flow statistics.

```
Router# show flow monitor fmm-ipv6 cache summary location 0/0/CPU0
Cache summary for Flow Monitor monitor1:
Cache size: 1000000
Current entries: 295
Flows added: 184409
Flows not added: 0
Ager Polls: 9824
 - Active timeout 183855
 - Inactive timeout 259
 - Immediate 0
 - TCP FIN flag 0
 - Emergency aged 0
 - Counter wrap aged 0
 - Total 184114
Periodic export:
- Counter wrap 0
- TCP FIN flag 0
Flows exported 184114
```

Cache records for SRv6 L2, QoS, and GTP-U services are verified and exported as expected.

NetFlow Version 9 is confirmed to be capturing and exporting flow data, including advanced service records. You can now analyze the exported data using a NetFlow analyzer.

What to do next

You can now analyze the exported data using a NetFlow analyzer.

- Review exported flow records for further troubleshooting or reporting.

Modify NetFlow Configuration

Modify NetFlow Configuration explains the key information for this topic.

You can modify only the following flow attributes that is already applied to an interface for a monitor map, exporter map, or a sampler map.

Note that when you modify the flow attributes, the cache counters are cleared and results in resetting of the counters. As a result there could be flow accounting mismatch.

Table 8: Flow Entities and Flow Attributes that can be altered

Flow Entity	Flow Attribute	Command
Monitor map	cache timeout - active - inactive - update - rate-limit	
	exporter	
	cache entries	
	cache permanent	
	options outphysint | bgpattr | filtered | outbundlemember	
Exporter Map	source <*source interface*>	
	destination <*destinaiton address*>	
	dscp <*dscp_value*>	
	version v9 | ipfix	
Sampler Map	sampling interval	

IPFIX (NetFlow version 10)

Explains the IPFIX (NetFlow version 10) protocol, its enhancements, export format, and sampling behavior for network traffic analysis.

A IPFIX (NetFlow version 10) is a network protocol that

- enables the transmission of flow information from exporters to collectors,
- uses templates for efficient and flexible data formatting, and

- supports advanced features such as IPv6 flow records, optional data fields, and multiple collectors.

IPFIX protocol enhancements and export behavior

Sampling and exporting information

Describes the IPFIX (NetFlow version 10) protocol, its enhancements, export format, and sampling behavior for network traffic analysis.

IPFIX (Internet Protocol Flow Information Export), standardized by the Internet Engineering Task Force (IETF), is an export protocol for transmitting NetFlow packets. Building upon NetFlow version 9, IPFIX introduces efficient flow data formatting through templates, ensuring scalability and adaptability to diverse network environments. IPFIX uses UDP as the transport protocol to transfer NetFlow information from exporters to collectors. With native support for IPv6 flow records, optional data fields, and the ability to send data to multiple collectors, IPFIX enables comprehensive traffic analysis, monitoring, and enhanced visibility into network behavior.

The Internet Engineering Task Force (IETF) has standardized Internet Protocol Flow Information Export (IPFIX) as an export protocol for sending IP flow information. Routers support the IPFIX 315 format for exporting flow information. The IPFIX 315 format enables the transmission of 'n' octets of frame information starting from the Ethernet header up to the transport header of the traffic flow over the network. IPFIX 315 supports the sending of variable-sized packet records with variable payload information, such as IPv4, IPv6, MPLS, and nested packets like OuterIP-GRE-InnerIP. The process involves sampling and exporting the traffic flow information. Along with the Ethernet frame information, the IPFIX 315 format exports the information of the incoming and outgoing interfaces of the sampled packet.

The information of the packets flowing through a device is used for a variety of purposes, including network monitoring, capacity planning, and traffic management.

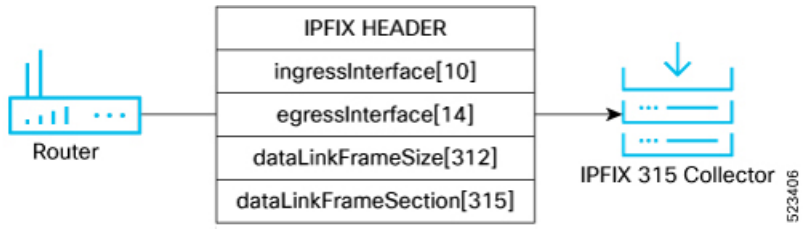
When exporting packets, a special cache-type called Immediate Aging is used. Immediate Aging ensures that the flows are exported as soon as they are added to the cache.

To sample the traffic flow information, configure a sampler-map that specifies the rate at which packets (one out of every 'n' packets) are sampled. Not all packets flowing through a device are exported; only the packets selected based on the sampling rate are exported.

- The size of the exported packet is determined by the sampled packet size from the outermost Layer 2 header.
- If the sampled packet size is less than 160 bytes, the actual packet size is exported.
- If the sampled packet size is 160 bytes or more, the exported packet size is capped at 160 bytes.
- The exported packet includes data up to the Layer 4 header if present; however, if the Layer 4 header is not found within the first 160 bytes, the export size is limited to 160 bytes.
- The location of the Layer 4 header is not used to reduce the exported packet size below the 160-byte cap.

This figure shows exported packet information for IPFIX 315 Export Packet Format.

Figure 3: IPFIX 315 Export Packet Format



Lists the feature history for IPFIX flow record enhancements for L2 and L3 traffic.

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
IPFIX flow record enhancements for L2 and L3 traffic	Release 7.2.12	This release introduces: - Support for flow-based IPFIX protocol version 10(v10), for L2 interfaces. Only L3 interfaces were supported in previous releases. - A new record-type, MPLS-IPv4, to capture BGP next-hop information.

Monitoring post-QoS data in NetFlow and IPFIX

Describes how monitoring post-QoS data in NetFlow and IPFIX enables in-depth analysis of network performance and QoS policy effectiveness.

A monitoring post-QoS data in NetFlow and IPFIX feature is a network visibility enhancement that

- enables export of post-QoS information elements for NetFlow and IPFIX
- allows in-depth analysis of packet-level QoS characteristics, and
- provides comprehensive visibility into how QoS policies affect network traffic.

The post-QoS information element field export for NetFlow and IPFIX allows monitoring of QoS policies. This improvement focuses on the analysis of packet-level information by incorporating the export of Post-QoS details, specifically the Differentiated Services Code Point (DSCP) in IPv4 and Traffic Class in IPv6.

Post-QoS monitoring in NetFlow and IPFIX

Describes how monitoring post-QoS data in NetFlow and IPFIX enables in-depth analysis of network performance and QoS policy effectiveness. This feature facilitates in-depth monitoring of sampled packets, emphasizing their post-processing QoS characteristics.

In NetFlow v9 or IPFIX packets, the post-QoS field utilizes the information element postIpClassOfService (IE55) for collecting post-QoS data. This feature also provides visibility to both pre- and post-QoS Type of Service (TOS)/DSCP values within the show command, enabling comprehensive assessment of network performance.

Note

The ingress QoS (pre-QoS) parameters are already exported in prior IOS-XR software releases.

Key information elements:

- ipClassOfService: For IPv4, represents the value of the Type of Service (TOS) field in the IPv4 packet header. For IPv6 packets, signifies the value of the Traffic Class field in the IPv6 packet header. Available prior to IOS-XR software release 24.1.1.
- postIpClassOfService: Identical to ipClassOfService, except it reports a potentially modified value caused by a router function after the packet has passed the observation point. Introduced in IOS-XR software release 24.1.1.

Lists the feature history and supported platforms for monitoring post-QoS data in NetFlow and IPFIX.

NetFlow versions

Explains the differences between NetFlow v9 and NetFlow v10 (IPFIX) based on transport, compatibility, flexibility, and information elements.

A NetFlow version is a network flow monitoring protocol that

- provides transport options for exporting flow data
- ensures compatibility with previous NetFlow versions, and
- offers varying flexibility and extensibility for information elements

Comparison of NetFlow v9 and NetFlow v10 (IPFIX)

Describes the differences between NetFlow v9 and NetFlow v10 (IPFIX) in transport, compatibility, flexibility, and information elements.

NetFlow v9 and NetFlow v10 (IPFIX) differ in several key areas:

Lists the main factors that distinguish NetFlow v9 from NetFlow v10 (IPFIX).

Table 10: NetFlow v9 and NetFlow v10 (IPFIX)

Factor	NetFlow v9	NetFlow v10 (IPFIX)
Transport	Typically uses UDP	Supports both UDP and TCP transport protocols
Compatibility	Compatible with older NetFlow versions	Backward-compatible with NetFlow v9
Flexibility	Fixed set of predefined fields	More flexible with variable-length information elements and custom-defined attribute
Information Elements	Limited set of predefined fields	Extensive list of predefined elements

Configure IPFIX

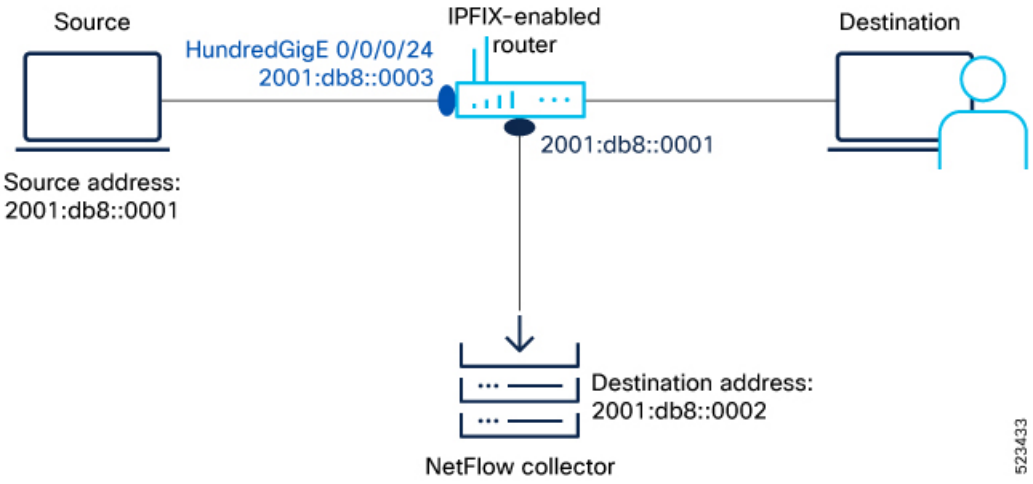
Configure IPFIX to monitor traffic by exporting flow data to a collector using exporter, monitor, and sampler maps.

Configure IPFIX to monitor network traffic by exporting flow data to a collector using exporter, monitor, and sampler maps.

- Enable traffic monitoring and analysis on the router.
- Export flow data for external collection and review.

This task configures IPFIX for traffic monitoring using exporter, monitor, and sampler maps. The configuration enables exporting flow data to a collector for analysis.

Figure 4: NetFlow IPFIX Configuration



To monitor traffic, configure one or more Flow exporter details on page 10 and associate it to a Flow monitor on page 10. Enable IPFIX on the interface in either egress or ingress direction. Optionally, configure a Flow sampler configuration parameters on page 10 to set the sampling rate for flow samples.

- IPFIX enables detailed traffic monitoring and flow export for analysis.

Before you begin

Ensure you have access to the router CLI and the following details:

- Source IP address: `2001:db8::0001`
- IPFIX Collector (destination) address: `2001:db8::0002`
- Router interface for IPFIX: `HundredGigE 0/0/0/24`
- NetFlow version: `IPFIX`

Follow these steps to configure IPFIX for traffic monitoring and flow export.

Procedure

1. Gather the required details to enable IPFIX on the router.

   Example

   ```
   Source IP address: 2001:db8::0001
   IPFIX Collector address: 2001:db8::0002
   Router interface: HundredGigE 0/0/0/24
   NetFlow version: IPFIX
   ```

   Collect all necessary information before starting the configuration.

   - Source IP address: 2001:db8::0001
   - IPFIX Collector address: 2001:db8::0002
   - Router interface: HundredGigE 0/0/0/24
   - NetFlow version: IPFIX

If...	Then...
If all required details are available	Proceed to configure IPFIX.
If any detail is missing	Obtain the missing information before continuing.
If incorrect values are used	Configuration may fail or traffic may not be monitored correctly.

