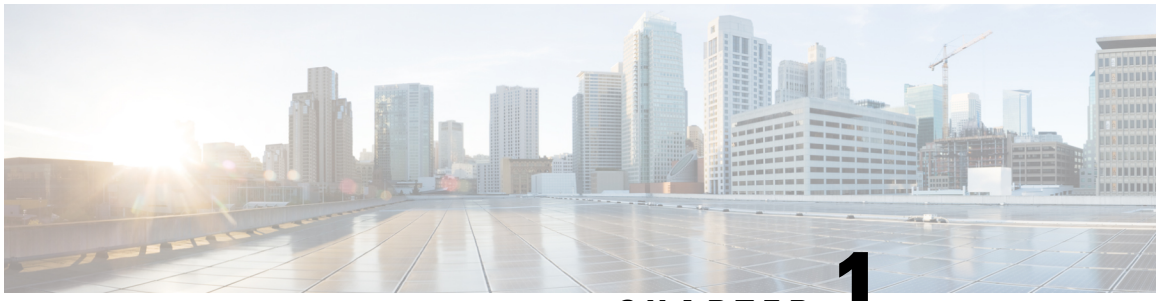
NetFlow and sFlow Configuration on Cisco NCS 5500 Series Routers, Cisco IOS XR Releases

First Published: 2024-03-15

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

About NetFlow and sFlow Documentation

Introducing the updated , Cisco IOS XR Releases.

Release-agnostic Document

This document for the Series Router features a single version that will be consistently kept up to date with the latest features and releases. Our goal is to make it easier for you to bookmark a single link and find one comprehensive version of the Netflow and sFlow Configuration for the Series Routers, rather than sift through multiple versions that are specific to each IOS XR release.

Where to begin

To get started with the NetFlow and sFlow implementation, we recommend referring to the Feature, Release, and Platform Matrix for NetFlow and sFlow. This section offers insights into NetFlow and sFlow features introduced not only for the Series Routers but also for all other IOS XR Routing platforms. Our aim is to simplify your access to our documentation and help you develop a comprehensive understanding.

- Releases Supported, on page 1

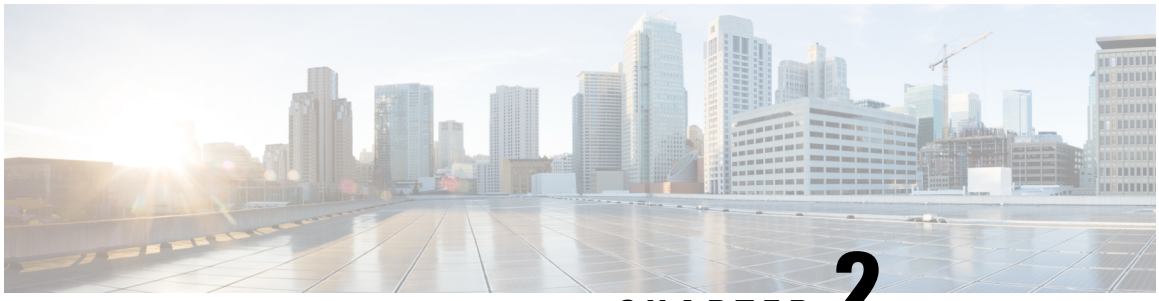
Releases Supported

The document is relevant for the following releases:

- 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 here have reached the End of Extended SW Maintenance Date. To access the notification, please visit the End-of-Life and End-of-Sale Notices page.



CHAPTER 2

Features Introduced in Cisco IOS XR Products and Releases

This table summarizes the features enhanced and introduced for NetFlow and sFlow.

- Feature, Release, and Platform Matrix for NetFlow and sFlow, on page 3

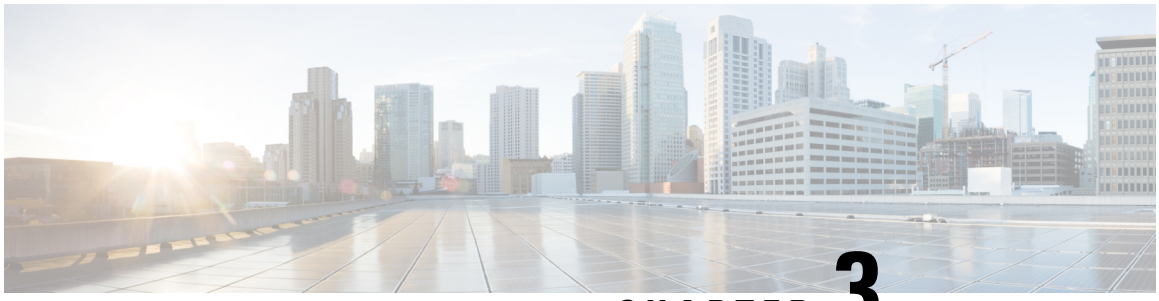
Feature, Release, and Platform Matrix for NetFlow and sFlow

Use this ready reckoner to locate features you're interested in and map their availability across platforms and releases.

Table 1: Feature, Release, and Platform Matrix

Feature	Cisco NCS 5500 Series	Other Routing Platforms
BGP Monitoring using IPFIX in MPLS Records	-	8000, R24.1.1
Monitoring Post-QoS Data in NetFlow and IPFIX	-	8000, R24.1.1
BGP Monitoring using IPFIX in MPLS Records	NCS 5500, R24.1.1	-
Simultaneous L2 and L3 Flow Monitoring using IPFIX	NCS 5500, R7.10.1	• ASR 9000, R7.10.1 • NCS 540, R7.10.1 • NCS 560, R7.10.1
sFlow Agent Address Assignment	NCS 5500, R7.10.1	ASR 9000, R7.10.1
IPFIX Enablement for SRv6 and Services over SRv6 Core	NCS 5500, R7.8.1	• NCS 560, R7.10.1 • ASR 9000, R7.10.1 • NCS 540, R7.8.1
System Alerts Related to sFlow	NCS 5500, R7.3.4	8000, R7.5.4

Feature	Cisco NCS 5500 Series	Other Routing Platforms
Enhanced NetFlow Sampling Rate of 1:2048 (2K)	NCS 5500, R7.4.1	-
MPLS top label type 4 for BGP Labeled Unicast traffic	NCS 5500, R7.4.1	NCS 540, R7.4.1
Tunnel Encapsulation and Increased sFlow datagram size	NCS 5500, R7.3.4	-
Increased sFlow Sample-Header Size	-	8000, R7.3.4
Ingress sFlow Enhancements	-	8000, R7.3.3
sFlow for L2 Interfaces	-	8000, R7.3.1
Flow Filter on Cisco NC57 Line Cards	NCS 5500, R7.2.2	-
Sampled Flow	NCS 5500, R7.5.1	• ASR 9000, R7.5.1 • NCS 540, R7.5.1 • 8000, R7.2.12
IPFIX Flow Record Enhancements for L2 and L3 traffic.	-	• 8000, R7.2.12 • ASR 9000, R7.4.1



CHAPTER 3

What are NetFlow and sFlow Protocols?

NetFlow and sFlow are both network monitoring technologies that provide insights into network traffic and performance.

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.

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

Benefits of NetFlow and sFlow

Network monitoring and traffic analysis offer insights into traffic, and help you understand network behavior.

Monitoring Network Applications and Use

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.

Network Planning

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 aspectssuch 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.

Security Analysis

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.

Billing and Accounting

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

Traffic Engineering

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

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

Flow Exporter

The flow exporter, also referred to as an exporter map, functions as a device tasked with collecting data regarding network flows. It transfers the compiled flow records to the designated collector. The exporter is responsibile for inspecting packets, identifying flows, and exporting flow-related data. The exporter map can transmit flow reports to a single destination. A maximum of 8 exporters are permitted per MAP configuration.

Contained within a flow exporter are particulars outlining network specifications and transport layer attributes related to the packets. These packets are exported to the collector through the utilization of the User Datagram Protocol (UDP) transport protocol. In cases where the source interface does not have an assigned IP address, the packet exporter remains inactive.

Flow Monitor

A flow monitor, also referred to as a Monitor map, serves the purpose of facilitating active traffic monitoring on a pre-configured interface. After the flow monitor is commited to an interface, a corresponding flow monitor cache is generated. This cache is used to collect traffic data based on both key and non-key fields outlined within the configured record.

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.

