Network Management Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches

First Published: 2020-08-10

Last Modified: 2024-04-05

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CHAPTER 1

Configuring Embedded Packet Capture

- Embedded Packet Capture Overview, on page 1
- Configuring Embedded Packet Capture, on page 1
- Monitoring and Maintaining Captured Data, on page 2
- Feature History, on page 3

Embedded Packet Capture Overview

Embedded Packet Capture (EPC) is an onboard packet capture facility that allows network administrators to capture packets flowing to, through, and from the device and to analyze them locally or save and export them for offline analysis. The captured data is stored in .pcap file format, which can be analyzed by using a standard packet analysis tool such as Wireshark. This feature facilitates troubleshooting by gathering information about the packet format. This feature also facilitates application analysis and security.

Embedded Packet Capture (EPC) provides an embedded systems management facility that helps in tracing and troubleshooting packets. The network administrator may define the capture buffer size and the maximum number of bytes of each packet to capture. The packet capture rate can be throttled using further administrative controls. For example, options allow for filtering the packets to be captured using an Access Control List and, optionally, further defined by specifying a maximum packet capture rate or by specifying a sampling interval.

Note Packet Capture is supported only on physical interfaces with the ingress direction. ACL filter needs to be configured before configuring EPC.

Configuring Embedded Packet Capture

Follow these steps to configure Embedded Packet Capture:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enable privileged EXEC mode.

	Command or Action	Purpose
Step 2	**monitor capture** *capture-name* **access-list** *access-list-name*	Configure a monitor capture specifying an access list as the core filter for the packet capture.
Step 3	**monitor capture** *capture-name* **limit duration** *seconds*	Configure monitor capture limits.
Step 4	**monitor capture** *capture-name* **interface** *interface-name* **in**	Configure monitor capture specifying an attachment point and the packet flow direction.
Step 5	**monitor capture** *capture-name* **buffer circular size** *bytes*	Configure a buffer to capture packet data. This size can be maximum 100 MB.
Step 6	**monitor capture** *capture-name* **start**	Start the capture of packet data at a traffic trace point into a buffer.
Step 7	**monitor capture** *capture-name* **export** *file-location/file-name*	Export captured data for analysis.
Step 8	**monitor capture** *capture-name* **stop**	Stop the capture of packet data at a traffic trace point.
Step 9	**monitor capture** *capture-name* **clear**	Clear the captured buffer data.
Step 10	**end**	Exit privileged EXEC mode.

Example

Monitoring and Maintaining Captured Data

Perform this task to monitor and maintain the packet data captured. Capture buffer details and capture point details are displayed.

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enable privileged EXEC mode.
Step 2	**show monitor capture** *capture-buffer-name* **buffer dump**	(Optional) Display a hexadecimal dump of captured packet and its metadata.
Step 3	**show monitor capture** *capture-buffer-name* **parameter**	(Optional) Display a list of commands that were used to specify the capture.
Step 4	**debug epc capture-point**	(Optional) Enable packet capture point debugging.
Step 5	**debug epc provision**	(Optional) Enables packet capture provisioning debugging.
Step 6	**exit**	Exit privileged EXEC mode.

Example

Feature History

Feature Name	Release	Feature Information
Embedded Packet Capture	Cisco IOS XE 16.11.1	Initial support on Cisco Catalyst IE 3200, 3300, 3400, and Cisco Embedded Service 3300 Series Switches

CHAPTER 2

Flexible NetFlow Export of Cisco TrustSec Fields

- Cisco TrustSec Fields in Flexible NetFlow, on page 5
- Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record, on page 6
- Configuring a Flow Exporter, on page 8
- Configuring a Flow Monitor, on page 9
- Applying a Flow Monitor on an Interface, on page 10
- Verifying Flexible NetFlow Export of Cisco TrustSec Fields, on page 11
- Configuration Examples for Flexible NetFlow Export of Cisco TrustSec Fields, on page 14

Cisco TrustSec Fields in Flexible NetFlow

The Flexible NetFlow Export of Cisco TrustSec Fields feature supports the Cisco TrustSec fields in the Flexible NetFlow (FNF) flow record and helps to monitor, troubleshoot, and identify non-standard behavior for Cisco TrustSec deployments.

Note Flexible netflow records and recording of Cisco TrustSec fields in the IP packets only work on IPv4 packets. IPv6 packets do not support capture of Cisco TrustSec fields.

The Cisco TrustSec fields, source security group tag (SGT) and destination security group tag (DGT) in the Flexible NetFlow (FNF) flow records help administrators correlate the flow with identity information. It enables network engineers to gain a detailed understanding of the customer use of the network and application resources. This information can then be used to efficiently plan and allocate access and application resources and to detect and resolve potential security and policy violations.

The Cisco TrustSec fields are supported for ingress FNF and for unicast and multicast traffic.

The following table presents Netflow v9 enterprise specific field types for Cisco TrustSec that are used in the FNF templates for the Cisco TrustSec source and destination source group tags.

ID	Description
CTS_SRC_GROUP_TAG	Cisco Trusted Security Source Group Tag
CTS_DST_GROUP_TAG	Cisco Trusted Security Destination Group Tag

The Cisco TrustSec fields are configured in addition to the existing match fields under the FNF flow record. The following configurations are used to add the Cisco TrustSec flow objects to the FNF flow record as non-key fields and to configure the source and destination security group tags for the packet.

The **collect flow cts {source | destination} group-tag** command is configured under flow record to specify the Cisco TrustSec fields as non-key fields. The values in non-key fields are added to flows to provide additional information about the traffic in the flows.

The flow record is then configured under flow monitor and the flow monitor is applied to the interface. To export the FNF data, a flow exporter needs to be configured and then added under the flow monitor.

Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **match ipv4 protocol**
5. **match ipv4 source address**
6. **match ipv4 destination address**
7. **match transport source-port**
8. **match transport destination-port**
9. **collect flow cts source group-tag**
10. **collect flow cts destination group-tag**
11. **collect counter packets**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow record** *record-name* **Example:** `Device(config)# flow record cts-record-ipv4`	Creates a new Flexible NetFlow (FNF) flow record, or modifies an existing FNF flow record, and enters Flexible NetFlow flow record configuration mode.