All required details are gathered for IPFIX configuration.

2. Configure a Flow Exporter using the flow exporter-map command to specify where and how packets are exported.

Example

```
Router(config)# flow exporter-map fem_ipfix
Router(config-fem)# destination 2001:db8::0002
Router(config-fem)# source Loopback 0
Router(config-fem)# transport udp 9001
Router(config-fem)# version ipfix
Router(config-fem-ipfix)# template data timeout 600
Router(config-fem-ipfix)# options interface-table
Router(config-fem-ipfix)# exit
```

Verify the Flow Exporter configuration using the show flow exporter-map command.

- Use the show flow exporter-map command to confirm exporter settings.

If...	Then...
If exporter configuration is correct	Flow data will be exported to the collector.
If exporter configuration is incorrect	Flow export may fail or data may not reach the collector.
If show flow exporter-map output matches configuration	Exporter is configured as intended.

```
Router# show flow exporter-map fem_ipfix
Flow Exporter Map : fem_ipfix
-----------------------------------------------
Id                  : 1
Packet-Length       : 1468
DestinationIpAddr   : 2001:db8::2
VRFName             : default
SourceIfName        :
SourceIpAddr        : 2001:db8::3
DSCP                : 0
TransportProtocol   : UDP
TransportDestPort   : 1024
Do Not Fragment     : Not Enabled

Export Version: IPFIX
  Common Template Timeout : 1800 seconds
  Options Template Timeout : 1800 seconds
  Data Template Timeout : 1800 seconds
  Interface-Table Export Timeout : 1800 seconds
```

```
  Sampler-Table Export Timeout : 0 seconds
  VRF-Table Export Timeout : 0 seconds
```

Flow Exporter is configured and verified.

3. Create a Flow Monitor using the flow monitor-map command to define the type of traffic to be monitored. You can include one or more exporter maps in the monitor map. A single flow monitor map can support up to eight exporters.

   Example

```
Router(config)# flow monitor-map fmm1
Router(config-fmm)# record ipv6
Router(config-fmm)# option filtered
Router(config-fmm)# exporter fem_ipfix
Router(config-fmm)# cache entries 65535
Router(config-fmm)# cache timeout active 1800
Router(config-fmm)# cache timeout inactive 15
Router(config-fmm)# exit
```

   The record type specifies the type of packets that are sampled as the packets pass through the router. MPLS, IPv4, and IPv6 packet sampling is supported.

   - Choose the record type based on the traffic to be monitored.

If...	Then...
If record type is set to ipv6	IPv6 packets will be monitored.
If record type is set to ipv4	IPv4 packets will be monitored.
If record type is set to mpls	MPLS packets will be monitored.

```
Router# show flow monitor-map fmm1

Flow Monitor Map : fmm1
-----------------------------------------------
Id:                1
RecordMapName:     ipv6
ExportMapName:     Expo1
CacheAgingMode:    Normal
CacheMaxEntries:   500000
CacheActiveTout:   60 seconds
CacheInactiveTout: 20 seconds
CacheUpdateTout:   N/A
CacheRateLimit:    2000
HwCacheExists:     False
HwCacheInactTout:  50
```

Flow Monitor is configured and verified.

4. Configure a Flow Sampler using the sampler-map command. Use the same sampler map configuration on the sub-interfaces and physical interfaces under a port.

Example

```
Router(config)# configure
Router(config)# sampler-map fsm1
Router(config-sm)# random 1 out-of 4000
Router(config)# exit
Router(config)# commit
Router(config)# exit
Router#
```

Verify the sampler map configuration using the show sampler-map command.

- Apply the sampler map consistently across interfaces.

If...	Then...
If sampler map is applied to all interfaces	Sampling rate is consistent across the router.
If sampler map is not applied to all interfaces	Sampling may be inconsistent.
If show sampler-map output matches configuration	Sampler map is configured as intended.

```
Router# show sampler-map fsm1

Sampler Map : fsm1
-------------------------------------------------
Id:      1
Mode:    Random (1 out of 4000 Pkts)
Router#
```

Flow Sampler is configured and verified.

5. View the running configuration to verify the configuration.

Example

Review the running configuration to confirm all settings are applied.

- Verify exporter, monitor, and sampler map configurations.

If...	Then...
If running configuration matches intended settings	IPFIX is configured as required.
If running configuration does not match	Review and correct configuration as needed.
If all maps are present	Traffic monitoring and export are enabled.

Configuration is verified in the running configuration.

6. Apply a Monitor Map and a Sampler Map to a physical interface using the flow command to enable IPFIX on the router.

Example

```
Router(config)# interface HundredGigE 0/0/0/24
Router(config-if)# flow ipv4 monitor fmm1 sampler fsm1 ingress
Router(config-if)# exit
```

Enable IPFIX monitoring on the desired interface.

- Apply monitor and sampler maps to the interface for traffic monitoring.

If...	Then...
If monitor and sampler maps are applied to the interface	IPFIX traffic monitoring is enabled.
If maps are not applied	Traffic monitoring will not occur on that interface.
If configuration is committed	Settings take effect immediately.

Monitor and sampler maps are applied to the interface. IPFIX is enabled.

IPFIX is now configured on the router. Traffic flow data is exported to the collector for monitoring and analysis.

What to do next

Review exported flow data in the collector to confirm monitoring and analysis are functioning as expected.

- Verify collector receives flow data.

Configure IPFIX 315

Configure IPFIX 315 for flow monitoring and statistics export on Cisco IOS XR Software.

Configure IPFIX 315 to enable advanced flow monitoring and export flow statistics on Cisco IOS XR Software.

Use this task to enable and configure IPFIX 315 for enhanced visibility into network traffic and to export flow statistics for analysis.

Procedure

---

1. Enable IPFIX 315 for flow monitoring.

   Example

   ```
   Router(config)# hw-module profile netflow ipfix315-enable
   ```

   Enables the IPFIX 315 profile for NetFlow on the device.

   IPFIX 315 is enabled for flow monitoring.

2. Configure an exporter map with IPFIX as the exporter version using the flow exporter-map command in global configuration mode to specify where and how the packets should be exported.

Example

```
Router(config)# flow exporter-map ipfix_exp
Router(config-fem)# version ipfix
Router(config-fem-ipfix)# template data timeout 10
Router(config-fem)# dscp 63
Router(config-fem)# transport udp 12000
Router(config-fem)# source Loopback 0
Router(config-fem)# destination 100.10.1.159
Router(config-fem)# exit
```

Defines the exporter map for IPFIX, sets the export version, template timeout, DSCP, transport protocol and port, source interface, and destination address.

The exporter map is configured for IPFIX export.

3. Create a flow monitor using the flow monitor-map command in global configuration mode to define the type of traffic to be monitored. You can include one or more exporter maps in the monitor map.

Example

```
Router(config)# flow monitor-map ipfix_mon
Router(config-fmm)# record datalinksectiondump
Router(config-fmm)# exporter ipfix_exp
Router(config-fmm)# cache immediate
Router(config-fmm)# exit
```

Creates a flow monitor map, specifies the record type, associates the exporter map, and configures the cache behavior.

The flow monitor map is created and linked to the exporter map.

4. Configure a sampler map using the sampler-map command to define the rate at which the packet sampling should be performed at the interface where IPFIX is enabled.

Example

```
Router# sampler-map ipfix_sm
Router(config-sm)# random 1 out-of 32000
Router(config)# exit
```

Configures a sampler map to randomly sample 1 out of every 32,000 packets for flow monitoring.

The sampler map is configured for packet sampling.

5. Apply a monitor map and a sampler map to a physical interface using the flow command to enable IPFIX on the router.

Example

```
Router(config)# interface TenGigE0/0/0/5
Router(config)# interface TenGigE0/0/0/1
Router(config-if)# ipv4 address 192.1.108.2 255.255.255.0
Router(config-if)# ipv6 address 1:108::2/64
Router(config-if)# flow datalinkframesection monitor ipfix_mon sampler ipfix_sm
 ingress
Router(config-if)# encapsulation dot1q 139
```

Assigns the flow monitor and sampler maps to the interface and configures IP addresses and encapsulation as needed.

The monitor and sampler maps are applied to the interface for IPFIX monitoring.

6. Verify the sampled and exported flow statistics using the show flow platform producer statistics location command.

Example

```
Router# show flow platform producer statistics location 0/0/0/5
Router# show flow platform producer statistics location 0/0/CPU0
Netflow Platform Producer Counters:
IPv4 Ingress Packets: 0
IPv4 Egress Packets: 0
IPv6 Ingress Packets: 0
IPv6 Egress Packets: 0
MPLS Ingress Packets: 0
MPLS Egress Packets: 0
IPFIX315 Ingress Packets: 630478
IPFIX315 Egress Packets: 0
Drops (no space): 0
Drops (other): 0
Unknown Ingress Packets: 0
Unknown Egress Packets: 0
Worker waiting: 2443
```

This show output demonstrates that the system has actively received and monitored a total of 630,478 IPFIX 315 packets.

Sampled and exported flow statistics are verified.

7. Verify the flow monitor statistics using the show flow monitor cache location command.

Example

```
Router# show flow platform producer statistics location 0/0/0/5
Router# show flow platform producer statistics location 0/0/CPU0
Cache summary for Flow Monitor ipfix_mon:
Cache size: 65535
Current entries: 0
Flows added: 50399
Flows not added: 0
Ager Polls: 2784
- Active timeout 0
- Inactive timeout 0
- Immediate 50399
- TCP FIN flag 0
- Emergency aged 0
- Counter wrap aged 0
- Total 50399
Periodic export:
- Counter wrap 0
- TCP FIN flag 0
Flows exported 50399
Matching entries: 0
```

This example shows that there were 50,399 flows added to the cache and exported.

Flow monitor statistics are verified.

IPFIX 315 is configured and operational for flow monitoring and statistics export on Cisco IOS XR Software.

IPFIX information elements for MPLS IPv4 and IPv6

Explains how collecting additional BGP information elements for MPLS IPv4 and IPv6 using IPFIX improves congestion mitigation and network monitoring.

IPFIX information elements for MPLS IPv4 and IPv6 are network features that:

- enable the collection of additional BGP information elements in IPFIX records for MPLS IPv4 and IPv6 traffic,
- improve congestion mitigation in core-edge link scenarios by providing more granular flow data, and
- support efficient network monitoring and optimization by recording key BGP and TTL fields.

IPFIX information elements for MPLS IPv4 and IPv6 reference information

Describes the IPFIX information elements for MPLS IPv4 and IPv6, including their purpose, supported fields, and configuration details.

IPFIX information elements for MPLS IPv4 and IPv6 enable the collection of additional BGP information elements in IPFIX records for MPLS IPv4 and IPv6 traffic, improve congestion mitigation in core-edge link scenarios by providing more granular flow data, and support efficient network monitoring and optimization by recording key BGP and TTL fields.

- Eight additional BGP fields are supported in IPFIX MPLS IPv4/IPv6 records.
- Two new Information Elements, Minimum Time-to-Live (TTL) and Maximum TTL, are recorded for each flow.
- IE number, or Information Element Number, is a unique identifier assigned to specific elements within network communication protocols, facilitating standardized interpretation and management.

For more information, refer to IP Flow Information Export (IPFIX) Entities .

Table 11: Information Elements

IE Field	IE Number
BgpSourceAsNumber	16
BgpDestinationAsNumber	17
BgpNextHopIPv4Address	18
BgpNextHopIPv6Address	63
DestinationIPv4PrefixLength	13
DestinationIPv6PrefixLength	30
IpNextHopIPv4Address	15
IpNextHopIPv6Address	62
Minimum TTL	52
Maximum TTL	53

Configuration example: Collect MPLS traffic with both IPv6 and IPv4 fields.