Furthermore, the option to include extended details such as router-specific elements like nexthop, source and destination mask lengths, and extended gateways attributes, including nexthop, communities, local preference, and AS (source AS, source peer AS, and destination AS path) information.

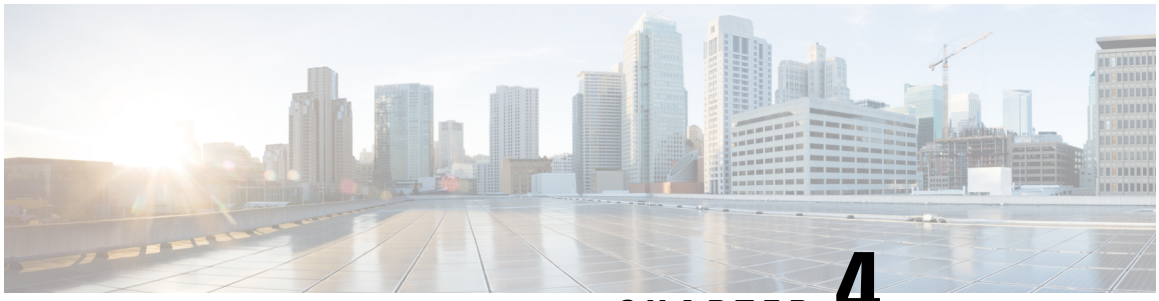
Flow Sampler

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

- The maximum supported sampling rate is 1:1, where every packet is processed.
- The minimum supported sampling rate is 1:65,536, indicating that only one out of every 65,536 packets 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.



CHAPTER 4

Determine the Monitoring Protocol for Your Network

This section will help you understand the key distinctions between NetFlow and sFlow, equipping you with the necessary insights and knowledge to make an informed decision when it comes to selecting the most suitable monitoring protocol for your network.

Key Differences between sFlow and Netflow

This section helps you understand the difference between Netflow and sFlow based on the following factors:

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.

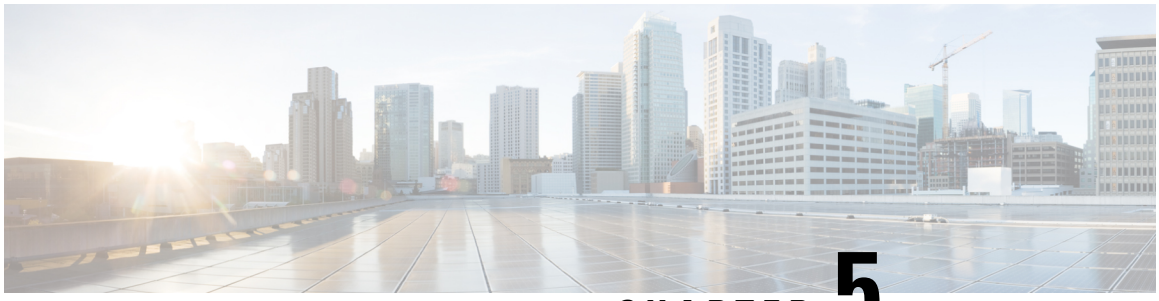
Factor	Netflow	sFlow
Mechanism	Packet aggregation into flows. Netflow is stateful as it stores the Flow record until the active timer expires.	Packets are sampled randomly, while counters are sampled based on time intervals. sFlow is not stateful since the data is not stored; it's directly forwarded 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.

Summary

Before you identify the best protocol to monitor traffic, you must consider the type of devices that make up your network and the factors that influence the network performance such as latency, scalability and so on.

- Opt for NetFlow when you want to gather extensive data and enhance visibility, though it might introduce a slightly higher latency compared to sFlow. NetFlow offers comprehensive insights into traffic flows. With this information, you can effectively address network issues, identify security risks, strategize network upgrades, and optimize bandwidth utilization.
- Opt for sFlow if you're aiming to oversee traffic in networks with limited bandwidth. sFlow imposes lesser load on network and computing resources compared to NetFlow. Consequently, for small and medium-sized businesses (SMBs) and smaller networks that use less powerful devices, sFlow is more suitable. It can effectively mitigate any performance concerns. sFlow operates with lower resource demands and because of it's sampling approach to collect data from a subset of packets. By utilizing this sampled data, you can analyze traffic patterns, enabling the identification of irregularities and the optimization of network performance.

If your network setup accommodates both NetFlow and sFlow, you have the option to leverage both technologies and take advantage of their respective strengths. By identifying specific use cases, you can make a well-informed choice regarding the protocol that meets your network's traffic monitoring needs.



CHAPTER 5

NetFlow Configuration for Traffic Monitoring and Analysis

This page will help you understand the fundamental principles, variations, benefits, and limitations associated with NetFlow. Additionally, it offers guidance on configuring NetFlow.

NetFlow Essential Concepts and Terms

- **Data source:** Specific locations within the router, such as physical interfaces and VLANs, where traffic measurements can be taken.
- **Flow**: Indicates 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**: Is 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**: Positioned within the router that has NetFlow enabled, an Exporter monitors incoming packets, and generates Flows from them. The Exporter transmits information derived from these Flows, encapsulates 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 storing it onto the hard disk.

- **NetFlow Cache**: The Cache is 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**: Is an external device or an application responsible for collecting and scrutinizing flow records to furnish valuable insights.
- **Collector address:** This comprises the IP address and a UDP port number. By default, the designated destination port number is 2055.

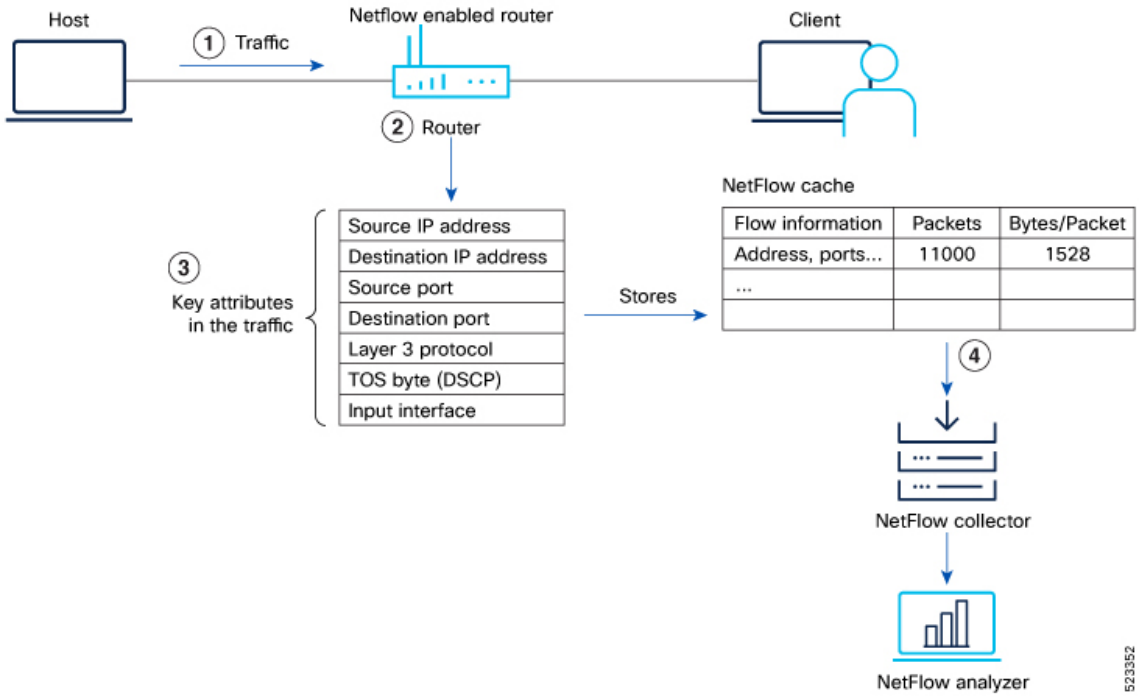
How NetFlow Works

NetFlow serves as a network monitoring protocol that facilitates the logging of metadata for each flow that traverses the router, both entering or leaving it. This protocol provides comprehensive insights into network flows, including details such as source and destination IP addresses, ports, and packet counts. It's commonly applied for traffic analysis, capacity planning, and network troubleshooting.

Recording of Packet Flows in NetFlow

The packet in NetFlow is recorded as follows:

Figure 1: Packet Flows in NetFlow



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.