	Command or Action	Purpose
Step 4	**match ipv4 protocol** **Example:** `Device(config-flow-record)# match ipv4 protocol`	(Optional) Configures the IPv4 protocol protocol as a key field for a flow record. **Note** For Cisco CSR100V, ISR 4400, and ASR 1000 platforms, Cisco TrustSec fields are supported only in IPv4 FNF records.
Step 5	**match ipv4 source address** **Example:** `Device(config-flow-record)# match ipv4 source address`	(Optional) Configures the IPv4 source address as a key field for a flow record. **Note** For Cisco CSR100V, ISR 4400, and ASR 1000 platforms, Cisco TrustSec fields are supported only in IPv4 FNF records.
Step 6	**match ipv4 destination address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	(Optional) Configures the IPv4 destination address as a key field for a flow record. **Note** For Cisco CSR100V, ISR 4400, and ASR 1000 platforms, Cisco TrustSec fields are supported only in IPv4 FNF records.
Step 7	**match transport source-port** **Example:** `Device(config-flow-record)# match transport source-port`	(Optional) Configures the transport source port as a key field for a flow record.
Step 8	**match transport destination-port** **Example:** `Device(config-flow-record)# match transport destination-port`	(Optional) Configures the transport destination port as a key field for a flow record.
Step 9	**collect flow cts source group-tag** **Example:** `Device(config-flow-record)# collect flow cts source group-tag`	(Optional) Configures the Cisco TrustSec source security group tag (SGT) in the FNF flow record as non-key fields.
Step 10	**collect flow cts destination group-tag** **Example:** `Device(config-flow-record)# collect flow cts destination group-tag`	(Optional) Configures the Cisco TrustSec destination security group tag (DGT) in the FNF flow record as non-key fields.
Step 11	**collect counter packets** **Example:** `Device(config-flow-record)# collect counter packets`	(Optional) Configures the number of packets seen in a flow as a non-key field and enables collecting the total number of packets from the flow.

	Command or Action	Purpose

Step 12	**end**
	Example:
	`Device(config-flow-record)# end`
		Exits Flexible NetFlow flow record configuration mode and returns to privileged EXEC mode.

Configuring a Flow Exporter

Each flow exporter supports only one destination. If you want to export the data to multiple destinations, you must configure multiple flow exporters and assign them to the flow monitor.

Before you begin

Ensure that you create a flow record. For more information see the “Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record” section and the “Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow exporter** *exporter-name*
4. **destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow exporter** *exporter-name* **Example:** `Device(config)# flow exporter EXPORTER-1`	Creates a flow exporter or modifies an existing flow exporter, and enters Flexible NetFlow flow exporter configuration mode.
Step 4	**destination** {*ip-address* | *hostname*} [**vrf** *vrf-name*] **Example:**	Specifies the IP address or hostname of the destination system for the exporter.

	Command or Action	Purpose
	`Device(config-flow-exporter)# destination 172.16.10.2`	

Step 5	**end**
	Example:
	`Device(config-flow-exporter)# end`
		Exits Flexible NetFlow flow exporter configuration mode and returns to privileged EXEC mode.

Configuring a Flow Monitor

Before you begin

To add a flow exporter to the flow monitor for data export, ensure that you create the flow exporter. For more information see the "Configuring a Flow Exporter" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **record** *record-name*
5. **exporter** *exporter-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**flow monitor** *monitor-name* **Example:** `Device(config)# flow monitor FLOW-MONITOR-1`	Creates a flow monitor or modifies an existing flow monitor, and enters Flexible NetFlow flow monitor configuration mode.
Step 4	**record** *record-name* **Example:**	Specifies the record for the flow monitor.

	Command or Action	Purpose
	`Device(config-flow-monitor)# record cts-record-ipv4`	
Step 5	**exporter** *exporter-name* **Example:** `Device(config-flow-monitor)# exporter EXPORTER-1`	Specifies the exporter for the flow monitor.

Step 6	**end**
	Example:
	`Device(config-flow-monitor)# end`
		Exits Flexible NetFlow flow monitor configuration mode and returns to privileged EXEC mode.

Applying a Flow Monitor on an Interface

To activate a flow monitor, the flow monitor must be applied to at least one interface.

Before you begin

Ensure that you create a flow monitor. For more information see the “Configuring a Flow Monitor” section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip flow monitor** *monitor-name* **input**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Gi1/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip flow monitor** *monitor-name* **input** **Example:** `Device (config-if)# ip flow monitor FLOW-MONITOR-1 input`	Activates a flow monitor that was created previously by assigning it to the interface to analyze traffic.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Verifying Flexible NetFlow Export of Cisco TrustSec Fields

SUMMARY STEPS

1. **enable**
2. **show flow record** *record-name*
3. **show flow exporter** *exporter-name*
4. **show flow monitor** *monitor-name*
5. **show flow monitor** *monitor-name* **cache**
6. **show flow interface** *type number*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show flow record** *record-name*

Displays the details of the specified Flexible NetFlow (FNF) flow record.

Example:

```
Device> show flow record cts-recordipv4

flow record cts-recordipv4:
  Description:        User defined
  No. of users:       1
  Total field space:  30 bytes
  Fields:
    match ipv4 protocol
    match ipv4 source address
    match ipv4 destination address
    match transport source-port
    match transport destination-port
    match interface output
    collect flow direction
    collect flow cts source group-tag
    collect flow cts destination group-tag
    collect counter packets
```

Step 3 **show flow exporter** *exporter-name*

Displays the current status of the specified FNF flow exporter.

Example:

```
Device> show flow exporter EXPORTER-1

Flow Exporter EXPORTER-1:
  Description:              User defined
  Export protocol:          NetFlow Version 9
  Transport Configuration:
    Destination IP address: 100.100.100.1
    Source IP address:      3.3.3.2
    Transport Protocol:     UDP
    Destination Port:       2055
    Source Port:            65252
    DSCP:                   0x0
    TTL:                    255
    Output Features:        Used
```

Step 4 **show flow monitor** *monitor-name*

Displays the status and statistics of the specified FNF flow monitor.

Example:

```
Device> show flow monitor FLOW-MONITOR-1

Flow Monitor FLOW-MONITOR-1:
  Description:       User defined
  Flow Record:       cts-recordipv4
  Flow Exporter:     EXPORTER-1
  Cache:
    Type:                 normal (Platform cache)
    Status:               allocated
    Size:                 200000 entries
    Inactive Timeout:     60 secs
    Active Timeout:       1800 secs
    Update Timeout:       1800 secs
    Synchronized Timeout: 600 secs
    Trans end aging:      off
```

Step 5 **show flow monitor** *monitor-name* **cache**

Displays the contents of the specified FNF flow monitor cache.