Configuring Monitor map:

```
Router(config)#flow monitor-map mpls-1
Router(config-fmm)#record mpls ipv4-ipv6-fields
Router(config-fmm)#commit
Router(config-fmm)#exit
```

Configuring Sampler map:

```
Router(config)#sampler-map fsm1
Router(config-sm)#random 1 out-of 4000
Router(config-sm)#commit
Router(config-sm)#exit
```

Apply a Monitor Map and a Sampler Map to a physical interface:

```
Router(config)#interface HundredGigE 0/0/0/24
Router(config-if)#flow mpls monitor mpls-1 sampler fsm1 ingress
Router(config-if)#exit
```

Verification: Use the **show flow monitor cache location command** to verify flow monitor statistics.

```
Router#show flow monitor mpls-1 cache summary location 0/0/CPU0
========== Record number: 1 ==========
LabelType         :   Unknown
Prefix/Length     : 20.1.1.0/24
Label1-EXP-S      :    16001-0-1
Label2-EXP-S      :       -
Label3-EXP-S      :       -
Label4-EXP-S      :       -
Label5-EXP-S      :       -
Label6-EXP-S      :       -
InputInterface    : FH0/0/0/1
OutputInterface   : FH0/0/0/0
ForwardStatus     : Fwd
FirstSwitched     : 00 08:28:52:189
LastSwitched      : 00 08:28:57:649
ByteCount         : 2352
PacketCount       : 56
Dir               : Ing
SamplerID         : 1
IPV4SrcAddr       : 30.1.1.1
IPV4DstAddr       : 20.1.1.1
IPV4TOS           : 0
IPV4Prot          : udp
L4SrcPort         : 2025
L4DestPort        : 2500
L4TCPFlags        : 0
IPV4SrcPrfxLen    : 24
IPV4DstPrfxLen    : 24
BGPNextHopV4      : 192.168.10.10
BGPNextHopV6      : ::
BGPSrcOrigAS      : 2000
BGPDstOrigAS      : 1000
IPV4NextHop       : 192.168.10.10
IPV6NextHop       : ::
MinimumTTL        : 90
MaximumTTL        : 110
InputVRFID        : default
OutputVRFID       : default
```

```
========== Record number: 1 ==========
LabelType         :   Unknown
Prefix/Length     : ::/0
Label1-EXP-S      :    16001-0-1
Label2-EXP-S      :       -
Label3-EXP-S      :       -
Label4-EXP-S      :       -
Label5-EXP-S      :       -
Label6-EXP-S      :       -
InputInterface    : FH0/0/0/1
OutputInterface   : FH0/0/0/0
ForwardStatus     : Fwd
FirstSwitched     : 00 08:27:38:692
LastSwitched      : 00 08:27:47:572
ByteCount         : 5580
PacketCount       : 90
Dir               : Ing
SamplerID         : 1
IPv6SrcAddr       : 50::1
IPv6DstAddr       : 40::1
IPv6TC            : 0
IPv6FlowLabel     : 0
IPv6OptHdrs       : 0x0
IPV6Prot          : udp
L4SrcPort         : 2025
L4DestPort        : 2500
L4TCPFlags        : 0
IPV6SrcPrfxLen    : 64
IPV6DstPrfxLen    : 64
BGPNextHopV4      : 0.0.0.0
BGPNextHopV6      : ::ffff:192.168.10.10
BGPSrcOrigAS      : 2000
BGPDstOrigAS      : 1000
IPV4NextHop       : 192.168.10.10
IPV6NextHop       : ::
MinimumTTL        : 195
MaximumTTL        : 205
InputVRFID        : default
OutputVRFID       : default
```

Note

Starting from Cisco IOS XR Software Release 25.2.1, when processing the ICMP Layer 4 header, the destination port is determined based on the ICMPv6 message type, instead of being set to zero. This behavior is specific to ICMPv6 and does not apply to ICMP for IPv4.

For example, in the following output, the L4DestPort value corresponds to ICMPv6 Msg Type 3 (Time Exceeded). For more information, refer to https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-codes-4.

```
IPv6SrcAddr      : 1700::2
IPv6DstAddr      : 1800::2
BGPDstOrigAS     : 0
BGPSrcOrigAS     : 0
BGPNextHopV6     : fcbb:bb00:3::
IPv6TC           : 8
IPv6FlowLabel    : 4
IPv6OptHdrs      : 0x0
IPV6Prot         : icmpv6
MinimumTTL       : 120
MaximumTTL       : 120
L4SrcPort        : 0
L4DestPort       : 3
L4TCPFlags       : 0
IPV6DstPrfxLen   : 64
IPV6SrcPrfxLen   : 128
InputInterface   : Hu0/0/0/1
OutputInterface  : Hu0/0/0/0
ForwardStatus    : Fwd
BGPNextHopV4     : 0.0.0.0
IPV4NextHop      : 0.0.0.0
IPV6NextHop      : ::
FirstSwitched    : 04 02:36:55:363
LastSwitched     : 04 02:37:19:963
ByteCount        : 25190
PacketCount      : 229
Dir              : Ing
SamplerID        : 4
SrcMacAddr       : 00:ca:ff:ee:00:01
DstMacAddr       : 04:00:00:07:1d:04
EthType          : 34525
Dot1qPriority    : 0
Dot1qVlanId      : 0
CustVlanId       : 0
InputVRFID       : vrf_1
OutputVRFID      : default
```

Feature History Table:

Table 12: Feature History Table

Revision histories

Explains the feature history for NetFlow and IPFIX monitoring features on Cisco 8000 series platforms.

A revision history is a network feature that

- This feature is supported on Cisco 8711-48Z-M routers.
- This information is available because exporting of GTP-U related Information Elements is enabled, and
- This feature modifies the output of the show flow monitor command.

Feature history for NetFlow and IPFIX monitoring features

Describes the feature history, supported platforms, and enhancements for NetFlow and IPFIX monitoring features on Cisco 8000 series platforms.

Lists the feature history for NetFlow and IPFIX monitoring features on Cisco 8000 series platforms.

6 Configure sFlow for network traffic monitoring

Topics:

Configure sFlow to enable scalable network traffic monitoring and analysis using statistical packet sampling.

Configure sFlow to monitor network traffic efficiently by sampling packets and exporting flow records for analysis.

sFlow is a network traffic monitoring technology that uses statistical packet sampling to provide scalable traffic analysis, enables real-time visibility into network usage, and supports integration with various network management tools.

sFlow operates by sampling packets and forwarding flow records to a central collector for analysis, reducing the processing load on network devices compared to full packet capture solutions.

Key concepts include the sFlow agent, which performs packet sampling and exports flow records, and the sFlow collector, which receives and analyzes the sampled data.

- Advantages of sFlow include low overhead, scalability, and vendor interoperability.
- Limitations may include reduced detail compared to full packet capture and reliance on sampling accuracy.

Before you begin

Ensure you have access to the router configuration interface and the IP address of the sFlow collector.

- Verify that your device supports sFlow.
- Determine appropriate sampling rate and polling interval for your monitoring requirements.

Procedure

1. Enable the sFlow agent on the router.

   This activates sFlow functionality and prepares the device for traffic sampling.

   The router is ready to sample network traffic using sFlow.

2. Specify the sFlow collector address.

   Configure the IP address and port of the central sFlow collector that will receive sampled flow records.

   Sampled flow records are directed to the specified collector for analysis.

3. Set the sFlow sampling rate and polling interval.

   Adjust these parameters according to your network monitoring requirements to balance detail and device resource usage.

   The router samples packets and exports flow records at the configured rate and interval.

4. Verify sFlow operation and data export to the collector.

Check the sFlow agent status and confirm that flow records are being received by the collector.

sFlow is operational and exporting network traffic data for analysis.

---

sFlow is configured and operational, providing scalable network traffic monitoring and analysis through statistical packet sampling and flow export.

What to do next

Monitor sFlow statistics and adjust sampling parameters as needed to optimize visibility and performance.

- Review collector data regularly to ensure accurate network analysis.

sFlow essential concepts and terms

Lists and explains the essential sFlow terms and concepts used in network monitoring and analysis.

This reference lists and explains the essential sFlow terms and concepts used in network monitoring and analysis.

The following list explains the essential sFlow terms and concepts used in network monitoring and analysis.

- Data source: A location within a network device that can make traffic measurements. Examples include physical interfaces and VLANs.
- Flow: A set of IP packets passing through a network device during a certain time interval. All packets that belong to a particular flow have a set of common properties derived from the data contained in the packet.
- Flow record: A set of key and non-key sFlow field values used to characterize flows. This record is created by inspecting packet headers and adding a description of packet information.
- sFlow agent: An entity inside the network device responsible for maintaining sFlow configuration, gathering sampled flow and counters from one or more data sources in the router, packaging them in sFlow datagram format, and exporting them to the sFlow collector.
- sFlow collector: An application that receives sFlow datagrams from one or more agents to perform further analysis and generate reports. The collector is external to the router.
- Sampling rate: The frequency that specifies how often packet sampling is performed, and determines how many packets (on average) that pass through the data source to generate a flow sample. A value of 100 means that on average, 1 out of 100 packets is randomly sampled to be exported.
- Sampling interval: The period at which counters are polled for populating the counter sample in the sFlow datagram.
- sFlow datagram: A User Datagram Protocol (UDP) datagram exported from the sFlow agent to the collector. The datagram contains information about the data source, one or more flow samples, and one or more counter samples.
- Collector address: The IP address and UDP port number of the collector. The default destination port number is 6343.

Flow monitoring on egress interfaces

Explains how flow monitoring on egress interfaces enhances network visibility, prioritizes outbound traffic, and improves security through encapsulated and decapsulated data analysis.

A flow monitoring on egress interfaces is a network feature that

- prioritizes outbound traffic for enhanced control,
- enables advanced monitoring and management of data exiting the network, and
- provides comprehensive insights into encapsulated and decapsulated packets using egress sFlow.

Flow monitoring on egress interfaces reference information

Describes how flow monitoring on egress interfaces enhances network visibility, prioritizes outbound traffic, and improves security through encapsulated and decapsulated data analysis.

Flow monitoring on egress interfaces is a network visibility feature that:

- prioritizes outbound traffic for enhanced control,
- enables advanced monitoring and management of data exiting the network, and
- provides comprehensive insights into encapsulated and decapsulated packets using egress sFlow.

Encapsulated and decapsulated data monitoring in sFlow plays a crucial role in protecting sensitive information transmitted across the network. Encapsulation adds an extra layer of information to data, allowing verification of authenticity and integrity, and making it more difficult for attackers to intercept or modify data during transmission. Decapsulation removes this layer, enabling network devices to analyze the underlying information and take appropriate actions in real time. This proactive approach helps identify and prevent attacks or anomalies, improving overall network security.

Lists the feature history and supported platforms for flow monitoring on egress interfaces.

Table 13: Feature History Table

Feature Name	Release Introduced	Description
Flow monitoring on Egress Interface	Release 26.2.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) This feature requires egress feature capability on supported platforms. Use the **hw-module profile edge-mode** CLI command to enable the feature. The feature introduces these changes: CLI: • **hw-module profile edge-mode** • **show hw-module profile edge-mode** *This feature is now supported on: • 8711-48Z-M • 8712-MOD-M

sFlow operations

Describes how sFlow samples network traffic in real time and enables near real-time traffic analysis for network monitoring and capacity planning.

A sFlow operation is a network monitoring method that

- samples network traffic in real time,
- streams packet headers and metadata to an external collector, and
- enables near real-time analysis of network traffic patterns and trends.

sFlow operations and traffic analysis

Describes how sFlow operates by sampling network traffic and transmitting packet headers and metadata to an external collector for near real-time analysis.

sFlow operates by sampling network traffic rather than capturing all packets, which distinguishes it from technologies like NetFlow. Devices in the network disaggregate the flow pipeline and transmit sampled packet headers and metadata as UDP datagrams to an external collector. The collector decodes these datagrams and generates flow records, providing a near real-time view of network activity.

This real-time analysis allows network administrators to monitor patterns and trends, automate traffic engineering, and make informed decisions for network capacity planning.