Flow monitoring on Egress Interface

Egress Interface Flow Monitoring enhances network visibility and control by prioritizing outbound traffic. This capability offers advanced monitoring and management of data exiting the network, providing a more comprehensive understanding of network dynamics. The key focus of this feature is to monitor packets that are either encapsulated or decapsulated through egress sFlow.

Encapsulated and decapsulated data monitoring in sFlow serves a crucial role in safeguarding sensitive information transmitted across the network. The process involves encapsulating data with an additional layer of information, enabling verification of its authenticity and integrity. This added layer makes it challenging for attackers to intercept or modify data during transmission. Conversely, decapsulation entails removing the encapsulated data layer, empowering network devices to analyze the information and take appropriate actions

in real-time. This proactive approach aids in identifying and preventing attacks or anomalies, enhancing the overall security of the network.

Collect Additional BGP Information Elements for MPLS IPv4 and IPV6 Using IPFIX

You can now monitor and optimize your network more effectively with IPFIX, which enhances the collection of BGP Information Elements (IEs) in IPFIX records. Specifically designed to improve congestion mitigation in core-edge link scenarios, this update introduces support for gathering eight additional BGP fields in IPFIX MPLS IPv4/IPv6 records.

Additionally, two new Information Elements, namely Minimum Time-to-Live (TTL) and Maximum TTL, are recorded. These elements provide information about the minimum Time to Live for a flow and the maximum Time to Live for a flow.

Table 3: 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

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 IP Flow Information Export (IPFIX) Entities.

Configuration

The following example shows how to 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

Verify the flow monitor stats statistics using the **show flow monitor cache location command**.

```
Router#show flow monitor mpls-1 cache summary location 0/0/CPU0========== Record number: 1
 ==========
========== 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
```

Monitor Traffic Within Your Network

NetFlow extends its support to IPv4, IPv6, MPLS, BGP, SRv6, and GTP-U flow types, providing the capacity to monitor a diverse range of packet information.

Monitor IP Traffic

NetFlow can be used to collect traffic data for both IPv4 and IPv6 networks. The data collected includes information such as source and destination IP addresses, protocol types, port numbers, and bandwidth usage. This data can be used to identify network trends, detect security threats, and optimize network performance.

Key IP traffic attributes monitored:

- Source and Destination IP Addresses
- Source and Destination MAC Addresses
- Source and Destination Ports for TCP/User Datagram Protocol (UDP) ports
- 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, among others
- Counts for Packets and Bytes
- Full Spectrum of IPv6 Header fields, encompassing Flow Label and Option Header, among others
- Flow timestamps

Monitor MPLS Traffic

NetFlow can be used to collect traffic data for MPLS traffic in a networks. NetFlow provides detailed visibility into MPLS traffic, allowing you to identify anomalies, detect cyber threats, and respond quickly to potential security incidents

Key MPLS traffic attribute monitored:

- MPLS Labels

Monitor BGP Traffic

NetFlow can be used to monitor BGP traffic in your network. NetFlow can capture BGP packets and provide information on their frequency, direction, and content. This information can be used to identify potential security threats and monitor BGP router behavior. BGP is used by network routers to exchange routing information and establish optimal paths for data to travel through a network. By monitoring BGP communication with NetFlow, you can gain valuable insights into their network's performance and ensure that their routing strategies are effective.

Key BGP traffic attributes monitored:

- 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

Monitor SRv6 Traffic

Table 4: Feature History Table

Feature Name	Release Information	Description
IPFIX Enablement for SRv6 and Services over SRv6 Core	Release 7.10.1	During the transition from conventional IP/MPLS networks to SRv6-based networks, the necessity for monitoring SRv6 traffic flow becomes crucial. This feature enables IPFIX to effectively monitor SRv6 IP traffic flow from network devices in the core. The feature introduces these changes: **CLI:** • The **srv6** keyword is introduced in the **record ipv6** command.

During the transition from conventional IP/MPLS networks to SRv6-based networks, the requirement for information elements specific to SRv6 traffic flow in the core arises. To address this requirement, we have introduced support for SRv6 monitoring using Netflow.

Starting from IOS-XR software release 7.10.1, you can monitor the performance of SRv6-based core networks using IPFIX.

Restriction and Limitation

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

Interface Types Supported with NetFlow

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

NetFlow Guidelines and Limitations

- 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.

Comparative Overview of NetFlow Version 9 and Version 10 (IPFIX)

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

NetFlow Version 9 is a template-based approach that provides flexibility in the record format. It enables enhancements to NetFlow services without concurrently altering the basic flow-record format.

NetFlow Options Template

The NetFlow Options Template serves as a distinctive template record designed to communicate the format of data associated with the NetFlow operation. Instead of sharing details about IP flows, these options serve the purpose of providing metadata pertaining to the NetFlow process itself. There are distinct options templates: the sampler options template and the interface options template. The NetFlow process exports these two tables. Furthermore, the NetFlow process also exports the VRF (Virtual Routing and Forwarding) table.

Sampler Table

The Sampler Table and Interface Option Templates play a significant role in organizing information.

The Sampler Options Template consists of a sampler table, while the Interface Option Templates consists of an interface table. Enabling these options for the sampler and interface tables simplifies the process for the collector to determine data flow information.

The sampler table offers insights into active samplers. Its primary purpose is to aid the collector in estimating the sampling rate for individual data flows. 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 Table

The interface table, contains data about interfaces that are monitored for data flow. With this data, the collector derives the interface names linked to the 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 Table

The VRF table consists mapping of VRF IDs to the 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 that can be configured manually. The default value is 1800 seconds.

The VRF table consists of the following information:

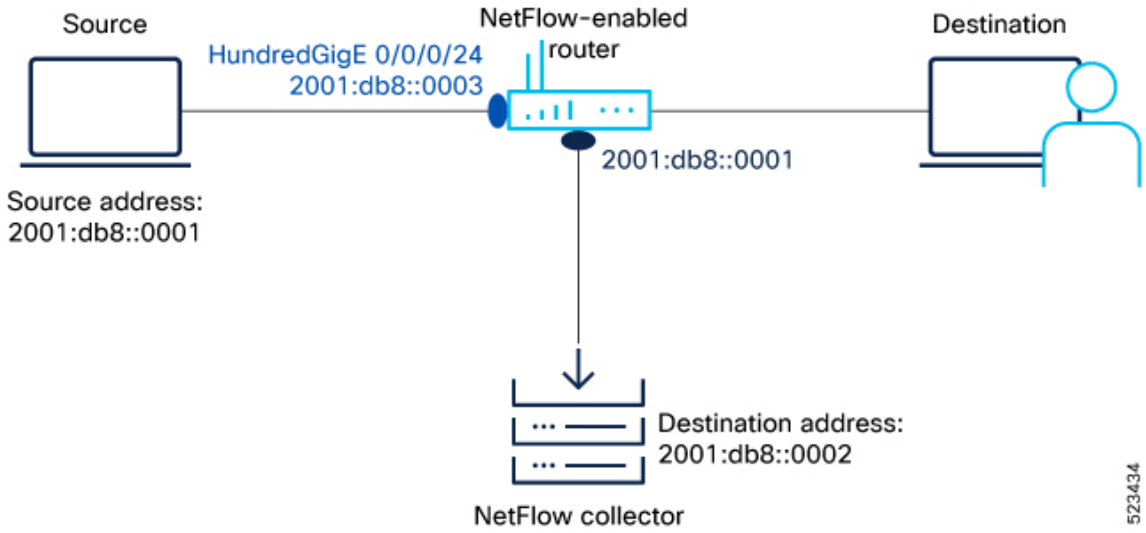
Field Name	Value
ingressVRFID	The identifier of the VRF with the name in the VRF-Name field.
VRF-Name	The VRF name has the VRFID value ingressVRFID. The value "default" indicates that the interface is not assigned explicitly to a VRF.

The data records contain ingressVRFID as an extra field in each record. The values of these fields are used to lookup 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

Let's consider the following topology to configure NetFlow.

Figure 2: NetFlow Version 9 Configuration



To monitor traffic, you must configure one or more Flow Exporter, on page 6 and associate it to a Flow Monitor Flow Monitor, on page 6 and enable NetFlow on the interface either in egress or ingress direction. Optionally, you can configure a Flow Sampler, on page 7 to set the sampling rate for flow samples.