Example:

```
Device> show flow monitor FLOW-MONITOR-1 cache

  Cache type:                                 Normal
  Cache size:                                 4096
  Current entries:                            2
  High Watermark:                             2

  Flows added:                                6
  Flows aged:                                 4
    - Active timeout      (1800 secs)         0
    - Inactive timeout    (15 secs)           4
    - Event aged                              0
    - Watermark aged                          0
    - Emergency aged                          0

  IPV4 SOURCE ADDRESS:                        10.1.0.1
  IPV4 DESTINATION ADDRESS:                   172.16.2.0
  TRNS SOURCE PORT:                           58817
  TRNS DESTINATION PORT:                      23
  FLOW DIRECTION:                             Input
  IP PROTOCOL:                                6
  SOURCE GROUP TAG:                           100
  DESTINATION GROUP TAG:                      200
  counter packets:                            10

  IPV4 SOURCE ADDRESS:                        172.16.2.0
  IPV4 DESTINATION ADDRESS:                   10.1.0.1
  TRNS SOURCE PORT:                           23
  TRNS DESTINATION PORT:                      58817
  FLOW DIRECTION:                             Output
  IP PROTOCOL:                                6
  SOURCE GROUP TAG:                           200
  DESTINATION GROUP TAG:                      100
  counter packets:                            8
```

Step 6 **show flow interface** *type number*

Displays the details of the FNF flow monitor applied on the specified interface. If a flow monitor is not applied on the interface, then the output is empty.

Example:

```
Device>  show flow interface Gi1/1

Interface GigabitEthernet1/1
  FNF:  monitor:          FLOW-MONITOR-1
        direction:        Input
        traffic(ip):      on
```

Configuration Examples for Flexible NetFlow Export of Cisco TrustSec Fields

Example: Configuring Cisco TrustSec Fields as Non-Key Fields in the Flow Record

The following example shows how to configure the Cisco TrustSec flow objects as non-key fields in an IPv4 Flexible NetFlow flow record:

```
Device> enable
Device# configure terminal
Device(config)# flow record cts-record-ipv4
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# collect flow cts source group-tag
Device(config-flow-record)# collect flow cts destination group-tag
Device(config-flow-record)# collect counter packets
Device(config-flow-record)# end
```

Example: Configuring a Flow Exporter

```
Device> enable
Device# configure terminal
Device(config)# flow exporter EXPORTER-1
Device(config-flow-exporter)# destination 172.16.10.2
Device(config-flow-exporter)# end
```

Example: Configuring a Flow Monitor

```
Device> enable
Device# configure terminal
Device(config)# flow monitor FLOW-MONITOR-1
Device(config-flow-monitor)# record cts-record-ipv4
Device(config-flow-monitor)# exporter EXPORTER-1
Device(config-flow-monitor)# end
```

Example: Applying a Flow Monitor on an Interface

The following example shows how to activate an IPv4 flow monitor by applying it to an interface to analyze traffic. To activate an IPv6 flow monitor, replace the **ip** keyword with the **ipv6** keyword.

```
Device> enable
Device# configure terminal
Device(config)# interface Gi1/1
Device(config-if)# ip flow monitor FLOW-MONITOR-1 input
Device(config-if)# end
```

CHAPTER 3

Configuring SNMP

- Prerequisites for SNMP, on page 17
- Restrictions for SNMP, on page 19
- Information About SNMP, on page 19
- How to Configure SNMP, on page 24
- SNMP Examples, on page 32
- Monitoring SNMP Status, on page 33

Prerequisites for SNMP

Supported SNMP Versions

This software release supports the following SNMP versions:

- SNMPv1—The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- SNMPv2C replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the community-string-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic. It has these features:
  - SNMPv2—Version 2 of the Simple Network Management Protocol, a Draft Internet Standard, defined in RFCs 1902 through 1907.
  - SNMPv2C—The community-string-based Administrative Framework for SNMPv2, an Experimental Internet Protocol defined in RFC 1901.
- SNMPv3—Version 3 of the SNMP is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network and includes these security features:
  - Message integrity—Ensures that a packet was not tampered with in transit.
  - Authentication—Determines that the message is from a valid source.
  - Encryption—Mixes the contents of a package to prevent it from being read by an unauthorized source.

Note To select encryption, enter the **priv** keyword.

Both SNMPv1 and SNMPv2C use a community-based form of security. The community of managers able to access the agent's MIB is defined by an IP address access control list and password.

SNMPv2C includes a bulk retrieval function and more detailed error message reporting to management stations. The bulk retrieval function retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2C report the error type.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy set up for a user and the group within which the user resides. A security level is the permitted level of security within a security model. A combination of the security level and the security model determine which security method is used when handling an SNMP packet. Available security models are SNMPv1, SNMPv2C, and SNMPv3.

The following table identifies characteristics and compares different combinations of security models and levels:

Table 1: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	Result
SNMPv1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv2C	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv3	noAuthNoPriv	Username	No	Uses a username match for authentication.
SNMPv3	authNoPriv	Message Digest 5 (MD5) or Secure Hash Algorithm (SHA)	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.

Model	Level	Authentication	Encryption	Result
SNMPv3	authPriv	MD5 or SHA	Data Encryption Standard (DES) or Advanced Encryption Standard (AES)	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Allows specifying the User-based Security Model (USM) with these encryption algorithms: • DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard. • 3DES 168-bit encryption • AES 128-bit, 192-bit, or 256-bit encryption

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1, SNMPv2C, or SNMPv3.

Restrictions for SNMP

Version Restrictions

- SNMPv1 does not support informs.

Information About SNMP

The following sections provide information about SNMP.

SNMP Overview

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a management information base (MIB). The SNMP manager can be part of a network management system (NMS) such as Cisco Prime Infrastructure. The agent and MIB reside on the device. To configure SNMP on the device, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a TCP connection, loss of connection to a neighbor, or other significant events.

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in the following table:

Table 2: SNMP Operations

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table.[1]
get-bulk-request[2]	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

[1] With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.

[2] The get-bulk command only works with SNMPv2 or later.

Note We recommend that the SNMP Manager exclude the **ciscoFlashFileDate** MIB object from its query, to avoid performance related issues. This is because, though the **ciscoFlashFileDate** object is published in the MIB, it is not supported on the product.