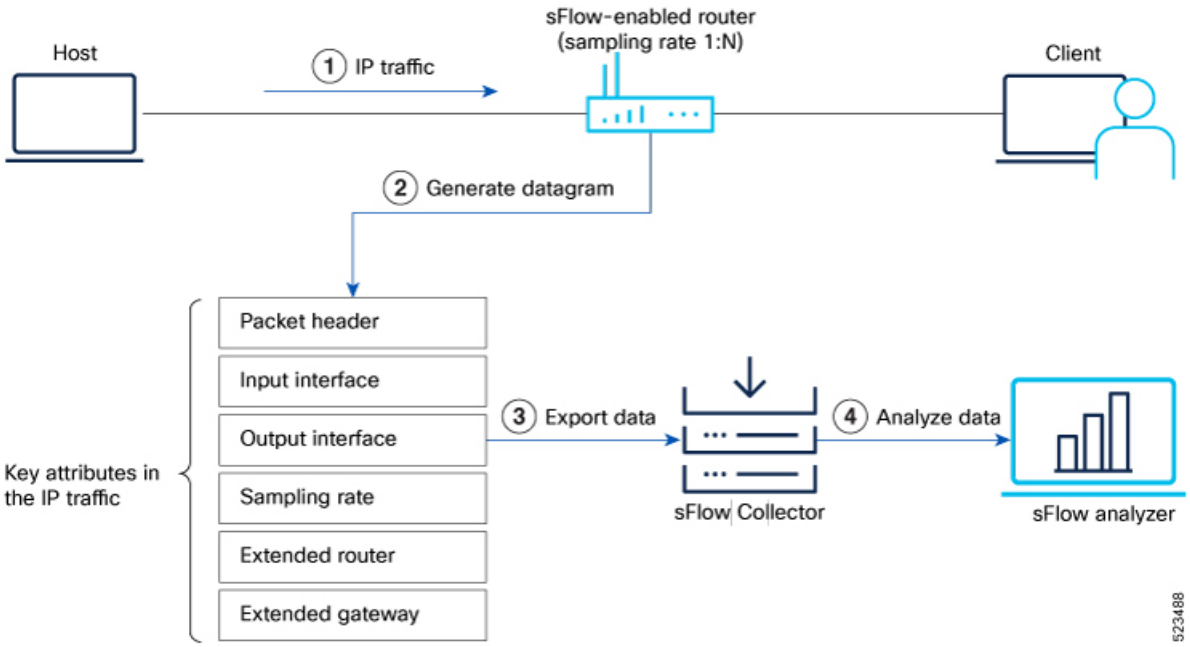
> **Note**
>
> Tip: sFlow sampling provides efficient network monitoring without the overhead of capturing every packet.

Recording of Packet Flows in sFlow

Recording of Packet Flows in sFlow explains the key information for this topic.

The packet in sFlow is recorded as follows:

Figure 5: Packet Flows in sFlow



rect 180, 19, 226, 63 Sampling
rect 226, 97, 272, 137 Datagram Generation
rect 304, 237, 341, 279 Data Export
rect 508, 236, 548, 277 Analysis and Reporting

In sFlow, the focus is on collecting sampled network traffic data rather than recording full packet flows. sFlow is designed to provide a statistical overview of network traffic by sampling packets and extracting relevant information for analysis.

Here's how sFlow handles the recording of packet flows:

1. Sampling: sFlow agent process in network devices sample packets based on a configured sampling rate. The sampling rate determines the percentage of packets that will be selected for analysis. For example, a sampling rate of 1-in-100 means that 1% of the packets will be sampled.
2. Datagram Generation: The sFlow agent generates datagrams that contain information about the sampled packets. These datagrams include details such as packet header, sampling rate, port numbers, protocol information, and various flow statistics.
3. Data Export: The sFlow datagrams are periodically exported from the sFlow agent to a designated sFlow collector or analyzer. The export can be done using protocols like UDP or TCP, and the datagrams are typically sent in a structured format like XDR.

4. Analysis and Reporting: Upon receiving the sFlow data, the sFlow collector or analyzer processes and analyzes the information. It aggregates the sampled data to provide statistical insights into network traffic, including top talkers, protocol distribution, traffic patterns, and other metrics.

sFlow parameters and default values

Lists the sFlow parameters and their default values for configuration on Cisco routers.

Lists the sFlow parameters and their default values for configuration on Cisco routers.

The following table lists the sFlow parameters and default values that you can use when configuring sFlow on the router:

Table 14: sFlow Parameters and Default Values

Parameter	Value	Command
Sampling rate	Default value: 1 out of 10000 packets	**sampler-map**
Sample header size	128 - 343 bytes (from Cisco IOS XR Release 7.3.4 onwards) 128 - 200 bytes (prior to Cisco IOS XR Release 7.3.4) Default value: 128 bytes	**flow monitor-map**
Counter poll interval	5-1800 seconds Default value: None	**flow monitor-map**
Collector port	Configurable. Default value: 6343	**flow exporter-map**

Table 15: Feature History Table

Feature Name	Release Information	Feature Description
Increased sFlow sample-header size	Release 7.3.4	You can now increase the sFlow sampling size to 343 bytes of the incoming or outgoing packet header. This enhancement lets the router export a larger sample to the flow-analyzer tool, enabling the tool to provide more accurate network analytics. In earlier releases, you could configure up to 200 bytes.

sFlow sampling methods

Explains the two primary sFlow sampling methods used for network traffic analysis.

A sFlow sampling method is a network traffic analysis technique that

- enables efficient monitoring of network performance and health,
- offers two distinct approaches for collecting data, and
- supports flexible selection based on monitoring objectives.

sFlow sampling methods for network traffic analysis

Describes the two primary sFlow sampling methods used for network traffic analysis and their key attributes.

The two primary sFlow sampling methods are:

- Counter sampling: Counter sampling collects specific counters or statistics, such as interface utilization, packet drops, CPU usage, and memory usage. This method provides a high-level view of network health and performance without capturing or analyzing individual packets or flows.
- Flow sampling: Flow sampling captures and analyzes sampled network flows, which are sequences of packets sharing common attributes like source and destination IP addresses, port numbers, and protocol information. This method allows for more granular analysis of network traffic, enabling detection of traffic patterns, anomalies, and performance issues.

Network administrators can choose the appropriate sampling method based on specific monitoring needs and objectives.

Configure sFlow

Configure sFlow to monitor network traffic using sampled data.

Enable sFlow to monitor network traffic by collecting sampled data from network devices.

- sFlow provides visibility into network traffic patterns and supports efficient monitoring.

sFlow is used to monitor network traffic by collecting sampled data from network devices, allowing administrators to analyze traffic patterns and optimize network performance.

This task guides you through configuring sFlow on network devices.

- Monitoring sampled data helps identify network issues and supports troubleshooting.

Before you begin

Ensure you have access to the device configuration interface and the necessary privileges to modify sFlow settings.

- Verify that the device supports sFlow and that you have administrative access.

Follow these steps to configure sFlow for network traffic monitoring.

Procedure

1. Access the device configuration interface.

   Example

   ```
   Access the CLI or GUI configuration interface for the network device.
   ```

   Ensure you are connected to the device and have the required privileges.

   - Administrative access is required to configure sFlow.

If...	Then...
The device supports sFlow.	Proceed with configuration steps.
You lack administrative privileges.	Request access from your network administrator.

If...	Then...
The device does not support sFlow.	Consult device documentation for alternative monitoring options.

The device configuration interface is accessible for sFlow setup.

2. Enable sFlow on the desired interfaces.

Example

```
Enable sFlow on each interface where traffic monitoring is required.
```

Choose interfaces based on monitoring needs.

- Enabling sFlow on multiple interfaces increases visibility.

If...	Then...
You enable sFlow on all interfaces.	All traffic is monitored using sampled data.
You enable sFlow on selected interfaces.	Only traffic on those interfaces is monitored.
No interfaces are selected.	sFlow does not monitor any traffic.

sFlow is enabled on the specified interfaces.

3. Specify the sampling rate and collector destination.

Example

```
Configure the sampling rate and set the collector IP address and port.
```

Adjust the sampling rate to balance accuracy and resource usage.

- Lower sampling rates provide more detailed data but use more resources.

If...	Then...
The sampling rate is set low.	More detailed traffic data is collected.
The sampling rate is set high.	Less detailed data is collected, reducing resource usage.
The collector destination is not configured.	sFlow data is not exported for analysis.

Sampling rate and collector destination are configured for sFlow.

4. Save the configuration to apply changes.

Example

```
Save the device configuration to activate sFlow monitoring.
```

Saving the configuration ensures sFlow settings persist after device reboot.

- Unsaved changes may be lost if the device restarts.

If...	Then...
Configuration is saved.	sFlow settings are applied and persist after reboot.
Configuration is not saved.	sFlow settings may be lost after device restart.
Device supports automatic saving.	Changes are applied without manual intervention.

sFlow configuration is saved and monitoring begins.

---

sFlow is enabled and begins monitoring network traffic using sampled data.

What to do next

Verify sFlow operation by checking traffic statistics and collector reports.

- Review collected data to ensure sFlow is functioning as expected.

sFlow guidelines and limitations

Lists the guidelines and limitations for sFlow configuration and operation on Cisco platforms.

Describes the guidelines and limitations for configuring and operating sFlow on Cisco platforms.

The following list outlines the general guidelines and limitations for sFlow configuration and operation on Cisco platforms:

- When egress sFlow is enabled, the Layer 2 information of ingress packets is captured instead of the egress interface. This behavior is not supported on the 88-LC1-12TH24FH-E and 88-LC1-52Y8H-EM line cards.
- sFlow samples are combined into UDP packets and forwarded to sFlow collectors for analysis. UDP is a connectionless protocol and does not ensure delivery of data. Using sFlow as a flow source could lead to inaccurate representations of traffic volumes, bidirectional flows, and a reduction in alerting capabilities.
- Only one sampler configuration per router is allowed.
- A sampling interval of 1 out of 262,144 packets as the maximum is supported.
- Only one active sampler per router is supported. Multiple sampler maps that are not used in flow configuration are allowed.

The following guidelines apply to flow exporter configuration:

- When sFlow is on a bundle with members located on different line cards (LCs), flows are exported with the same ifindex id for the bundle interface. However, they have distinct sub-agent ids and sequence numbers.
- A maximum of 8 export destinations per monitor map for both IPv4 and IPv6 are allowed.

The following guidelines apply to flow monitor configuration:

- Ingress sFlow is supported on Cisco 8200 and Cisco 8800 Series Routers.
- Egress sFlow is supported only on Cisco 8200 Series Routers.

The following guidelines apply to interface support:

- Up to 2000 L3 interfaces are supported.
- Tunnel and Ethernet PseudoWire (PW) interfaces are not supported.

- ARP, multicast, broadcast, and IP-in-IP packets are excluded from the sampling process.

The following guideline applies to load balancing sFlow traffic:

- Cisco 8000 modular routers do not load balance sFlow traffic across multiple ECMP paths. When more than one path is available, the router selects only one. Typically, sending sFlow traffic over a single path is sufficient, and load balancing is not necessary.

  To enable load balancing across all ECMP paths, configure the router to use multiple UDP source ports for sFlow export packets. This method, known as "source-port entropy," uses a hashing-based algorithm to select paths, resulting in multiple ECMP paths being used. This approach does not provide statistics on the number of sFlow packets per path, as the sFlow process does not know the outgoing interface used. To configure multiple source ports, use the **transport udp source-port multiple** command.

Configure sFlow on a router

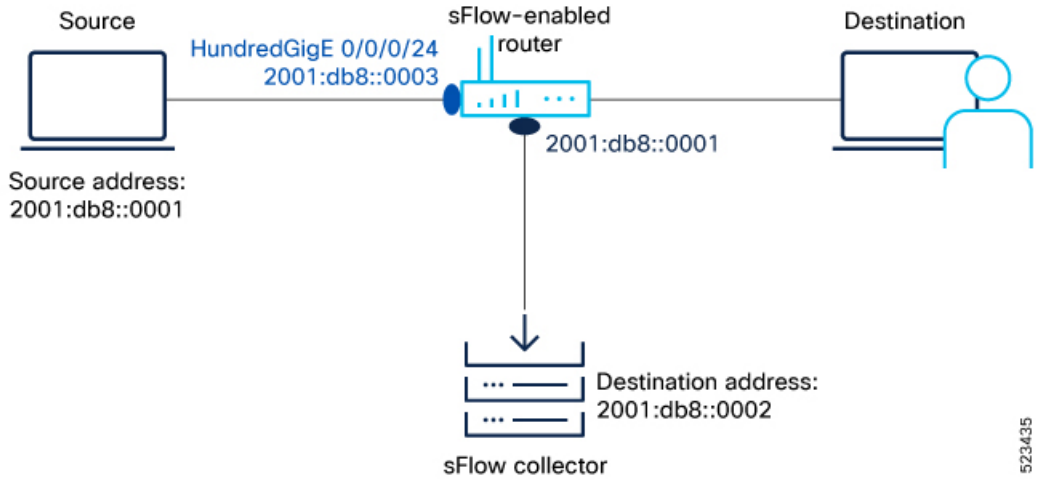
Configure sFlow to monitor network traffic by sampling packets and exporting data to a collector.