Before you begin

First, let's gather the required details to enable NetFlow on a router:

- The IP address of the source is : `2001:db8::0003`
- The IP address of the NetFlow Collector (Destination address): `2001:db8::0002`
- Interface of the router where you want to enable Netflow: `HundredGigE 0/0/0/24`
- The NetFlow version used to transport the data to the collector: `version 9`

Step 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
```

Step 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.

Example:

Here are the examples to record 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
outer(config-fmm)# commit
Router(config-fmm)# root
Router(config)#exit
```

- **MPLS Packet Monitoring:** In this example, you create a flow monitor map to record the 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)#cache permanent
Router(config-fmm)#exit
```

- **BGP Packet Monitoring:** In this example, you create a flow monitor map to record the BGP packets with the 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
```

- **SRv6 Packet Monitoring:** Starting from Cisco IOS-XR release 7.10.1, you can create a flow monitor map to record the 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)# !
Router(config-fmm)# sampler-map SAMP
Router(config-fmm)# random 1 out-of 1000
Router(config-fmm)# !
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 ingres
```

Step 3 Configure a Flow Sampler using the sampler-map command to define the rate at which the packet sampling should be performed at the interface where NetFlow is enabled. 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 262144
Router(config)# exit
Router(config)#commit
Router(config)#exit
Router#
```

Step 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 choose to enable IPv4, IPv6, MPLS-aware NetFlow on the interface. Enable NetFlow in the ingress direction to monitor the 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 the sub-interfaces and physical interfaces under a port.

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
```

Step 5 View the running configuration to verify the configuration that you have configured.

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
```

Step 6 You can verify the the above configurations using the following 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#
```

What to do next

You can now analyse the exported data using a NetFlowAnalyser.

Verify NetFlow Version 9

Verify the flows captured using the show flow monitor name cache command.

- **Cache Summary**

  In the following example, you can verify the amount of flows added and exported.

```
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 Record for SRv6 L2 services**

  This example shows the complete recorded data for SRv6 L2 services

```
Router#show flow monitor fmm-ipv6 cache record location 0/0/CPU0
========== Record number: 1 ==========
IPv6SrcAddr : 2::2
IPv6DstAddr : bbbb:bc00:88:e000::
BGPDstOrigAS : 0
```

```
BGPSrcOrigAS : 0
BGPNextHopV6 : fe80::232:17ff:fe7e:1ce1
IPv6TC : 0
IPv6FlowLabel : 50686
IPv6OptHdrs : 0x0
IPV6Prot : 143
L4SrcPort : 0
L4DestPort : 0
L4TCPFlags : 0
IPV6DstPrfxLen : 48
IPV6SrcPrfxLen : 128
InputInterface : Hu0/0/0/10
OutputInterface : BE111.1
ForwardStatus : Fwd
FirstSwitched : 01 18:51:25:797
LastSwitched : 01 18:51:25:797
ByteCount : 61004304
PacketCount : 113814
Dir : Ing
SamplerID : 1
InputVRFID : default
OutputVRFID : default
InnerIPV4SrcAddr : 0.0.0.0
InnerIPV4DstAddr : 0.0.0.0
InnerIPv6SrcAddr : ::
InnerIPv6DstAddr : ::
InnerL4SrcPort : 0
InnerL4DestPort : 0
SrcMacAddr : 00:0c:29:0e:d8:32
DstMacAddr : 00:0c:29:0e:d8:3c
EthType : 2048
Dot1qPriority : 0
Dot1qVlanId : 2001
RecordType : SRv6 L2 Service Record
SRHFlags : 0x0
SRHTags : 0x0
SRHSegmentsLeft : 0
SRHNumSegments : 0
```

What to do next

You can now analyse the exported data using a NetFlowAnalyser.

Modify NetFlow Configuration

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 5: Flow Entities and Flow Attributes that can be altered

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

IPFIX (NetFlow Version 10)

Internet Protocol Flow Information Export (IPFIX) has been standardized by the Internet Engineering Task Force (IETF) as 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. Utilizing UDP as the transport protocol, IPFIX facilitates the seamless transfer of NetFlow information from exporters to collectors. With native support for IPv6 flow records, the inclusion of optional data fields, and the ability to send data to multiple collectors, IPFIX proves to be a versatile and powerful solution for network administrators, enabling comprehensive traffic analysis, monitoring, and enhanced visibility into network behavior.

IPFIX 315

The Internet Engineering Task Force (IETF) has standardized Internet Protocol Flow Information Export (IPFIX) as an export protocol for sending IP flow information. Router supports 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, and more. The process involves sampling and exporting the traffic flow information. Also, 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, traffic management, and more.

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.

Sampling and Exporting Information

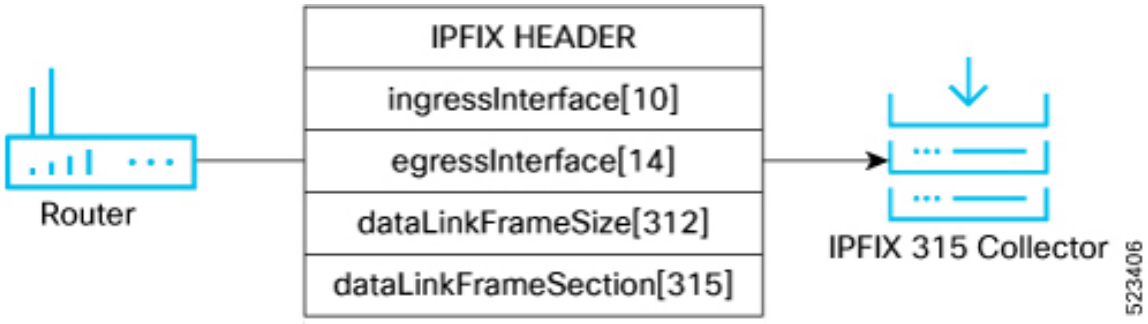
To sample the traffic flow information, configure a sampler-map that specifies the rate at which packets (one out of every 'n' packets) are sampled. Not all packets flowing through a device are exported; only the packets selected based on the sampling rate are exported.

The size of the exported packet depends on the sampled packet size and the location of the L4 header. The exported packet size is determined as follows:

- If the sampled packet size is more than 160 bytes and the L4 header is not obtained within the first 160 bytes, the exported packet size is 160 bytes.
- If the L4 header is within the first 160 bytes, the exported packet size is equal to the length of the sampled packet until the L4 header.
- If the packet size is less than 160 bytes and the L4 header isn’t within the first 160 bytes, the exported packet size is equal to the length of the packet.

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

Figure 3: IPFIX 315 Export Packet Format



Monitoring Post-QoS Data in NetFlow and IPFIX

You can now monitor information on QoS policies through the post-QoS information element field export for NetFlow and IPFIX. 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. This enhancement 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. Moreover, this feature brings added visibility to both pre- and post-QoS

Type of Service (TOS)/DSCP values within the show command, providing you with the data to assess network performance comprehensively.

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

- ipClassOfService - For IPv4, it represents the value of the Type of Service (TOS) field in the IPv4 packet header. For IPv6 packets, it signifies the value of the Traffic Class field in the IPv6 packet header. This information element is available prior to IOS-XR software release 24.1.1.
- postIpClassOfService - The definition of this information element is identical to the above mentioned definition of 'ipClassOfService,' except that it reports a potentially modified value caused by a router function after the packet has passed the observation point. This information element is introduced in IOS-XR software release 24.1.1.