SNMP Agent Functions

The SNMP agent can receive requests from one or more SNMP managers. Every request carries the NMS IP address, the number of times an NMS polls the agent, and a timestamp of polling. This information can be tracked for both IPv4 and IPv6 servers.

The SNMP agent responds to SNMP manager requests as follows:

- Get a MIB variable—The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- Set a MIB variable—The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

Use the **show snmp stats hosts** command to display the list of the SNMP managers requests in the queue, and use the **clear snmp stats hosts** command to clear the queue.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the device, the community string definitions on the NMS must match at least one of the three community string definitions on the device.

A community string can have one of the following attributes:

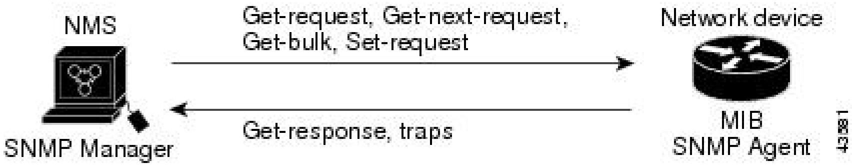
- Read-only (RO)—Gives all objects in the MIB except the community strings read access to authorized management stations, but does not allow write access.
- Read-write (RW)—Gives all objects in the MIB read and write access to authorized management stations, but does not allow access to the community strings.
- When a cluster is created, the command device manages the exchange of messages among member devices and the SNMP application. The Network Assistant software appends the member device number (@esN, where N is the device number) to the first configured RW and RO community strings on the command device and propagates them to the member devices.

SNMP MIB Variables Access

An example of an NMS is the Cisco Prime Infrastructure network management software. Cisco Prime Infrastructure software uses the device MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed as a graph and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in the figure, the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in *get-request*, *get-next-request*, and *set-request* format.

Figure 1: SNMP Network



SNMP Notifications

SNMP allows the device to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword traps refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Note SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be resent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the device and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be resent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the device is a concern and notification is not required, use traps.

SNMP ifIndex MIB Object Values

The SNMP agent's IF-MIB module comes up shortly after reboot. As various physical interface drivers are initialized they register with the IF-MIB module, essentially saying "Give me an ifIndex number". The IF-MIB module assigns the next available ifIndex number on a first-come-first-served basis. That is, minor differences in driver initialization order from one reboot to another can result in the same physical interface getting a different ifIndex number than it had before the reboot (unless ifIndex persistency is enabled of course).

SNMP and Syslog Over IPv6

To support both IPv4 and IPv6, IPv6 network management requires both IPv6 and IPv4 transports. Syslog over IPv6 supports address data types for these transports.

Simple Network Management Protocol (SNMP) and syslog over IPv6 provide these features:

- Support for both IPv4 and IPv6
- IPv6 transport for SNMP and to modify the SNMP agent to support traps for an IPv6 host

- SNMP- and syslog-related MIBs to support IPv6 addressing
- Configuration of IPv6 hosts as trap receivers

For support over IPv6, SNMP modifies the existing IP transport mapping to simultaneously support IPv4 and IPv6. These SNMP actions support IPv6 transport management:

- Opens User Datagram Protocol (UDP) SNMP socket with default settings
- Provides a new transport mechanism called *SR_IPV6_TRANSPORT*
- Sends SNMP notifications over IPv6 transport
- Supports SNMP-named access lists for IPv6 transport
- Supports SNMP proxy forwarding using IPv6 transport
- Verifies SNMP Manager feature works with IPv6 transport

For information on SNMP over IPv6, including configuration procedures, see the “Managing Cisco IOS Applications over IPv6” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

For information about syslog over IPv6, including configuration procedures, see the “Implementing IPv6 Addressing and Basic Connectivity” chapter in the *Cisco IOS IPv6 Configuration Library* on Cisco.com.

Default SNMP Configuration

Feature	Default Setting
SNMP agent	Disabled[3].
SNMP trap receiver	None configured.
SNMP traps	None enabled except the trap for TCP connections (tty).
SNMP version	If no version keyword is present, the default is Version 1.
SNMPv3 authentication	If no keyword is entered, the default is the **noauth** (noAuthNoPriv) security level.
SNMP notification type	If no type is specified, all notifications are sent.

[3] This is the default when the device starts and the startup configuration does not have any **snmp-server** global configuration commands.

SNMP Configuration Guidelines

The device requires one of the following global configuration commands configured in order to open SNMP UDP ports 161 and 162 and enable the SNMP agent: **snmp-server host**, or **snmp-server user**, or **snmp-server community**, or **snmp-server manager**.

An SNMP *group* is a table that maps SNMP users to SNMP views. An SNMP *user* is a member of an SNMP group. An SNMP *host* is the recipient of an SNMP trap operation. An SNMP *engine ID* is a name for the local or remote SNMP engine.

When configuring SNMP, follow these guidelines:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** global configuration command auto-generates a notify view for the user and then adds it to the group associated with that user. Modifying the group's notify view affects all users associated with that group.
- To configure a remote user, specify the IP address or port number for the remote SNMP agent of the device where the user resides.
- Before you configure remote users for a particular agent, configure the SNMP engine ID, using the **snmp-server engineID** global configuration command with the **remote** option. The remote agent's SNMP engine ID and user password are used to compute the authentication and privacy digests. If you do not configure the remote engine ID first, the configuration command fails.
- When configuring SNMP informs, you need to configure the SNMP engine ID for the remote agent in the SNMP database before you can send proxy requests or informs to it.
- If a local user is not associated with a remote host, the device does not send informs for the **auth** (authNoPriv) and the **priv** (authPriv) authentication levels.
- Changing the value of the SNMP engine ID has significant results. A user's password (entered on the command line) is converted to an MD5 or SHA security digest based on the password and the local engine ID. The command-line password is then destroyed, as required by RFC 2274. Because of this deletion, if the value of the engine ID changes, the security digests of SNMPv3 users become invalid, and you need to reconfigure SNMP users by using the **snmp-server user** *username* global configuration command. Similar restrictions require the reconfiguration of community strings when the engine ID changes.
- When you configure the SNMP server host with the default UDP port, 162, the output of the **show running-config** command does not display the UDP port value. If you specify a UDP port value other than the default by using the **snmp-server host** {*host-addr*} *community-string* **udp-port** *value* command, the UDP port number will be displayed in the **show running-config** command output. You can configure the **snmp-server host** command with or without the default UDP port 162; however, you cannot configure both simultaneously.