Configure sFlow to monitor network traffic by sampling packets and exporting data to a collector for analysis.

- Enable sFlow monitoring to collect traffic data from ingress and egress ports.
- Export sampled data using version 5 format to a centralized sFlow analyzer.

To enable sFlow monitoring, configure the sFlow agent to use a sampling mechanism that forwards traffic data from both ingress and egress ports to a centralized data collector, also known as the sFlow analyzer. The sampled data is sent using the version 5 export format. This task provides instructions for configuring sFlow on the router. The following topology is used as a reference for configuring sFlow.

Figure 6: sFlow Configuration



Before you begin

Ensure you have the following details before configuring sFlow:

- IP address of the source: `2001:db8::0001`
- IP address of the sFlow collector (destination address): `2001:db8::0002`
- Interface of the router where sFlow will be enabled: `HundredGigE 0/0/0/24`
- sFlow version used to transport the data to the collector: `version 5`

Follow these steps to configure sFlow on the router.

Procedure

1. Configure the flow exporter using the flow exporter-map command to specify where and how the packets should be exported.

   Example

```
Router(config)# flow exporter-map EXP-MAP
Router(config-fem)# version sflow v5
Router(config-fem)# packet-length 9000
Router(config-fem)# transport udp 6343
Router(config-fem)# source HundredGigE 0/0/0/1
Router(config-fem)# source-address 192.127.10.1
Router(config-fem)# destination 192.127.0.1
Router(config-fem)# dfbit set
```

   The following attributes can be configured while creating exporter map:

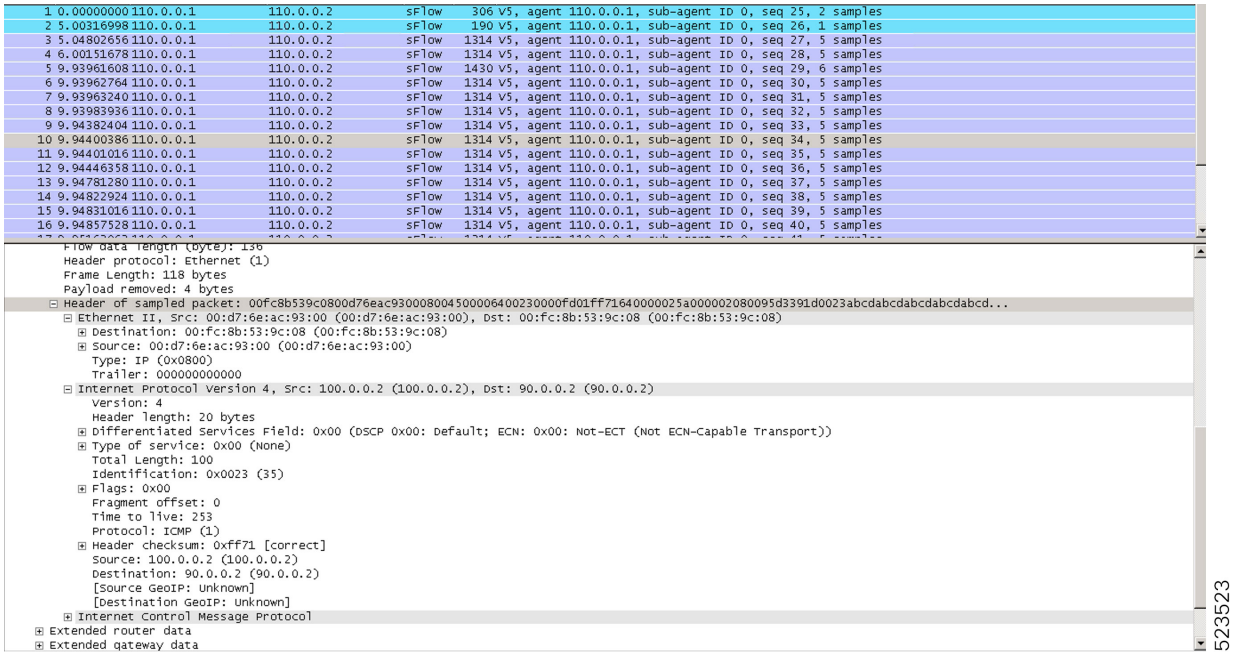
   - Export destination IP address (IPv4 or IPv6 address). The same packets can be exported to multiple IPv4 or IPv6 destinations.
   - DSCP value
   - Source interface and its IP address
   - Transport protocol
   - UDP port number, where the collector listens to the packets
   - Maximum datagram length
   - Don't Fragment bit (DF-bit). The DF-bit within the IP header is supported only on IPv4 transport and determines whether the router is allowed to fragment a packet.

   Verify the flow exporter configuration using the show flow exporter-map command.

```
Router# show flow exporter-map EXP-MAP
Flow Exporter Map : EXP-MAP
------------------------------------------------
Id                  : 1
Packet-Length       : 9000
DestinationIpAddr   : 192.127.0.1
VRFName             : default
SourceIfName        : HundredGigE 0/0/0/3
SourceIpAddr        : 192.127.10.1
DSCP                : 0
TransportProtocol   : UDP
TransportDestPort   : 6343

Export Version      : sFlow v5
```

   - The Export Version: sFlow v5 indicates that the exporter map configuration is successful.



The flow exporter map is configured and ready for use.

2. Configure the flow sampler using the sampler-map command to define the sampling rate for flow samples.

Example

```
Router(config)# sampler-map SAMP-MAP
Router(config-sm)# random 1 out-of 4096
```

Verify the sampler map configuration using the show sampler-map command.

```
Router# show sampler-map SAMP-MAP
Sampler Map : SAMP-MAP
-----------------------------------------------
Id:          1
Mode:        Random (1 out of 4096 Pkts)
```

- The sampler map configuration is successful with a sample rate of 1 out of every 4096 packets.

If...	Then...
If the sample rate is set to 1 out of 4096 packets	One sample is generated for every 4096 packets.
If the sampler map is not configured	No sampling occurs for sFlow.
If the sampler map configuration is successful	Sampling begins as per the defined rate.

The flow sampler is configured and ready for use.

3. Configure the flow monitor using the flow monitor-map command to define the type of traffic to be monitored and the polling frequency.

Example

```
Router(config)# flow monitor-map MON-MAP
Router(config-fmm)# record sflow
Router(config-fmm)# sflow options
Router(config-fmm-sflow)# extended-router
Router(config-fmm-sflow)# extended-gateway
Router(config-fmm-sflow)# if-counters polling-interval 120
Router(config-fmm-sflow)# input ifindex physical
Router(config-fmm-sflow)# output ifindex physical
Router(config-fmm-sflow)# sample-header size 200
Router(config-fmm-sflow)# exporter EXP-MAP
```

You can export input and output interface handles if the ingress or egress interface is a bundle or a Bridge-Group Virtual Interface (BVI).

Verify the monitor map configuration using the show flow monitor-map command.

```
Router# show flow monitor-map MON-MAP

Flow Monitor Map : MON-MAP
-----------------------------------------------
Id:                2
RecordMapName:     sflow
ExportMapName:     EXP-MAP
ExtendedRouter:    Enabled
ExtendedGateway:   Enabled
InterfaceCounters: Enabled
PollingInterval:   30 seconds
SampledHeaderSize: 200

Input ifhandle physical
Output ifhandle physical
```

- The monitor map is configured successfully with the associated exporter map to monitor the interface counters at a polling interval of 30 seconds.

If...	Then...
If the monitor map is configured with an exporter map	sFlow data is exported as per the polling interval.
If the polling interval is set to 30 seconds	Interface counters are monitored every 30 seconds.
If the monitor map is not associated with an exporter map	No sFlow data is exported.

The flow monitor is configured and ready for use.

4. Apply sFlow on an interface using the flow datalinkframesection command in global configuration mode.

Example

```
Router(config)# interface HundredGigE 0/0/0/3
Router(config)# ipv4 address 192.127.0.56 255.255.255.0
Router(config)# ipv6 address FFF2:8:DE::56/64
Router(config)# flow datalinkframesection monitor-map MON-MAP sampler SAMP-MAP
 ingress
```

Apply the monitor map MON-MAP and the sampler map SAMP-MAP on the HundredGigE 0/0/0/3 interface in the ingress direction to monitor incoming packets.

- Monitor incoming packets using sFlow on the specified interface.

If...	Then...
If the monitor and sampler maps are applied to the interface	sFlow monitoring is enabled for incoming packets.
If the interface is not configured	No sFlow monitoring occurs.
If the ingress direction is specified	Only incoming packets are monitored.

sFlow is applied to the interface for traffic monitoring.

5. Enable sFlow on the RP (0/RP0/CPU0) or on the line card using the hw-module profile netflow sflow-enable command.

Example

```
Router(config)# hw-module profile netflow sflow-enable location 0/0/CPU0
```

- Enable sFlow on the RP or line card to activate sFlow monitoring.

If...	Then...
If sFlow is enabled on the RP or line card	sFlow monitoring is activated for the specified location.
If the location is not specified	sFlow may not be enabled on the intended hardware.
If sFlow is not enabled	No sFlow monitoring occurs.

sFlow is enabled on the RP or line card.

6. Reload the line card using the hw-module reset auto command.

Example

```
Router# reload location 0/0/CPU0
```

- With this configuration, sFlow is enabled on the line card.

If...	Then...
If the line card is reloaded	sFlow configuration is applied and enabled.
If the reload command is not issued	sFlow may not be active on the line card.
If sFlow is enabled after reload	Traffic monitoring begins on the line card.

sFlow is enabled and active on the line card.

7. Verify the statistics of exported traffic flow at the producer and exporter using show flow platform producer statistics location and show flow exporter commands.

   Example

```
Router# show flow platform producer statistics location 0/0/CPU0
Netflow Platform Producer Counters:
IPv4 Ingress Packets:                          0
IPv4 Egress Packets:                           0
IPv6 Ingress Packets:                          0
IPv6 Egress Packets:                           0
MPLS Ingress Packets:                          0
MPLS Egress Packets:                           0
IPFIX315 Ingress Packets:                      0
IPFIX315 Egress Packets:                       0
sFlow Ingress Samples:                    100000
Drops (no space):                              0
Drops (other):                                 0
Unknown Ingress Packets:                       0
Unknown Egress Packets:                        0
Worker waiting:                                0
```

```
Router# show flow exporter EXP-MAP location 0/0/CPU0
Wed June 21 04:21:36.263 UTC
Flow Exporter: EXP-MAP
Export Protocol: sFlow v5
Flow Exporter memory usage: 5247776
Used by flow monitors: MON-MAP

Status: Normal
Transport:   UDP
Destination: 192.127.0.1     (6343) VRF default
Source:      192.127.10.1    (6331)
Flows exported:                                50245 (9631004 bytes)
Flows dropped:                                     0 (0 bytes)

Packets exported:                               7372 (19262008 bytes)
Packets dropped:                                   0 (0 bytes)

Total export over last interval of:
  1 hour:                                       7363 pkts
                                              9629960 bytes
                                                50236 flows
  1 minute:                                       12 pkts
                                                1392 bytes
                                                  12 flows
  1 second:                                        0 pkts
                                                   0 bytes
                                                   0 flows
```

- The sFlow configuration with flow and packet data is successful on the router.

If...	Then...
If the statistics show exported flows and packets	sFlow is successfully monitoring and exporting traffic data.
If no flows or packets are exported	Check the configuration for errors.
If drops are reported	Investigate for resource or configuration issues.

sFlow statistics confirm successful traffic monitoring and export.