NetFlow v9 and NetFlow v10 (IPFIX)

This section helps you understand the NetFlow v9 and NetFlow v10 (IPFIX) based on the following factors:

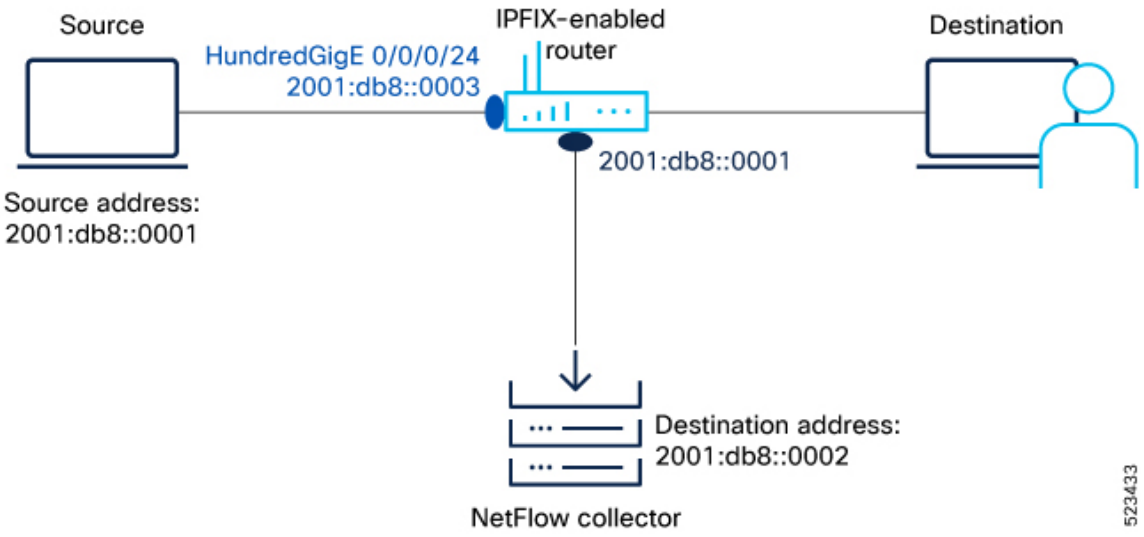
Table 6: NetFlow v9 and NetFlow v10 (IPFIX)

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

Let's consider the following topology to configure IPFIX:

Figure 4: NetFlow IPFIX Configuration



To monitor traffic, you must configure one or more Flow Exporter, on page 6 and associate it to a Flow Monitor, on page 6 and enable IPFIX on the interface either in egress or ingress direction. Optionally, you can configure a Flow Sampler, on page 7 to set the sampling rate for flow samples.

Step 1 First, let's gather the required details to enable IPFIX on a router:

- The IP address of the source : `2001:db8::0001`
- The IP address of the IPFIX Collector (Destination address): `2001:db8::0002`
- Interface of the router where we will enable IPFIX: `HundredGigE 0/0/0/24`
- NetFlow version used to transport the data to the collector: `IPFIX`

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

```
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)# exit
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.

```
Router#show 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
```

Step 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.

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.

```
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
```

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

```
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
```

Step 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.

```
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.

```
Router#show sampler-map fsm1

Sampler Map : fsm1
```

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

Step 5 View the running configuration to verify the configuration that you have configured.

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

```
Router(config)#
Router(config-if)#flow ipv4 monitor fmm1 sampler fsm1 ingress
Router(config-if)#exit
```

Configure IPFIX 315

This section provides you instructions to enable IPFIX 315 on Cisco IOS XR Software.

Step 1 Enable IPFIX 315 for flow monitoring.

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

Step 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.

```
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
```

Step 3 Create a flow monitor using the flow monitor-mapcommand 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.

```
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
```

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

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

Step 5 Apply a monitor map and a Sampler Map to a physical interface using the flowcommand to enable IPFIX on the router.

```
Router(config)#
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
```

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

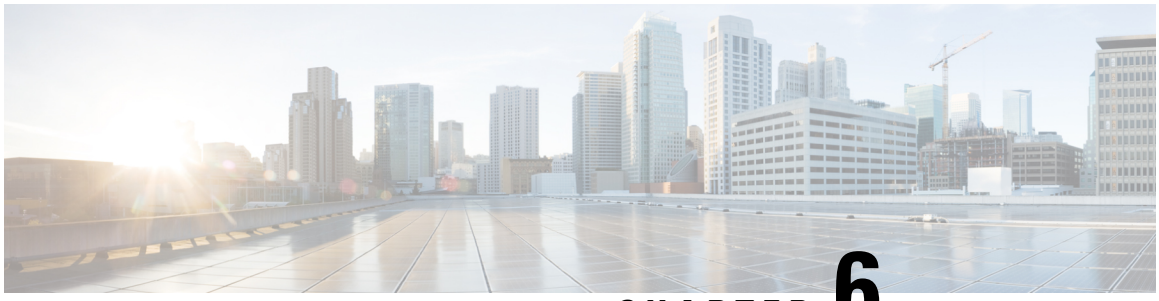
In this show output, you can see that the system has actively received and monitored a total of 630,478 IPFIX 315 packets.

```
Router#
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
```

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

This example shows that there were 50399 flows added to the cache and exported.

```
Router#
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
```



CHAPTER 6

sFlow Configuration for Traffic Monitoring and Analysis

This page helps you with information about the key concepts, advantages and limitations of sFlow, and steps to configure sFlow on your router.

sFlow Essential Concepts and Terms

This section helps you get familiar with the sFlow key terms and concepts:

- **Data source:** Location within a network device that can make traffic measurements. Examples are physical interfaces, VLANs.
- **Flow**: A Flow is defined as a set of IP packets passing a network device in the network 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 Flow record is 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:** Entity inside the network device responsible for maintaining sFlow configuration, gathering the 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:** Application that receives the sFlow datagrams from one or more agents to perform further analysis and generate reports. The collector is external to the router.
- **Sampling rate:** 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:** Period at which counters will be polled for populating the counter sample in the sFlow datagram.

- **sFlow datagram:** User Datagram Protocol (UDP) datagram exported from sFlow agent to collector. The datagram contains information about the data source, one or more flow samples, and one or more counter samples.
- **Collector address:** IP and UDP port number. The default destination port number is 6343.

Flow monitoring on Egress Interface

Egress Interface Flow Monitoring enhances network visibility and control by prioritizing outbound traffic. This capability offers advanced monitoring and management of data exiting the network, providing a more comprehensive understanding of network dynamics. The key focus of this feature is to monitor packets that are either encapsulated or decapsulated through egress sFlow.

Encapsulated and decapsulated data monitoring in sFlow serves a crucial role in safeguarding sensitive information transmitted across the network. The process involves encapsulating data with an additional layer of information, enabling verification of its authenticity and integrity. This added layer makes it challenging for attackers to intercept or modify data during transmission. Conversely, decapsulation entails removing the encapsulated data layer, empowering network devices to analyze the information and take appropriate actions in real-time. This proactive approach aids in identifying and preventing attacks or anomalies, enhancing the overall security of the network.

How sFlow Works

sFlow, a monitoring technology, operates by sampling data network traffic in real-time. However, it's important to note that sFlow doesn't encompass all network traffic, unlike Netflow.

In the context of traffic monitoring, sFlow functions by disaggregating the flow pipeline. Devices within the network stream packet headers and metadata, which are subsequently transmitted as UDP datagrams to an external collector. This collector deciphers the packets and creates flow records. A notable feature of sFlow is its capability to export this data promptly, facilitating the creation of a near real-time representation of network traffic by the collector.

The advantage of this real-time traffic analysis is its ability monitor patterns and trends within the network, facilitate the automation of traffic engineering, and aid in making well-informed decisions when planning network capacity.

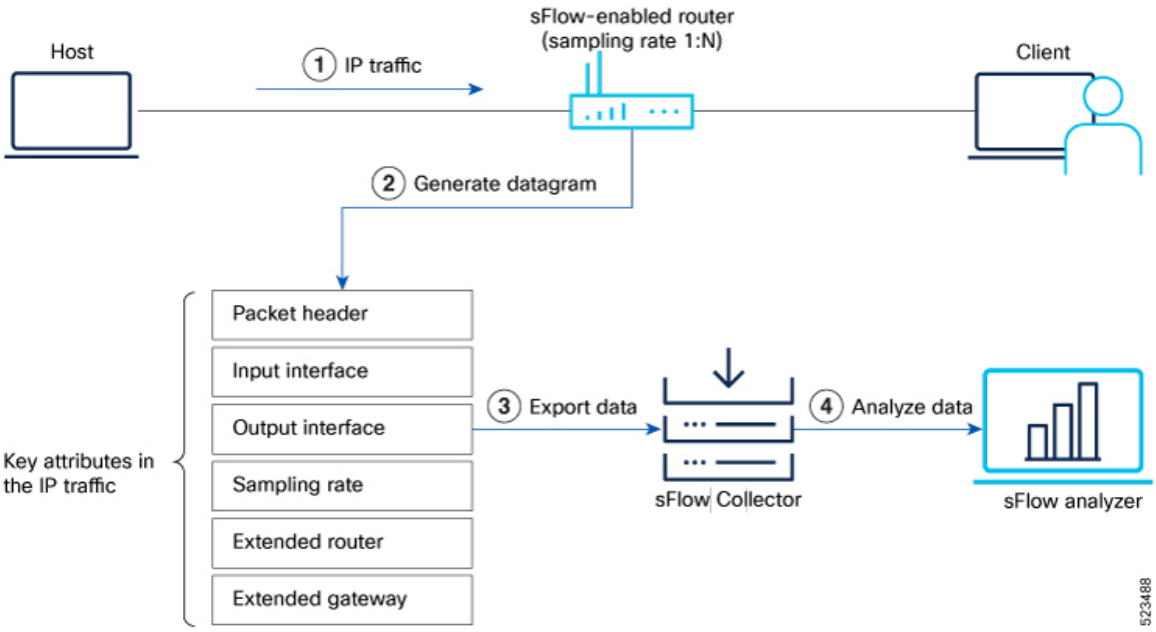
Table 7: Feature History Table

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.