  The following examples are correct:

  ```
  Device(config)# snmp-server host 10.10.10.10 community udp-port 163
  Device(config)# snmp-server host 10.10.10.10 community

  Device(config)# snmp-server host 10.10.10.10 community udp-port 163
  Device(config)# snmp-server host 10.10.10.10 community udp-port 162
  ```

  The following examples are incorrect:

  ```
  Device(config)# snmp-server host 10.10.10.10 community udp-port 163
  Device(config)# snmp-server host 10.10.10.10 community
  Device(config)# snmp-server host 10.10.10.10 community udp-port 162

  Device(config)# snmp-server host 10.10.10.10 community udp-port 163
  Device(config)# snmp-server host 10.10.10.10 community udp-port 162
  Device(config)# snmp-server host 10.10.10.10 community
  ```

How to Configure SNMP

The following sections provide information on how to configure SNMP.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the device, the community string definitions on the NMS must match at least one of the three community string definitions on the device.

A community string can have one of the following attributes:

- Read-only (RO)—Gives all objects in the MIB except the community strings read access to authorized management stations, but does not allow write access.
- Read-write (RW)—Gives all objects in the MIB read and write access to authorized management stations, but does not allow access to the community strings.
- When a cluster is created, the command device manages the exchange of messages among member devices and the SNMP application. The Network Assistant software appends the member device number (@esN, where N is the device number) to the first configured RW and RO community strings on the command device and propagates them to the member devices.

Configuring SNMP Groups and Users

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the device. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

Follow these steps to configure SNMP groups and users on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*}
4. **snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
5. **snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*]} {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [*priv* {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*]
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**snmp-server engineID** {**local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string*}
	Example:
	`Device(config)# snmp-server engineID local 1234`
		Configures a name for either the local or remote copy of SNMP.
		• The *engineid-string* is a 24-character ID string with the name of the copy of SNMP. You need not specify the entire 24-character engine ID if it has trailing zeros. Specify only the portion of the engine ID up to the point where only zeros remain in the value. The Step Example configures an engine ID of 123400000000000000000000.
		• If you select **remote**, specify the *ip-address* of the device that contains the remote copy of SNMP and the optional User Datagram Protocol (UDP) port on the remote device. The default is 162.

Step 4	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] **Example:** `Device(config)# snmp-server group public v2c access lmnop`	Configures a new SNMP group on the remote device. For *group-name*, specify the name of the group. Specify one of the following security models: • **v1** is the least secure of the possible security models. • **v2c** is the second least secure model. It allows transmission of informs and integers twice the normal width. • **v3**, the most secure, requires you to select one of the following authentication levels: **auth**—Enables the Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) packet authentication. **noauth**—Enables the noAuthNoPriv security level. This is the default if no keyword is specified. **priv**—Enables Data Encryption Standard (DES) packet encryption (also called privacy). (Optional) Enter **read** *readview* with a string (not to exceed 64 characters) that is the name of the view in which you can only view the contents of the agent. (Optional) Enter **write** *writeview* with a string (not to exceed 64 characters) that is the name of the view in which you enter data and configure the contents of the agent.

	Command or Action	Purpose
		(Optional) Enter **notify** *notifyview* with a string (not to exceed 64 characters) that is the name of the view in which you specify a notify, inform, or trap. (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.
Step 5	**snmp-server user** *username group-name* {**remote** *host* [**udp-port** *port*] } {**v1** [**access** *access-list*] | **v2c** [**access** *access-list*] | **v3** [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*] } [*priv* {**des** | **3des** | **aes** {**128** | **192** | **256**} } *priv-password*] **Example:** `Device(config)#  snmp-server user Pat public v2c`	Adds a new user for an SNMP group. The *username* is the name of the user on the host that connects to the agent. The *group-name* is the name of the group to which the user is associated. Enter **remote** to specify a remote SNMP entity to which the user belongs and the hostname or IP address of that entity with the optional UDP port number. The default is 162. Enter the SNMP version number (**v1**, **v2c**, or **v3**). If you enter **v3**, you have these additional options: • **encrypted** specifies that the password appears in encrypted format. This keyword is available only when the **v3** keyword is specified. • **auth** is an authentication level setting session that can be either the HMAC-MD5-96 (**md5**) or the HMAC-SHA-96 (**sha**) authentication level and requires a password string *auth-password* (not to exceed 64 characters). If you enter **v3** you can also configure a private (**priv**) encryption algorithm and password string *priv-password* using the following keywords (not to exceed 64 characters): • **priv** specifies the User-based Security Model (USM). • **des** specifies the use of the 56-bit DES algorithm. • **3des** specifies the use of the 168-bit DES algorithm. • **aes** specifies the use of the DES algorithm. You must select either 128-bit, 192-bit, or 256-bit encryption. (Optional) Enter **access** *access-list* with a string (not to exceed 64 characters) that is the name of the access list.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

SNMP Notifications

SNMP allows the device to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword traps refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Note SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be resent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the device and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be resent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Therefore, traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the device is a concern and notification is not required, use traps.

Setting the Agent Contact and Location Information

Follow these steps to set the system contact and location of the SNMP agent so that these descriptions can be accessed through the configuration file.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server contact** *text*
4. **snmp-server location** *text*

5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server contact** *text* **Example:** `Device(config)# snmp-server contact Dial System Operator at beeper 21555`	Sets the system contact string.
Step 4	**snmp-server location** *text* **Example:** `Device(config)# snmp-server location Building 3/Room 222`	Sets the system location string.

Step 5	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Limiting TFTP Servers Used Through SNMP

Follow these steps to limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server tftp-server-list** *access-list-number*
4. **access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*]
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server tftp-server-list** *access-list-number* **Example:** `Device(config)# snmp-server tftp-server-list 44`	Limits the TFTP servers used for configuration file copies through SNMP to the servers in the access list. For *access-list-number*, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
Step 4	**access-list** *access-list-number* {**deny** | **permit**} *source* [*source-wildcard*] **Example:** `Device(config)# access-list 44 permit 10.1.1.2`	Creates a standard access list, repeating the command as many times as necessary. For *access-list-number*, enter the access list number specified in Step 3. The **deny** keyword denies access if the conditions are matched. The **permit** keyword permits access if the conditions are matched. For *source*, enter the IP address of the TFTP servers that can access the device. (Optional) For *source-wildcard*, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. The access list is always terminated by an implicit deny statement for everything.