---

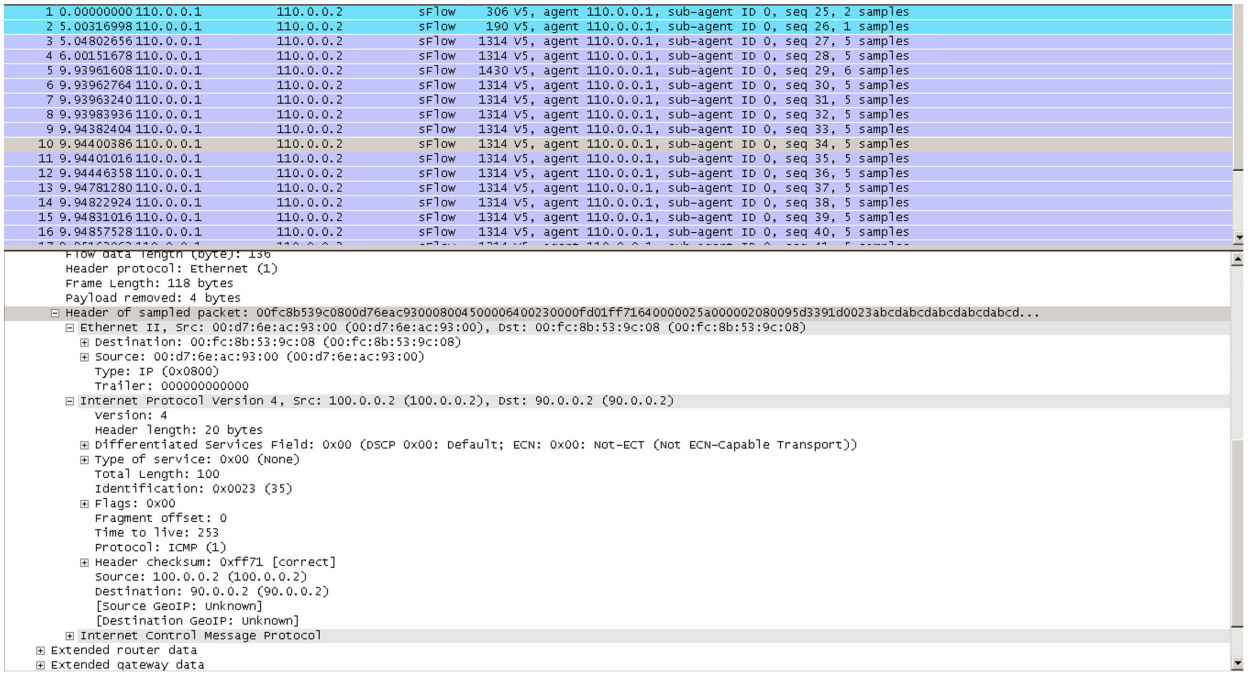
sFlow is configured on the router, enabling traffic monitoring and export of sampled data to the collector for analysis.

What to do next

Analyze the traffic using the UDP datagram and sampled data collected at the sFlow collector.

- Review the sampled data at the sFlow collector to validate traffic monitoring.

This image shows an example of sampled data that has been collected at the sFlow collector.



Feature history tables

Explains the release history and supported hardware for sFlow-related features.

A feature history table is a reference resource that

- provides details about system alerts such as BUFFER_SIZE_EXCEEDED, which signals that the flow monitor buffer has reached its capacity with sampled flow data due to a low export rate limit or a high sampling rate,
- includes BUFFER_EXCEEDING_STOPPED alerts that indicate when the flow monitor buffer returns to its regular state, and
- documents enhancements such as tunnel encapsulation, where the packet header supports an extended structure encompassing tunnel header information.

Feature history and hardware support for sFlow-related features

Provides a lookup table of sFlow-related features, their release history, and supported hardware platforms.

Table 16: Feature History Table

Feature Name	Release Introduced	Description
System alerts related to sFlow	Release 7.5.3	The following syslog notifications are available with sFlow: • FLOW_SAMPLES_DROPPED - This alert is seen whenever the buffer becomes full with sampled flow data, either due to a high sampling rate or an increase in the traffic rate. • FLOW_SAMPLES_DROPPING_STOPPED - This alert is seen when the buffer reverts to its regular state. • BUFFER_SIZE_EXCEEDED - This alert signals that the flow monitor buffer has reached its capacity with sampled flow data, which could be a result of a low export rate limit or a high sampling rate. • BUFFER_EXCEEDING_STOPPED - This alert is seen when the flow monitor buffer reverts to its regular state.
Ingress sFlow enhancements	Release 7.3.3	The incoming sFlow packet offers the following enhancements to improve scalability and decrease the volume of packets received: • Expansion of sFlow datagram size—from 1500B to 9KB • Tunnel encapsulation—The packet header now supports an extended structure encompassing tunnel header information. The egress packet extracts the tunnel information during decapsulation. • sFlow collector indicates discarded packets and locally targeted packets at the output interfaces, in a specific format along with drop value.
sFlow for L2 interfaces	Release 7.3.1	Ingress sFlow on an L2 interface is introduced. Support for sFlow existed in earlier releases.
Sampled flow	Release 7.2.12	Sampled flow (sFlow) allows you to monitor real-time traffic in data networks. It uses sampling mechanism in the sFlow agent software to monitor traffic and to forward the sample data to the central data collector.

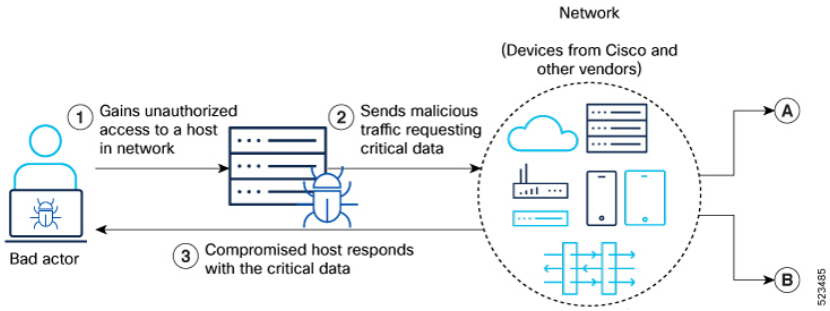
7 Use Case: NetFlow and sFlow in Action

Topics:

Let's explore a use case and learn how to deploy NetFlow and sFlow protocols to monitor and curb malicious attack on the network via a use case.

Here's a hypothetical use case illustrating a bad actor (attacker) sending malicious traffic and the network getting compromised:

Figure 7: Malicious activity in a network



rect 683, 71, 728, 116 Scenario A: Traffic Monitoring Without NetFlow and sFlow
rect 689, 223, 721, 263 Scenario B: Traffic Monitoring With NetFlow and sFlow

- Attack entry point–An Enterprise becomes the target of a cyber attack. The attacker employs various tactics to gain unauthorized initial access to the network.
- Generate malicious traffic– After the attacker identifies vulnerable devices as potential targets, they compromise a host and start generating malicious traffic and potentially launch DDoS attacks, to steal sensitive data, or take control of the network using these compromised machines as a platform.
- Breach data–The malicious traffic triggers a series of attacks within the network. With access to sensitive data, the attacker attempts retrieving critical data from the compromised network.

With this context, lets explore these two scenarios:

traffic monitoring scenarios without NetFlow and sFlow

Explains the consequences of not implementing NetFlow or sFlow for network traffic monitoring in an enterprise environment.

A traffic monitoring scenario without NetFlow and sFlow is a network feature that:

- lacks implementation of network traffic monitoring protocols such as NetFlow or sFlow,
- results in undetected malicious activity due to insufficient visibility, and
- increases the risk of data breaches and loss of trust.

Consequences of not using NetFlow or sFlow for traffic monitoring

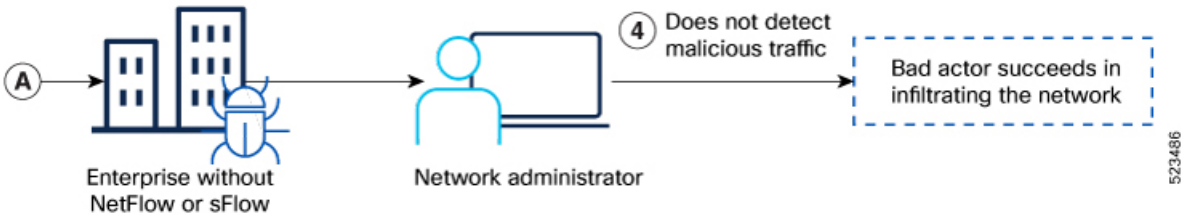
Describes the impact of not implementing NetFlow or sFlow for network traffic monitoring in an enterprise environment, including security risks and loss of visibility.

Without NetFlow or sFlow, the enterprise lacks network traffic monitoring protocols, resulting in:

- Bypassed threat detection and response—the network administrator does not detect any unusual network patterns or intrusions immediately after the attack.
- Successful data breach—the network is compromised by malicious traffic that goes undetected, leading to loss of critical data and trust.

The overall network security posture is compromised due to the lack of traffic monitoring mechanisms, resulting in poor visibility of the network and its functions.

Figure 8: Traffic monitoring without Flow data to identify malicious activity



Example of traffic monitoring scenario without NetFlow and sFlow

In an enterprise network without NetFlow or sFlow, a malicious actor initiates an attack. The network administrator does not detect any unusual network patterns or intrusions immediately after the attack, resulting in a successful data breach and loss of critical data and trust.

Counter-example: traffic monitoring scenario with NetFlow or sFlow

In an enterprise network with NetFlow or sFlow implemented, the network administrator can detect unusual network patterns or intrusions, enabling timely response to threats and preventing data breaches.

Analogy for traffic monitoring without NetFlow and sFlow

Not using NetFlow or sFlow for traffic monitoring is like driving a car without a dashboard—critical information about the vehicle's status is missing, increasing the risk of accidents and breakdowns.

Traffic monitoring protocols with NetFlow and sFlow

Explains how NetFlow and sFlow protocols enable proactive network traffic monitoring and threat detection.

A traffic monitoring protocol with NetFlow and sFlow is a network protocol that

- collects and retains flow records of transmitted traffic using technologies such as NetFlow or sFlow
- enables analysis of flow data to identify patterns and anomalies that may indicate security threats, and
- facilitates prompt detection and response to malicious activity by generating alerts and supporting investigation.

Traffic monitoring protocol attributes and workflow

Describes how NetFlow and sFlow protocols collect, analyze, and respond to network traffic for threat detection.

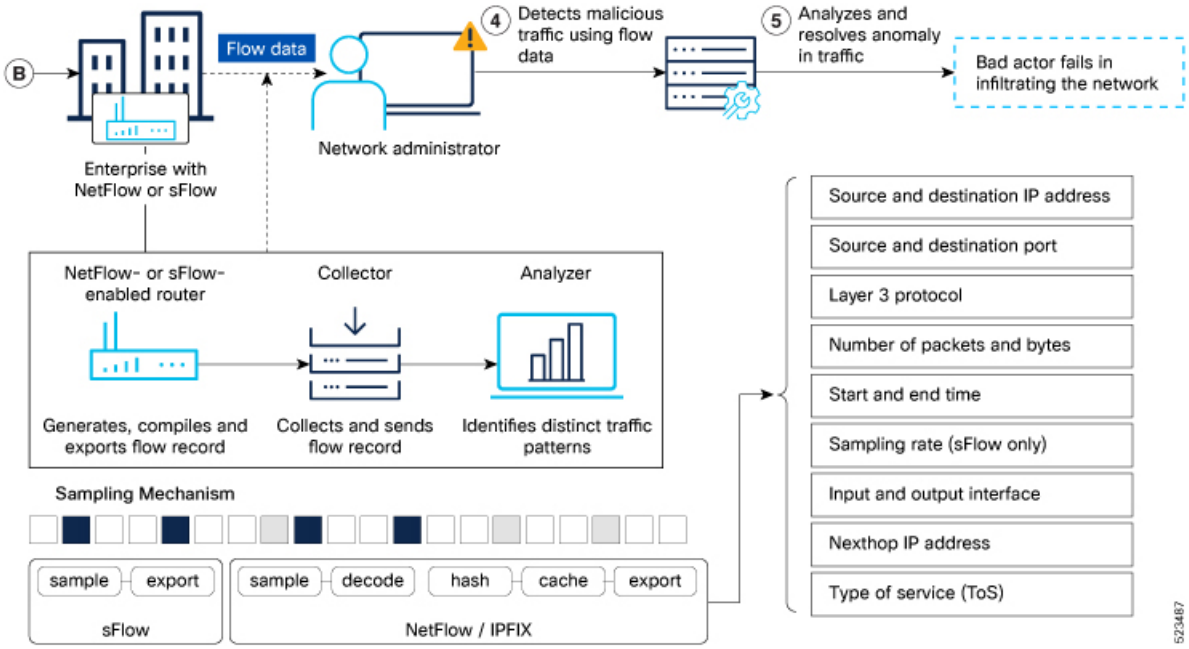
- collects and retains flow records of transmitted traffic using technologies such as NetFlow or sFlow
- enables analysis of flow data to identify patterns and anomalies that may indicate security threats, and
- facilitates prompt detection and response to malicious activity by generating alerts and supporting investigation.