Recording of Packet Flows in sFlow

The packet in sFlow is recorded as follows:

Figure 5: Packet Flows in sFlow



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

Table 8: 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.

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

Table 9: 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**

sFlow Sampling

The following methods are used in sFlow for capturing and analyzing network traffic:

- **Counter Sampling:** In the counter sampling method, only specific counters or statistics are sampled and collected for analysis. Instead of capturing and analyzing packets or flows, counter sampling focuses on monitoring and collecting information about specific network metrics or performance indicators. These metrics can include interface utilization, packet drops, CPU usage, memory usage, and other relevant statistics. Counter sampling provides a high-level view of network health and performance without the need to capture and analyze every single packet.
- **Flow Sampling:** Flow sampling, on the other hand, involves capturing and analyzing sampled network flows. A flow can be defined as a sequence of packets that share common attributes, such as source and destination IP addresses, port numbers, and protocol information. Flow sampling selects a subset of these flows for analysis. By capturing and analyzing flows, you can gain insights into traffic patterns, detect

anomalies, and monitor performance. Flow sampling allows for more granular analysis of network traffic compared to counter sampling.

You can choose the method depending on the specific monitoring needs and objectives of the network.

Configure sFlow

This page explains how to configure sFlow for monitor network traffic using sampled data.

sFlow Guidelines and Limitations

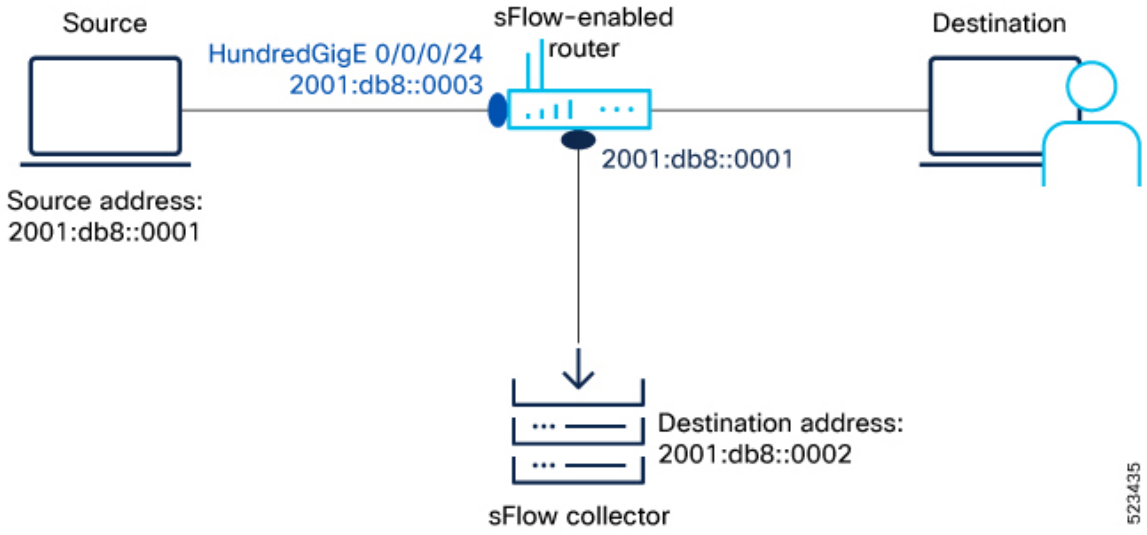
The guidelines and limitations of using flow samples and interface counters using sFlow monitoring is as follows:

- When you enable egress sFlow, the L2 information of ingress packets is captured instead of egress interface
- 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 possess distinct sub-agent ids and sequence numbers
- sFlow samples are combined into UDP packets and forwarded to sFlow collectors for analysis. It's important to note that UDP, being a connectionless protocol, doesn't ensure the delivery of data. Consequently, utilizing sFlow as a flow source could potentially lead to inaccurate representations of traffic volumes, bidirectional flows, and a reduction in alerting capabilities
- Ingress sFlow is supported on Cisco 8200 and Cisco 8800 Series Routers
- Egress sFlow is supported only on Cisco 8200 Series Routers
- A maximum of 8 export destinations per monitor map for both IPv4 and IPv6 are allowed
- Only one sampler configuration per router is allowed
- A sampling interval of 1 out of 262,144 packets as the maximum is supported
- L3 interfaces, L3 bundle interfaces, L3 sub-interfaces, L3 bundle sub-interfaces, and L3 BVI interfaces are supported
- 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

Configuring sFlow

To enable this monitoring, it is necessary to configure the sFlow agent to use a sampling mechanism, forwarding traffic data from both ingress and egress ports to a centralized data collector, also referred to as the sFlow analyzer. The sampled data is forwarded using the version 5 export format. You'll find instructions for configuring sFlow on the router in the following section. Let's use the following topology as a reference for configuring sFlow.

Figure 6: sFlow Configuration



Step 1 First, let's gather the required details to enable sFlow on a router:

- The IP address of the source : `2001:db8::0001`
- The IP address of the sFlow Collector (Destination address): `2001:db8::0002`
- Interface of the router where we will enable sFlow: `HundredGigE 0/0/0/24`
- sFlow version used to transport the data to the collector: `version 5`

Step 2 Configure the Flow Exporter using the flow flow exporter-map command to specify where and how the packets should be exported.

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.

In this example, you create an exporter map called `EXP-MAP` to export the IPv4 packets to the destination address 192.127.10.1 using the UDP transport protocol:

```
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
```

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.

Step 3 Configure the Flow Sampler using sampler-map command to define the sampling rate for flow samples, which determines how many packets (on average) that pass through the data source will generate a flow sample.

In this example, you create a sampler map called `SAMP-MAP` to sample 1 out of every 4096 packets.

```
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)
```

In this example, the sampler map configuration is successful with a sample rate of 1 out of every 4096 packets.

Step 4 Configure the Flow Monitor using flow monitor-map command to define the type of traffic to be monitored and the polling frequency. You can include one or more exporter maps in the monitor map.

In this example, you create a monitor map called `MON-MAP` and include the exporter map `EXP-MAP` to record sFlow data at a polling interval of 120 seconds:

```
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 ouput 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 sampler-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
```

In this example, the monitor map is configured successfully with the associated exporter map to monitor the interface counters at a polling interval of 30 seconds.

Step 5 Apply sFlow on an interface using command in Global Configuration mode.

In this example, you 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 the incoming packets:

```
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
```

Step 6 Enable sFlow on the RP (0/RP0/CPU0) or on the line card using hw-module profile netflow sflow-enable command.

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

Step 7 Reload the line card using **hw-module reset auto** command.

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

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

Step 8 Verify the statistics of exported traffic flow at the producer and exporter using show flow platform producer statistics location and show flow exportercommands.