	Command or Action	Purpose

Step 5	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Disabling the SNMP Agent

The **no snmp-server** global configuration command disables all running versions (Version 1, Version 2C, and Version 3) of the SNMP agent on the device and shuts down the SNMP process. You reenable all versions of the SNMP agent by entering one of the following commands in global configuration mode: **snmp-server host**, or **snmp-server user**, or **snmp-server community**, or **snmp-server manager**. There is no Cisco IOS command specifically designated for enabling SNMP.

Follow these steps to disable the SNMP agent.

Before you begin

The SNMP Agent must be enabled before it can be disabled. The SNMP agent is enabled by the first **snmp-server** global configuration command entered on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no snmp-server**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no snmp-server** **Example:** `Device(config)# no snmp-server`	Disables the SNMP agent operation.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string *public*. This configuration does not cause the device to send any traps.

```
Device(config)# snmp-server community public
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The device also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string *public* is sent with the traps.

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps vtp
Device(config)# snmp-server host 192.180.1.27 version 2c public
Device(config)# snmp-server host 192.180.1.111 version 1 public
Device(config)# snmp-server host 192.180.1.33 public
```

This example shows how to allow read-only access for all objects to members of access list 4 that use the *comaccess* community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2C to the host *cisco.com* using the community string *public*.

```
Device(config)# snmp-server community comaccess ro 4
Device(config)# snmp-server enable traps snmp authentication
Device(config)# snmp-server host cisco.com version 2c public
```

This example shows how to send Entity MIB traps to the host *cisco.com*. The community string is restricted. The first line enables the device to send Entity MIB traps in addition to any traps previously enabled. The second line specifies the destination of these traps and overwrites any previous **snmp-server** host commands for the host *cisco.com*.

```
Device(config)# snmp-server enable traps entity
Device(config)# snmp-server host cisco.com restricted entity
```

This example shows how to enable the device to send all traps to the host *myhost.cisco.com* using the community string *public*:

```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host myhost.cisco.com public
```

This example shows how to associate a user with a remote host and to send **auth** (authNoPriv) authentication-level informs when the user enters global configuration mode:

```
Device(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Device(config)# snmp-server group authgroup v3 auth
Device(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5 mypassword
Device(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Device(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Device(config)# snmp-server enable traps
Device(config)# snmp-server inform retries 0
```

This example shows how to display the entries of SNMP Managers polled to an SNMP Agent:

```
Device# show snmp stats host
Request Count                  Last Timestamp              Address
2                               00:00:01 ago               3.3.3.3
1                               1w2d ago                   2.2.2.2
```

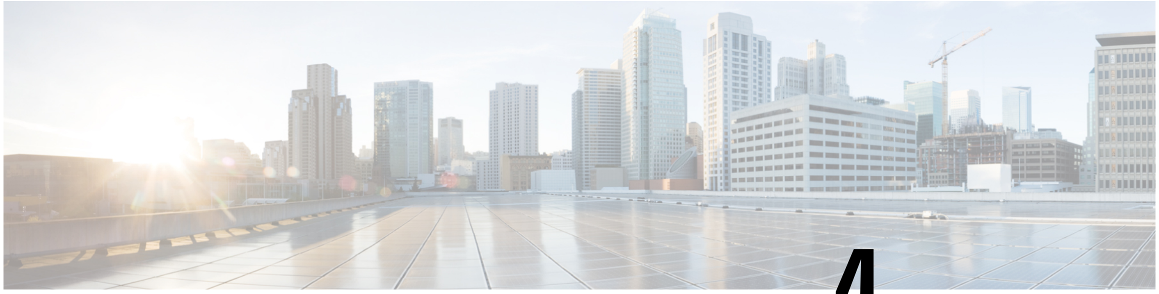
Monitoring SNMP Status

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command. You also can use the other privileged EXEC commands listed in the table to display SNMP information.

Table 3: Commands for Displaying SNMP Information

Command	Purpose
show snmp	Displays SNMP statistics.
	Displays information on the local SNMP engine and all remote e have been configured on the device.
show snmp group	Displays information on each SNMP group on the network.
show snmp pending	Displays information on pending SNMP requests.

Command	Purpose
show snmp sessions	Displays information on the current SNMP sessions.
show snmp user	Displays information on each SNMP user name in the SNMP users t **Note** You must use this command to display SNMPv3 config information for **auth** | **noauth** | **priv** mode. This informa displayed in the **show running-config** output.



CHAPTER 4

Configuring SPAN and RSPAN

- Prerequisites for SPAN and RSPAN, on page 35
- Restrictions for SPAN and RSPAN, on page 35
- Information About SPAN and RSPAN, on page 38
- Configuring SPAN and RSPAN, on page 46
- How to Configure SPAN and RSPAN, on page 47
- Monitoring SPAN and RSPAN Operations, on page 69
- Configuration Examples for SPAN and RSPAN, on page 69

Prerequisites for SPAN and RSPAN

SPAN

- You can limit SPAN traffic to specific VLANs by using the **filter vlan** keyword. If a trunk port is being monitored, only traffic on the VLANs specified with this keyword is monitored. By default, all VLANs are monitored on a trunk port.

RSPAN

- We recommend that you configure an RSPAN VLAN before you configure an RSPAN source or a destination session.

Restrictions for SPAN and RSPAN

SPAN

The restrictions for SPAN are as follows:

- On each device, you can configure 66 sessions. A maximum of 2 source sessions can be configured and the remaining sessions can be configured as RSPAN destinations sessions. A source session is either a local SPAN session or an RSPAN source session.

- For SPAN sources, you can monitor traffic for a single port or VLAN or a series or range of ports or VLANs for each session. You cannot mix source ports and source VLANs within a single SPAN session.

- The destination port cannot be a source port; a source port cannot be a destination port.
- A source port can be used in only one monitor session.
- You cannot have two SPAN sessions using the same destination port.
- When you configure a device port as a SPAN destination port, it is no longer a normal device port; only monitored traffic passes through the SPAN destination port.
- Entering SPAN configuration commands does not remove previously configured SPAN parameters. You must enter the **no monitor session** {*session_number* | **all** | **local** | **remote**} global configuration command to delete configured SPAN parameters.
- For local SPAN, outgoing packets through the SPAN destination port carry the original encapsulation headers—untagged, ISL, or IEEE 802.1Q—if the **encapsulation replicate** keywords are specified. If the keywords are not specified, the packets are sent in native form.
- You can configure a disabled port to be a source or destination port, but the SPAN function does not start until the destination port and at least one source port or source VLAN are enabled.
- You cannot mix source VLANs and filter VLANs within a single SPAN session.