In this scenario, the enterprise implements network traffic monitoring protocols such as NetFlow or sFlow to enhance security posture. The process involves several key stages:

- Flow data collection: Routers enabled with NetFlow or sFlow capture and retain flow records of transmitted traffic. These records store essential metadata related to the traffic's journey, including source and destination domains, the count and volume of inbound and outbound packets, and timestamps. The recorded flow records are then sent to a designated collector.
- Data analysis: A NetFlow or sFlow analyzer or security monitoring tool processes and analyzes the collected data. The tool identifies patterns and anomalies that may indicate a security threat, such as unusual traffic patterns, unexpected communication between hosts, or a high volume of traffic from suspicious sources.
- Threat detection: The analyzer applies algorithms and rules to detect potential threats based on the analyzed data. It compares network traffic with predefined security policies. If a potential threat is detected, the analyzer generates an alert, which can be sent to the network administrator for further investigation.
- Prompt investigation and responsive action: Upon receiving the alert, the network administrator investigates the identified threat. They analyze additional logs, inspect packet captures, or perform other security measures to gather more information. Once the threat is confirmed, appropriate actions are taken to mitigate the impact, such as blocking malicious IP addresses and isolating affected hosts.

By leveraging NetFlow and sFlow for threat identification, organizations can proactively detect and respond to security threats, allowing for early threat detection and faster incident response, ultimately reducing the risk of a successful attack.

Figure 9: Traffic monitoring workflow with Flow data to identify malicious activity


B
Flow data
4
Detects malicious traffic using flow data
5
Analyzes and resolves anomaly in traffic
Bad actor fails in infiltrating the network
Network administrator
Enterprise with NetFlow or sFlow
NetFlow- or sFlow-enabled router
Collector
Analyzer
Generates, compiles and exports flow record
Collects and sends flow record
Identifies distinct traffic patterns
Sampling Mechanism
sample
export
sFlow
sample
decode
hash
cache
export
NetFlow / IPFIX
Source and destination IP address
Source and destination port
Layer 3 protocol
Number of packets and bytes
Start and end time
Sampling rate (sFlow only)
Input and output interface
Nexthop IP address
Type of service (ToS)
523487

8 Configure YANG data models for NetFlow and sFlow

Topics:

Configure YANG data models to manage NetFlow and sFlow on Cisco 8000 and Cisco ASR 9000 Series Routers.

Use YANG data models to configure and retrieve the operational status of NetFlow and sFlow on Cisco 8000 and Cisco ASR 9000 Series Routers.

- Enable programmatic management of NetFlow and sFlow features.
- Support both CLI and YANG-based configuration methods.

YANG data models provide a programmatic interface for configuring and monitoring NetFlow and sFlow features on Cisco 8000 and Cisco ASR 9000 Series Routers.

Cisco IOS XR supports both traditional CLI and YANG-based configuration for NetFlow and sFlow. Refer to the official documentation for guidance on accessing and using these models.

- Consult Cisco IOS XR documentation for supported YANG models and operational guidance.

Before you begin

Ensure you have access to Cisco 8000 or Cisco ASR 9000 Series Routers running IOS XR with NetFlow and sFlow features enabled.

- Verify that you have administrative privileges to configure YANG data models.

Follow these steps to configure YANG data models for NetFlow and sFlow.

Procedure

Access the router using a supported management interface (CLI or NETCONF).

Example

```
Refer to the Cisco IOS XR documentation for
NETCONF/YANG access commands and configuration
examples.
```

YANG data models allow you to configure NetFlow and sFlow programmatically, providing flexibility and automation for network management.

- Use YANG models to retrieve operational status and statistics for NetFlow and sFlow.

If...	Then...
You use CLI for configuration	Configure NetFlow and sFlow using traditional CLI commands.
You use YANG data models	Configure NetFlow and sFlow programmatically using NETCONF/YANG.

If...	Then...
You need operational status	Retrieve status and statistics using YANG operational data.

NetFlow and sFlow are configured and operational status can be retrieved using the selected method.

NetFlow and sFlow are successfully managed using YANG data models or CLI, enabling programmatic configuration and monitoring on Cisco 8000 and Cisco ASR 9000 Series Routers.

What to do next

Verify the configuration and monitor NetFlow and sFlow operational status using YANG or CLI as needed.

- Review logs and statistics to ensure proper operation.

List of YANG data models for NetFlow and sFlow

Lists the YANG data models available for configuring and managing NetFlow and sFlow on Cisco routers.

Lists the YANG data models available for configuring and managing NetFlow and sFlow on Cisco routers.

The following YANG data models are available for configuring and managing NetFlow and sFlow on Cisco routers:

- Cisco IOS XR Native Data Models: Cisco-IOS-XR-traffmon-netflow-cfg, Cisco-IOS-XR-ofa-netflow-oper
- Unified Data Models: Cisco-IOS-XR-um-8000-hw-module-profile-netflow-cfg, Cisco-IOS-XR-um-flow-cfg
- OpenConfig Data Model: openconfig-sampling-sflow.yang

Table 17: NetFlow and sFlow YANG Data Models

Cisco IOS XR Native Data Model	Unified Data Model	OpenConfig Data Model
Cisco-IOS-XR-traffmon-netflow-cfg	Cisco-IOS-XR-um-8000-hw-module-profile-netflow-cfg	openconfig-sampling-sflow.yang
Cisco-IOS-XR-ofa-netflow-oper	Cisco-IOS-XR-um-flow-cfg	

Note

We recommend using unified data models over native data models.

You can access the data models using one of these options:

Access data models

Explains the available options for accessing data models and describes their configuration context.

An access data model is a configuration mode that

- describes the supported behavior and operating context
- identifies relevant configuration, support, or operational details, and
- helps users understand when and how to apply the feature.

Options for accessing data models

Describes the available options for accessing data models and their organizational structure.

A data model is a structured representation that defines how data is organized, accessed, and managed in a system.

You can access data models using one of these options:

- Directly through the system interface
- By using APIs provided by the platform
- Through downloadable model files for offline analysis

Access data models from the router

Configure NETCONF and retrieve YANG data model capabilities from the router.

Accessing data models from the router enables remote configuration and management using NETCONF and YANG, allowing you to view supported data models for your software version.

- Facilitates remote device management and configuration.
- Allows you to retrieve and examine YANG data model capabilities.

You can configure the router to provide access to YANG data models using the NETCONF protocol. This enables remote configuration and management of the device and allows you to view supported data models for your software version.

NETCONF provides a standardized interface for interacting with YANG data models on Cisco routers.

- Use NETCONF to automate device configuration and retrieve model information.

Procedure

1. Enter global configuration mode.

   Example

   ```
   Router# configure
   ```

   Global configuration mode allows you to make changes to the router's settings.

   - Ensure you have the necessary privileges to access configuration mode.

If...	Then...
You are not in privileged EXEC mode	Enter 'enable' to access privileged EXEC mode before configuring.
You do not have configuration privileges	Contact your administrator for access rights.
You are already in global configuration mode	Proceed to the next step.

   You are now in global configuration mode and can configure NETCONF.

2. Configure the NETCONF network management protocol to enable remote configuration and management using YANG data models.

   Example

   ```
   Router(config)# netconf-yang agent ssh
   ```

   This command enables NETCONF over SSH, allowing secure remote access to YANG data models.

   - Ensure SSH is enabled and accessible on the router.

If...	Then...
SSH is not enabled	Enable SSH before configuring NETCONF.

If...	Then...
NETCONF is already enabled	Verify configuration or proceed to commit changes.
Configuration fails	Check for syntax errors or missing prerequisites.

NETCONF is enabled for remote management using YANG data models.

3. Commit the configuration.

Example

```
Router(config)# commit
```

Committing saves the configuration changes to the router.

- Ensure all changes are correct before committing.

If...	Then...
Configuration is valid	Changes are applied successfully.
Configuration is invalid	Review error messages and correct configuration.
Commit fails	Check for system errors or insufficient privileges.

Configuration changes are committed to the router.

4. Establish a NETCONF session with the device and retrieve the capabilities information.

Example

```
Router# show netconf-yang capabilities
Tue Sep 19 22:03:26.305 UTC
[Netconf capabilities]

   D: Has deviations

 Capability
      | Revision |D
-------------------------------------------------------------------------------+---------+-
urn:ietf:params:netconf:base:1.1
      | -        |
urn:ietf:params:netconf:capability:candidate:1.0
      | -        |
urn:ietf:params:netconf:capability:confirmed-commit:1.1
      | -        |
urn:ietf:params:netconf:capability:interleave:1.0
      | -        |
urn:ietf:params:netconf:capability:notification:1.0
      | -        |
urn:ietf:params:netconf:capability:rollback-on-error:1.0
      | -        |
urn:ietf:params:netconf:capability:validate:1.1
      | -        |
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-fib-platform-cfg
      |2019-04-05|
```

```
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-lpts-oper
      |2022-05-05|
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-platforms-npu-resources-oper
      |2020-10-07|
http://cisco.com/ns/yang/Cisco-IOS-XR-8000-qos-oper
      |2021-06-28|
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-act
      |2021-03-22|
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-cfg
      |2022-07-13|
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-datatypes
      |2021-10-06|
http://cisco.com/ns/yang/Cisco-IOS-XR-Ethernet-SPAN-oper
      |2022-09-05|
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-aaacore-cfg
      |2019-04-05|
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-ldapd-cfg
      |2022-06-22|
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-ldapd-oper
      |2022-05-20|
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-lib-cfg
      |2020-10-22|
http://cisco.com/ns/yang/Cisco-IOS-XR-aaa-lib-datatypes
---------------------------------   Truncated for brevity
-------------------------------------
```

By examining the capabilities, you can view the available data models for the software version installed on the router.

- Use the output to determine which YANG models are supported.

If...	Then...
The capabilities list includes desired YANG models	You can proceed with configuration using those models.
Some YANG models are missing	Check software version or contact Cisco support for model availability.
Capabilities output is empty	Verify NETCONF configuration and session status.

You have retrieved the NETCONF YANG capabilities and can view supported data models.

Once completed, NETCONF is enabled and you can remotely access and view YANG data model capabilities for your router's software version.

What to do next

Review the retrieved YANG data models to ensure compatibility with your intended configuration and management tasks.

- Update device configuration as needed based on available YANG models.

Access data models from Cisco Feature Navigator

Access data models from Cisco Feature Navigator to view, filter, and download YANG models for Cisco IOS XR releases.

Access and download YANG data models for Cisco IOS XR releases using Cisco Feature Navigator.

- View, filter, and export YANG data models for automation and integration tasks.

Cisco Feature Navigator provides regularly updated YANG data models for Cisco IOS XR releases. The platform allows you to search, filter, and download models for automation and integration. If you encounter issues or have suggestions, use the feedback link provided on the site.

Data models are organized by type: Cisco XR native models, Unified models, and OpenConfig models. The visual tree structure helps you understand relationships between modules, containers, leaves, and leaf lists.

- Use filters to narrow down data model definitions by platform, release, and node status (deprecated, obsolete, unsupported).

Before you begin

Ensure you have access to Cisco Feature Navigator. A Cisco.com account is optional but allows for additional features.

- Optional: Cisco.com account credentials for enhanced access.

Follow these steps to access data models from Cisco Feature Navigator.