Producer:

```
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
```

Exporter:

```
Router#show flow exporter EXP-MAP location 0/0/CPU0
Wed June 21 04:21:36.263 UTC
```

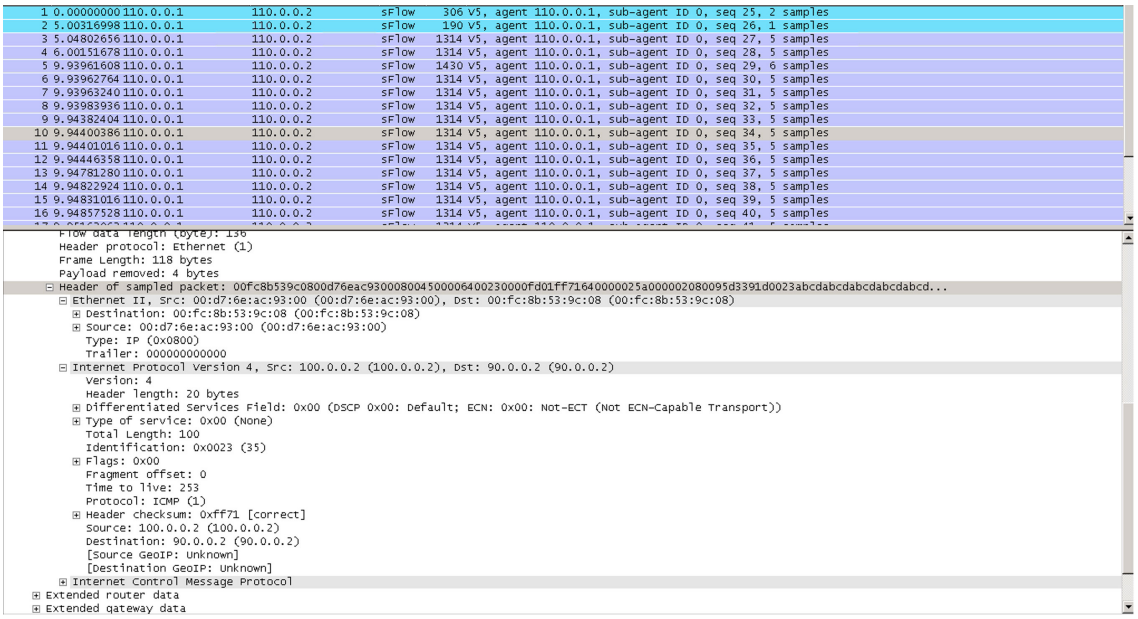
```
Flow Exporter: EXP-MAP
Export Protocol: sFlow v5
Flow Exporter memory usage: 5247776
Used by flow monitors: MON-MAP

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
```

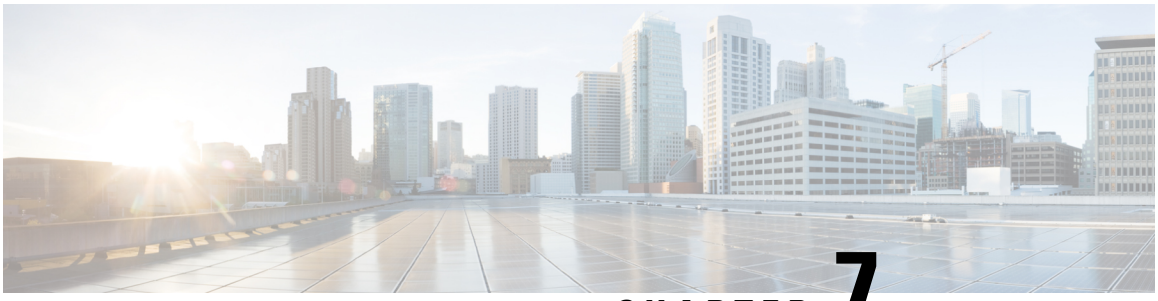
The sFlow configuration with flow and packet data is successful on the router.

What to do next

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

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



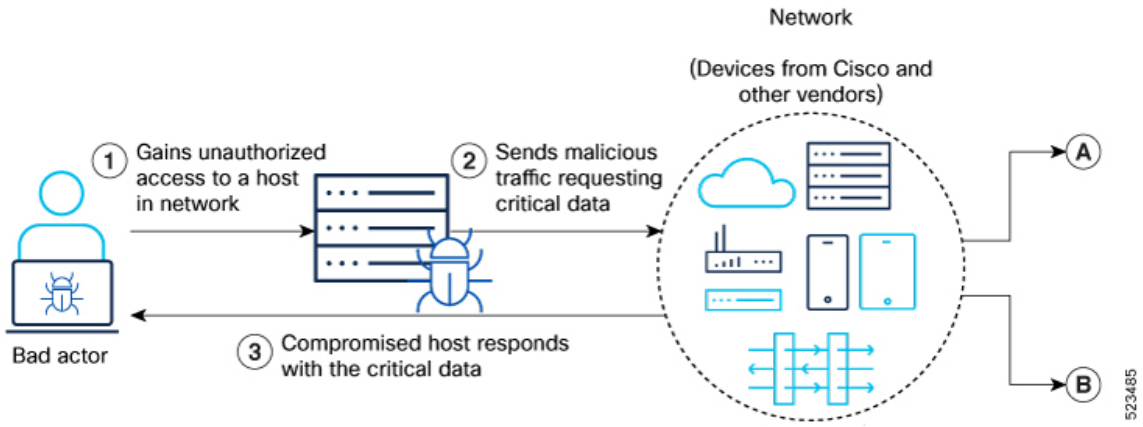


CHAPTER 7

Use Case: NetFlow and sFlow in Action

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



- 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.

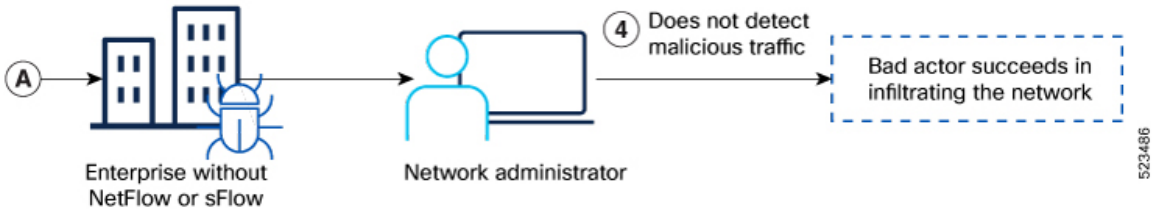
With this context, lets explore these two scenarios:

- Scenario A: Traffic Monitoring Without NetFlow and sFlow, on page 45
- Scenario B: Traffic Monitoring With NetFlow and sFlow, on page 46

Scenario A: Traffic Monitoring Without NetFlow and sFlow

In this particular situation, the enterprise had failed to implement any network traffic monitoring protocols such as NetFlow or sFlow.

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



Here is a high-level outline of the network's response to the attack:

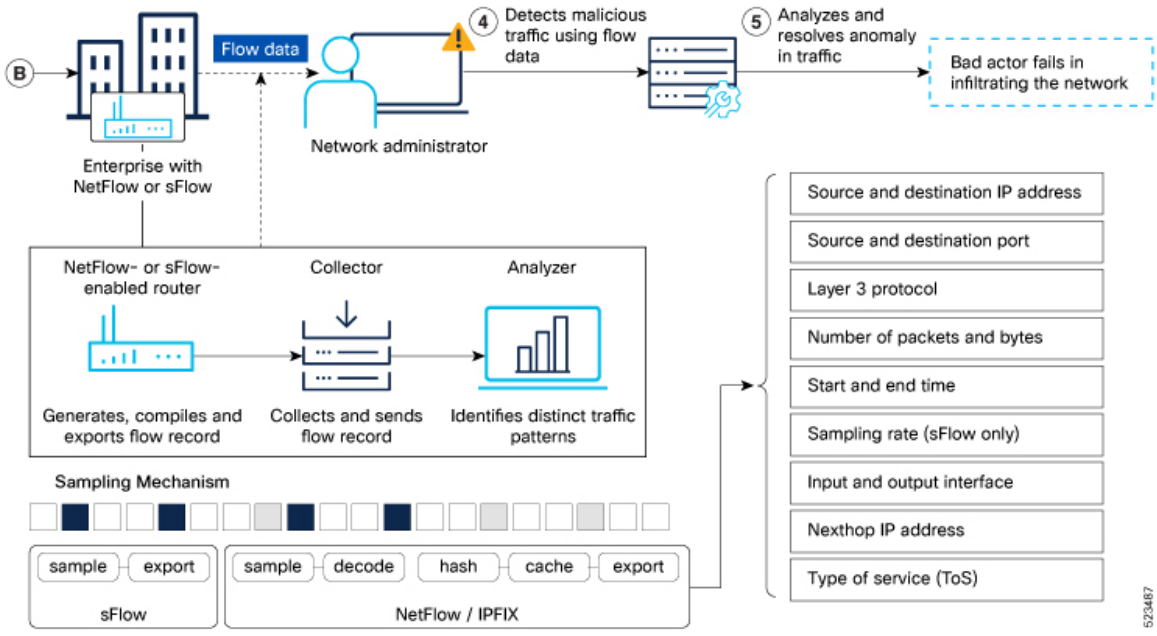
- Bypassed threat detection and response—The network administrator does not detect any unusual network patterns or intrusions immediately following the attack.
- Successful data breach—Consequently, the network is compromised through malicious traffic that gets undetected leading to loss of critical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

The overall network security posture is compromised due to lack of traffic monitoring mechanisms leading to poor visibility of the network and its functionalities.