Traffic monitoring in a SPAN session has the following restrictions:

- Sources can be ports or VLANs, but you cannot mix source ports and source VLANs in the same session.
- Wireshark does not capture egress packets when egress span is active.
- You can run both a local SPAN and an RSPAN source session in the same device or device stack. The device or device stack supports a total of 66 source and RSPAN destination sessions.
- Both switched and routed ports can be configured as SPAN sources and destinations.
- SPAN sessions do not interfere with the normal operation of the device. However, an oversubscribed SPAN destination, for example, a 10-Mb/s port monitoring a 100-Mb/s port, can result in dropped or lost packets.
- When SPAN or RSPAN is enabled, each packet being monitored is sent twice, once as normal traffic and once as a monitored packet. Monitoring a large number of ports or VLANs could potentially generate large amounts of network traffic.
- You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.
- The device does not support a combination of local SPAN and RSPAN in a single session.
    - An RSPAN source session cannot have a local destination port.
    - An RSPAN destination session cannot have a local source port.
    - An RSPAN destination session and an RSPAN source session that are using the same RSPAN VLAN cannot run on the same device or device stack.
- SPAN sessions capture only Dynamic Host Configuration Protocol (DHCP) ingress packets when DHCP snooping is enabled on the device.

RSPAN

The restrictions for RSPAN are as follows:

- IE31xx, IE3x00, and ESS3300 platforms support only one RSPAN session, due to a hardware limitation.
- Due to the processing chip's hardware limitation, it cannot differentiate between control packets and data traffic. Hence, if we enable mirroring, all packets, including control packets, are mirrored.

Caution Use the RSPAN feature cautiously, as it could impact your production network.

- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that the VLAN remote-span feature is supported in all the participating devices.
- You can configure the RSPAN VLAN on only one trunk interface. If you attempt to configure remote vlan on more than one trunk interface, the system displays an error, for example:

```
Switch(config-if)#do sh vlan id 2508

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
2508 VLAN2508                         active    Gi1/1, Gi1/2          >>>>>>>>>>>>

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
2508 enet  102508     1500  -      -      -        -    -        0      0

Remote SPAN VLAN
----------------
Enabled

Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------

Switch(config-if)#exit
Switch(config)#mon sess 1 destination remote vlan 2508
% Platform cannot support remote-span mirroring on VLAN with more than one member ports.
```

Note To prevent errors on the receiving interface of the RSPAN caused by the addition of the 4-byte RSPAN VLAN ID, it is recommended to set the MTU size to 4 bytes larger than the Maximum packet size monitored.

- The platform does not support RSPAN mirroring on a VLAN that is associated with a port-channel containing more than one member port.
- RSPAN VLANs are included as sources for port-based RSPAN sessions when source trunk ports have active RSPAN VLANs. RSPAN VLANs can also be sources in SPAN sessions. However, since the device does not monitor spanned traffic, it does not support egress spanning of packets on any RSPAN VLAN identified as the destination of an RSPAN source session on the device.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network for VLAN IDs that are lower than 1005.

- It is recommended not to configure RSPAN VLAN as Native VLAN.

Information About SPAN and RSPAN

The following sections provide information about SPAN and RSPAN.

SPAN and RSPAN

You can analyze network traffic passing through ports or VLANs by using SPAN or RSPAN to send a copy of the traffic to another port on the device or on another device that has been connected to a network analyzer or other monitoring or security device. SPAN copies (or mirrors) traffic received or sent (or both) on source ports or source VLANs to a destination port for analysis. SPAN does not affect the switching of network traffic on the source ports or VLANs. You must dedicate the destination port for SPAN use. Except for traffic that is required for the SPAN or RSPAN session, destination ports do not receive or forward traffic.

Only traffic that enters or leaves source ports or traffic that enters or leaves source VLANs can be monitored by using SPAN; traffic routed to a source VLAN cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another VLAN to the source VLAN cannot be monitored; however, traffic that is received on the source VLAN and routed to another VLAN can be monitored.

You can use the SPAN or RSPAN destination port to inject traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

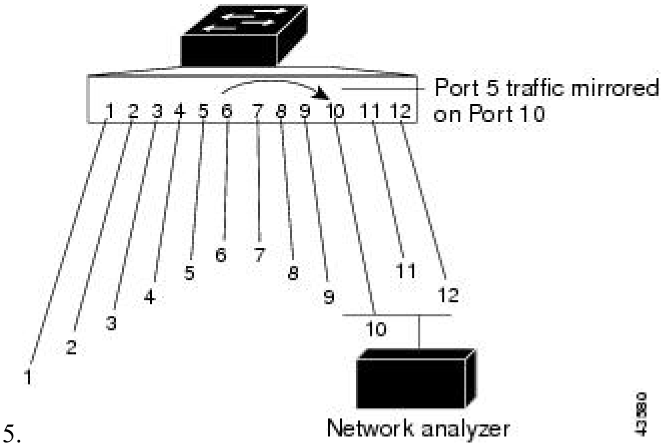
Local SPAN

Local SPAN supports a SPAN session entirely within one device; all source ports or source VLANs and destination ports are in the same device or device stack. Local SPAN copies traffic from one or more source ports in any VLAN or from one or more VLANs to a destination port for analysis.

Local SPAN supports a SPAN session entirely within one switch; all source ports and destination ports are in the same switch. Local SPAN copies traffic from one or more source ports to a destination port for analysis.