Procedure

1. Go to Cisco Feature Navigator.

   Example

   ```
   N/A
   ```

   N/A

If...	Then...
You have a Cisco.com account	Log in with your credentials for enhanced access.
You do not have a Cisco.com account	Continue as guest.
N/A	N/A

   N/A

2. If you have a Cisco.com account, log in with your credentials. If you do not have an account, continue as guest.

   Example

   ```
   N/A
   ```

   You are directed to the Cisco Feature Navigator main page.

If...	Then...
You log in with Cisco.com account	You access additional features.
You continue as guest	You access basic features.
N/A	N/A

   N/A

3. Select **YANG Data Models**.

Example

```
N/A
```

N/A

If...	Then...
N/A	N/A
N/A	N/A
N/A	N/A

N/A

4. Select the **Product** and **Cisco IOS XR Release** based on your requirement.

Example

```
N/A
```

The data models are listed by type: Cisco XR native models, Unified models, and OpenConfig models.

- Use the search field to find a specific data model.

If...	Then...
You select product and release	Data models are listed by type.
You use search field	You find specific data model.
N/A	N/A

N/A

5. Click the data model of interest to view more details.

Example

```
N/A
```

The data model displays in a hierarchical tree structure, making it easier to navigate and understand the relationships between YANG modules, containers, leaves, and leaf lists. Apply filters to narrow down the data model definitions for the selected platform and release based on status such as deprecated, obsolete, and unsupported nodes.

- Click the Download icon to export the data model information in Excel format.
- This visual tree form helps you get insights into the nodes you can use to automate your network.
- The data models on Cisco Feature Navigator are regularly updated based on IOS XR release. If you encounter any problems or have suggestions for improvements, share your experience using the Send us your feedback link.

If...	Then...
You click Download icon	Data model information is exported in Excel format.

If...	Then...
You use filters	Data model definitions are narrowed by platform, release, and node status.
You use feedback link	You can share your experience or suggestions.

N/A

You can view, filter, and download YANG data models for Cisco IOS XR releases from Cisco Feature Navigator.

What to do next

N/A

Access data models from GitHub

Access data models from the GitHub repository by browsing release folders and downloading YANG files as needed.

Access Cisco IOS XR data models from the GitHub repository by browsing release folders and downloading YANG files as needed.

- Obtain the latest and historical data models for IOS XR releases.
- Download YANG files for specific releases and features.
- Contribute to the repository or review deviations and supported sensor paths.

The GitHub repository for Cisco IOS XR data models provides folders organized by release, allowing users to access, review, and download YANG files for each supported release.

The repository is regularly updated and includes documentation for supported and unsupported sensor paths, as well as opportunities for community contributions.

- Use the repository to find data models for specific IOS XR releases.
- Review the Available-Content.md file for a comprehensive list of supported data models.

Before you begin

Ensure you have internet access and a web browser to access the GitHub repository.

- Optional: Create a GitHub account if you plan to contribute or open issues.

Follow these steps to access data models from GitHub.

Procedure

1. Go to the GitHub repository for Cisco IOS XR data models.

   Example

   ```
   N/A
   ```

   The repository page lists folders based on IOS XR releases.

   - Each folder corresponds to a specific IOS XR release version.

If...	Then...
You want data models for a specific IOS XR release	Navigate to the corresponding release folder.
You need to review supported sensor paths	Check the Available-Content.md file or relevant YANG files.
You want to contribute or report an issue	Submit a pull request or open an issue in the repository.

The repository page displays folders for each IOS XR release.

2. Navigate to the release folder of interest to view the list of supported data models and their definitions. For example, to access data models for IOS XR release 7.10.1, click the folder named `7.10.1`.

Example

```
N/A
```

Each folder contains YANG files representing different data models.

- Folders are named according to IOS XR release versions.
- YANG files define models or augment existing models with additional XPaths.

If...	Then...
You want to see all supported data models for a release	Review the Available-Content.md file in the folder.
You need details about supported sensor paths	Check the `openconfig-acl.yang` file.
You want to see unsupported sensor paths	Review the `cisco-xr-openconfig-acl-deviations.yang` file.

The release folder displays YANG files and documentation for supported and unsupported sensor paths.

3. Click the YANG file you want to access to view its contents.

Example

```
N/A
```

You can click the **Raw** button to see the raw code or use the **Download** button to download the file to your computer.

- Each data model defines a complete and cohesive model or augments an existing data model with additional XPaths.
- Unsupported sensor paths are documented as deviations.

If...	Then...
You want to view the contents of a YANG file	Click the file and use the Raw or Download button.
You need to review deviations for unsupported sensor paths	Open the `cisco-xr-openconfig-acl-deviations.yang` file.

If...	Then...
You want to see a comprehensive list of supported data models	Navigate to the Available-Content.md file in the repository.

The YANG file contents are displayed, and you can download the file as needed.

4. Repeat these steps for other versions or data models of interest.

Example

```
N/A
```

The GitHub repository for IOS XR data models is regularly updated based on release. You can contribute to the repository by submitting pull requests or opening issues if you encounter problems or have suggestions for improvements.

- Repository updates may include new releases, features, or corrections.
- Community contributions help improve the repository and documentation.

If...	Then...
You want to access data models for another release	Navigate to the desired release folder and repeat the steps.
You want to contribute or report an issue	Submit a pull request or open an issue in the repository.
You need to stay updated on new releases	Monitor the repository for updates and new folders.

You can access and download data models for multiple IOS XR releases and contribute to the repository as needed.

You can access, review, and download Cisco IOS XR data models from the GitHub repository for any supported release. The repository provides comprehensive documentation and opportunities for community contributions.

What to do next

Monitor the repository for updates and new releases. Contribute to the repository or open issues as needed.

- Review downloaded YANG files for compatibility with your IOS XR deployment.

IOS XR YANG data models

Explains the IOS XR YANG data models that enable programmatic management and remote configuration of network devices.

A IOS XR YANG data model is a configuration framework that

- enables programmatic management of network devices,
- supports remote configuration using protocols such as NETCONF or gRPC, and
- allows automation and validation of device configurations.

Key attributes and configuration steps for IOS XR YANG data models

Describes the IOS XR YANG data models, their key attributes, and the steps to programmatically configure a router using these models.

IOS XR YANG data models provide a network configuration framework that:

- enables programmatic management of network devices,
- supports remote configuration using protocols such as NETCONF or gRPC, and
- allows automation and validation of device configurations.

To programmatically configure your router using IOS XR YANG data models, follow these steps:

1. Enable a network management protocol to manage the router remotely using protocols such as NETCONF or gRPC.
2. Install the necessary libraries and tools. Depending on the programming language, you may need to install libraries or tools to interact with the router. For example, if you use Python, install the `ncclient` library.
3. Establish a session with the router using your chosen programming language and protocol (NETCONF or gRPC). Provide connection parameters such as device IP address, username, password, and port number.
4. Retrieve the router capabilities to view the supported features and functionalities available on the router.
5. Create or modify configurations using YANG data models to update the router's configuration.
6. Apply the configuration by pushing the updated configuration via NETCONF or gRPC to modify the router's running configuration.
7. Validate the configuration by verifying that the changes are successfully applied. Retrieve the running configuration or specific parameters to ensure the device is configured as intended.

For detailed instructions about using the data models, refer to the *Programmability Configuration Guide for Cisco 8000 Series Routers*.

9 Configure and verify NetFlow and sFlow using CLI commands

Topics:

Lists the CLI commands for configuring and verifying NetFlow and sFlow protocols for traffic monitoring.

Configure and verify NetFlow and sFlow protocols for traffic monitoring using command-line interface (CLI) commands.

- Enable traffic monitoring and analysis on Cisco IOS XR devices.
- Use CLI commands to configure NetFlow and sFlow features.
- Verify protocol operation and data collection.

The Cisco Command Reference Guide provides a comprehensive list of command-line interface (CLI) commands for configuring and verifying NetFlow and sFlow implementations.

Cisco IOS XR supports configuring NetFlow and sFlow using both traditional CLI commands and programmatically through YANG data models.

- NetFlow and sFlow are used for traffic monitoring and network analysis.
- CLI commands allow manual configuration and verification of these protocols.

Before you begin

Ensure you have access to a Cisco IOS XR device with administrative privileges.

- Verify that NetFlow and sFlow features are supported on your device and software version.
- Confirm that you have the necessary credentials to access the CLI.

Follow these steps to configure and verify NetFlow and sFlow using CLI commands.

Procedure

1. Enter the CLI on your Cisco IOS XR device.

   Example

   ```
   router#
   ```

   Access the device using SSH, console, or Telnet as appropriate.

   - Ensure you are in privileged EXEC mode.

If...	Then...
You do not have administrative privileges	Request access from your network administrator.
NetFlow or sFlow features are not supported	Upgrade your device or software to a supported version.

If...	Then...
CLI access is unavailable	Use YANG models or another supported configuration method.

- Configure NetFlow using CLI commands.
- Configure sFlow using CLI commands.

You are ready to configure NetFlow and sFlow protocols for traffic monitoring.

2. Configure NetFlow protocol using CLI commands.

Example

```
router(config)# flow exporter EXPORTER_NAME
router(config-flow-exporter)# destination
<IP_ADDRESS>
router(config-flow-exporter)# transport udp <PORT>
router(config-flow-exporter)# exit
router(config)# flow monitor MONITOR_NAME
router(config-flow-monitor)# record netflow ipv4
router(config-flow-monitor)# exporter EXPORTER_NAME
router(config-flow-monitor)# exit
router(config)# interface <INTERFACE>
router(config-if)# flow monitor MONITOR_NAME
ingress
```

Replace EXPORTER_NAME, MONITOR_NAME, IP_ADDRESS, PORT, and INTERFACE with your specific values.

- Configure flow exporter and monitor for NetFlow data collection.
- Apply the monitor to the desired interface.

If...	Then...
Exporter configuration fails	Verify IP address and port settings.
Monitor is not applied to interface	Check interface name and monitor configuration.
NetFlow data is not collected	Review exporter and monitor settings for errors.

NetFlow protocol is configured for traffic monitoring.

3. Configure sFlow protocol using CLI commands.

Example

```
router(config)# sflow agent <INTERFACE>
router(config)# sflow collector <IP_ADDRESS> <PORT>
router(config)# sflow sampling-rate <RATE>
router(config)# sflow enable
```

Replace INTERFACE, IP_ADDRESS, PORT, and RATE with your specific values.

- Configure sFlow agent, collector, and sampling rate.

- Enable sFlow protocol for traffic monitoring.

If...	Then...
sFlow agent configuration fails	Verify interface and collector settings.
Sampling rate is incorrect	Adjust sampling rate to match network requirements.
sFlow data is not collected	Review agent and collector configuration for errors.

sFlow protocol is configured for traffic monitoring.

4. Verify NetFlow and sFlow configuration using CLI commands.

Example

```
router# show flow monitor MONITOR_NAME
router# show sflow
```

Use the show commands to verify NetFlow and sFlow operation.

- Check NetFlow monitor statistics and exported data.
- Review sFlow agent and collector status.

If...	Then...
Monitor statistics are missing	Verify monitor and exporter configuration.
sFlow collector status is inactive	Check collector IP address and port settings.
Exported data is not received	Review network connectivity and configuration.

NetFlow and sFlow operation is verified.

NetFlow and sFlow protocols are configured and verified for traffic monitoring on Cisco IOS XR devices.

What to do next

Review collected traffic data and adjust configuration as needed for optimal monitoring.

- Monitor network performance and update exporter or collector settings if required.

NetFlow command references

Lists the available resources for NetFlow command-line interface (CLI) commands and where to find detailed command information.

A NetFlow command reference is a documentation resource that

- provides a comprehensive catalog of CLI commands for NetFlow configuration and verification,
- offers detailed descriptions and usage guidelines for each command, and
- enables users to efficiently manage and troubleshoot NetFlow features on Cisco devices.

Available NetFlow command resources and usage

Provides the available resources for NetFlow command-line interface (CLI) commands and where to find detailed command information.

- NetFlow command references provide a comprehensive catalog of CLI commands for NetFlow configuration and verification.
- They offer detailed descriptions and usage guidelines for each command.
- They enable users to efficiently manage and troubleshoot NetFlow features on Cisco devices.

For a complete list of supported NetFlow commands and their descriptions, access the Cisco Command Reference Guide at *NetFlow Command Reference for Cisco 8000 Series Routers* .