Scenario B: Traffic Monitoring With NetFlow and sFlow

In this particular situation, the enterprise has implemented network traffic monitoring protocols such as NetFlow or sFlow.

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



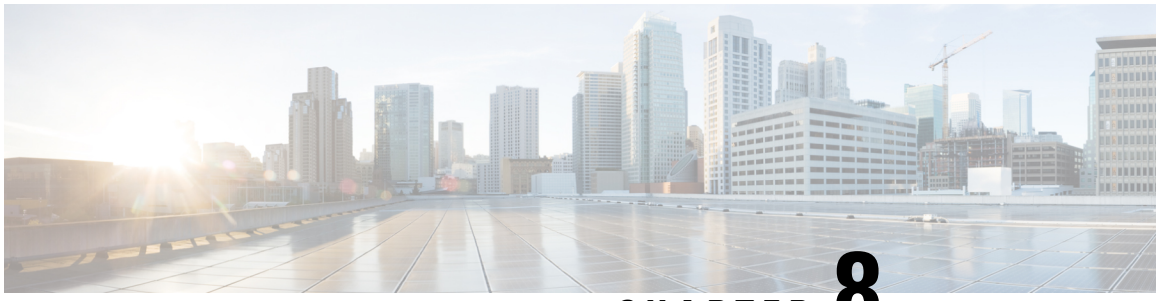
Here is a high-level outline of the network's response to the attack:

- 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, timestamps and so on. The recorded flow records are then sent to a designated collector.

- Data analysis—Utilize a NetFlow or sFlow analyzer or security monitoring tool to process and analyze the collected data. The tool can identify 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 can compare network traffic with predefined security policies. If a potential threat is detected, the analyzer generates an alert. This alert can be sent to the network administrator for further investigation.
- Prompt investigation and responsive action—Upon receiving the alert, the network administrator can investigate the identified threat. They can analyze additional logs, inspect packet captures, or perform other security measures to gather more information about the threat. Once the threat is confirmed, appropriate actions can be taken to mitigate the impact by blocking the malicious IP addresses and isolating affected hosts to prevent further harm.

By leveraging NetFlow and sFlow for threat identification, you can proactively detect and respond to security threats, enhancing the overall network security posture. It allows for early threat detection, and faster incident response, ultimately reducing the risk of a successful attack.



CHAPTER 8

YANG Data Models for NetFlow and sFlow

In this section, you'll learn to use the YANG data models to configure and retrieve the operational status of NetFlow and sFlow on Series Routers.

What You'll Find in This Section

Cisco IOS XR supports configuring NetFlow and sFlow using both traditional Command Line Interface (CLI) using commands as well as programmatically using YANG data models. In this section, you'll find references to supported YANG data models and an understanding about accessing and using these data models.

To get started with using these data models, see:

- List of YANG Data Models for NetFlow and sFlow, on page 49
- Access Data Models, on page 50
- Get Started With IOS XR YANG Data Models, on page 52

List of YANG Data Models for NetFlow and sFlow

Here is a list of YANG data models that you can use to configure and manage NetFlow and sFlow on the router:

Table 10: 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 following options:

Access Data Models

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

Access Data Models From Router

To access data models directly from the router, you can follow these steps:

Step 1 Enter the global configuration mode.

Example:

```
Router#configure
```

Step 2 Configure the NETCONF network management protocol to remotely configure and manage the router using YANG data models.

Example:

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

Step 3 Commit the configuration.

Example:

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

Step 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.

Access Data Models From Cisco Feature Navigator

To access data models from Cisco Feature Navigator, you can follow these steps:

Step 1 Go to Cisco Feature Navigator.

Step 2 If you have a Cisco.com account, click on the **Login** button and enter your credentials. If you don't have an account, you can click **Continue as Guest**.

You will be directed to the Cisco Feature Navigator main page.

Step 3 Click **YANG Data Models**.

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

The data models are listed based on type—Cisco XR native models, Unified models and OpenConfig models.

You can use the search field to search for specific data model of interest.

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

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

You can also click the **Download** icon to export the data model information in Excel format.

This visual tree form helps you get insights into the nodes that you can use to automate your network.

The data models on Cisco Feature Navigator is regularly updated based on IOS XR release. If you encounter any problem or have suggestions for improvements, share your experience using Send us your feedback link.

Access Data Models From GitHub

To access the data models from GitHub repository, you can follow these steps:

Step 1 Go to the GitHub repository for data models.

On the repository page, you will find a list of folders based on IOS XR releases.

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

Inside the folder, you will find a list of YANG files representing different data models.

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

You can also click on 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. To view a comprehensive list of the data models supported in a release, navigate to the **Available-Content.md** file in the repository. The unsupported sensor paths are documented as deviations. For example, `openconfig-acl.yang` provides details about the supported sensor paths, whereas `cisco-xr-openconfig-acl-deviations.yang` shows the unsupported sensor paths for `openconfig-acl.yang` model.

Step 4 Repeat the above steps for other versions or data models of interest.

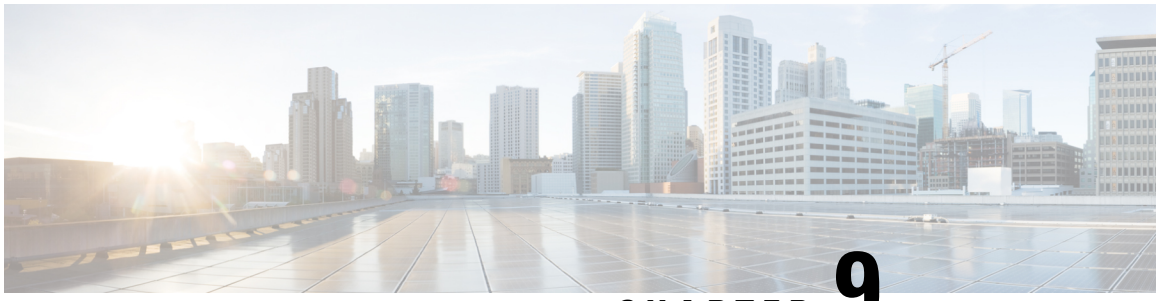
The GitHub repository for IOS XR data models is regularly updated based on release. You can also contribute to the repository by submitting pull requests, opening issues if you encounter any problems or have suggestions for improvements.

Get Started With IOS XR YANG Data Models

Here is a generic outline of the steps involved in programmatically configuring your router using YANG data models:

1. Enable network management protocol—Manage the router remotely using the protocols such as NETCONF or gRPC.
2. Install the necessary libraries and tools—Depending on the programming language you are using, you may need to install libraries or tools to programatically interact with the router. For example, if you are using Python, you might need to install the `ncclient` library.
3. Establish a session with the router—Use the programming language of your choice to establish a connection to the router using NETCONF or gRPC protocols. This involves providing connection parameters such as device IP address, username, password, and port number.
4. Retrieve the router capabilities—View the supported features and functionalities available on the router.
5. Create or modify configurations—Use YANG data models to create or modify the configuration on the router.
6. Apply the configuration—Push the updated configuration via the NETCONF or gRPC protocol to modify the router's running configuration to reflect the desired changes.
7. Validate the configuration—Verify that the changes are successfully applied. You can retrieve the running configuration or specific configuration parameters to ensure that the device is configured as intended.

For detailed instructions about using the data models, refer the *Programmability Configuration Guide for Cisco NCS 5500 Series Routers*.



CHAPTER 9

Command-line Interface (CLIs) for NetFlow and sFlow

The Cisco Command Reference Guide serves as a comprehensive resource, offering a catalog of command-line interface (CLI) commands for configuring and verifying NetFlow and sFlow implementation.

What You'll Find in This Section

Cisco IOS XR supports configuring NetFlow and sFlow using both traditional Command Line Interface (CLI) using commands as well as programmatically using YANG data models.

In this section, you'll find:

Reference to Command Reference Guide

The Cisco Command Reference Guide serves as a comprehensive resource, offering a catalog of command-line interface (CLI) commands for configuring and verifying NetFlow settings.

To view the list of supported commands, refer *Netflow Command Reference for Cisco NCS 5500 Series Routers and Cisco NCS 540and NCS 560 Series Routers*.