Figure 2: Example of Local SPAN Configuration on a Single Device

All traffic on port 5 (the source port) is mirrored to port 10 (the destination port). A network analyzer on port 10 receives all network traffic from port 5 without being physically attached to port



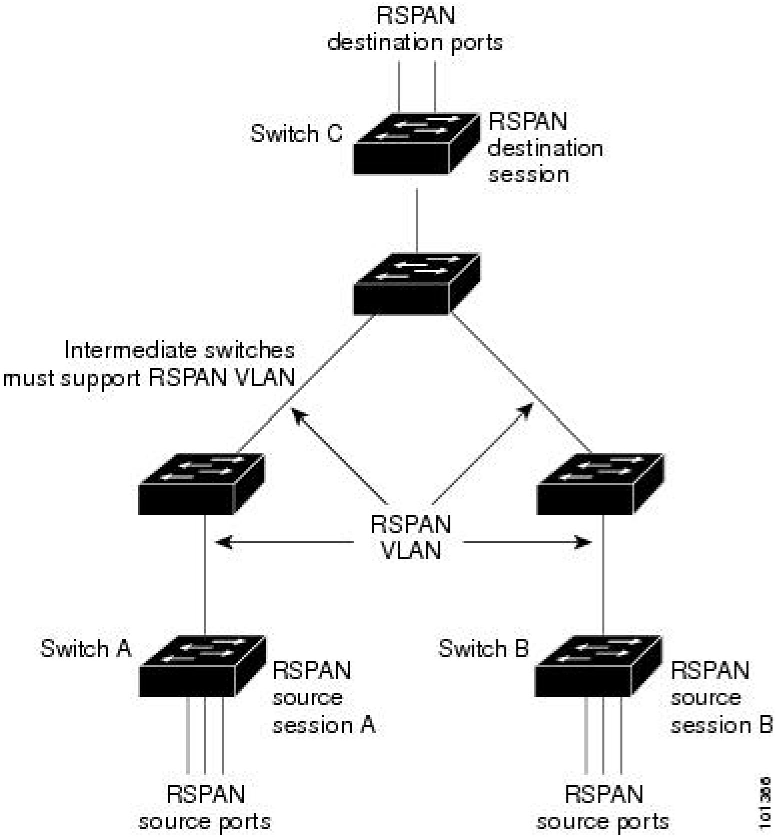
5.

Remote SPAN

RSPAN supports source ports, source VLANs, and destination ports on different devices (or different device stacks), enabling remote monitoring of multiple devices across your network.

Figure 3: Example of RSPAN Configuration

The figure below shows source ports on Device A and Device B. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating devices. The RSPAN traffic from the source ports or VLANs is copied into the RSPAN VLAN and forwarded over trunk ports carrying the RSPAN VLAN to a destination session monitoring the RSPAN VLAN. Each RSPAN source device must have either ports or VLANs as RSPAN sources. The destination is always a physical port, as shown on Device C in the figure.



SPAN and RSPAN Concepts and Terminology

SPAN Sessions

SPAN sessions (local or remote) allow you to monitor traffic on a port, or one or more VLANs, and send the monitored traffic to one or more destination ports.

A local SPAN session is an association of a destination port with source ports or source VLANs, all on a single network device. Local SPAN does not have separate source and destination sessions. Local SPAN sessions gather a set of ingress and egress packets specified by the user and form them into a stream of SPAN data, which is directed to the destination port.

RSPAN consists of at least one RSPAN source session, an RSPAN VLAN, and at least one RSPAN destination session. You separately configure RSPAN source sessions and RSPAN destination sessions on different network devices. To configure an RSPAN source session on a device, you associate a set of source ports or source VLANs with an RSPAN VLAN. The output of this session is the stream of SPAN packets that are sent to the RSPAN VLAN. To configure an RSPAN destination session on another device, you associate the destination port with the RSPAN VLAN. The destination session collects all RSPAN VLAN traffic and sends it out the RSPAN destination port.

An RSPAN source session is very similar to a local SPAN session, except for where the packet stream is directed. In an RSPAN source session, SPAN packets are relabeled with the RSPAN VLAN ID and directed over normal trunk ports to the destination device.

An RSPAN destination session takes all packets received on the RSPAN VLAN, strips off the VLAN tagging, and presents them on the destination port. The session presents a copy of all RSPAN VLAN packets (except Layer 2 control packets) to the user for analysis.

Traffic monitoring in a SPAN session has these restrictions:

- Sources can be ports or VLANs, but you cannot mix source ports and source VLANs in the same session.
- SPAN sessions do not interfere with the normal operation of the device. However, an oversubscribed SPAN destination, for example, a 10-Mb/s port monitoring a 100-Mb/s port, can result in dropped or lost packets.
- When SPAN or RSPAN is enabled, each packet being monitored is sent twice, once as normal traffic and once as a monitored packet. Therefore monitoring a large number of ports or VLANs could potentially generate large amounts of network traffic.
- You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.
- The device does not support a combination of local SPAN and RSPAN in a single session.
    - An RSPAN source session cannot have a local destination port.
    - An RSPAN destination session cannot have a local source port.
    - An RSPAN destination session and an RSPAN source session that are using the same RSPAN VLAN cannot run on the same device or device stack.

Monitored Traffic

SPAN sessions can monitor these traffic types:

- Receive (Rx) SPAN—Receive (or ingress) SPAN monitors as much as possible all of the packets received by the source interface or VLAN before any modification or processing is performed by the device. A copy of each packet received by the source is sent to the destination port for that SPAN session.

  Packets that are modified because of routing or Quality of Service (QoS)—for example, modified Differentiated Services Code Point (DSCP)—are copied before modification.

  Features that can cause a packet to be dropped during receive processing have no effect on ingress SPAN; the destination port receives a copy of the packet even if the actual incoming packet is dropped. These features include IP standard and extended input Access Control Lists (ACLs), ingress QoS policing, VLAN ACLs, and egress QoS policing.
- Transmit (Tx) SPAN—Transmit (or egress) SPAN monitors as much as possible all of the packets sent by the source interface after all modification and processing is performed by the device. A copy of each

packet sent by the source is sent to the destination port for that SPAN session. The copy is provided after the packet is modified.

Packets that are modified because of routing (for example, with modified time-to-live (TTL), MAC address, or QoS values) are duplicated (with the modifications) at the destination port.

Features that can cause a packet to be dropped during transmit processing also affect the duplicated copy for SPAN. These features include IP standard and extended output ACLs and egress QoS policing.

- Both—In a SPAN session, you can also monitor a port or VLAN for both received and sent packets. This is the default.

Therefore, a local SPAN session with encapsulation replicate enabled can have a mixture of untagged and IEEE 802.1Q tagged packets appear on the destination port.

Device congestion can cause packets to be dropped at ingress source ports, egress source ports, or SPAN destination ports. In general, these characteristics are independent of one another. For example:

- A packet might be forwarded normally but dropped from monitoring due to an oversubscribed SPAN destination port.
- An ingress packet might be dropped from normal forwarding, but still appear on the SPAN destination port.
- An egress packet dropped because of device congestion is also dropped from egress SPAN.

In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination port. For example, a bidirectional (both Rx and Tx) SPAN session is configured for the Rx monitor on port A and Tx monitor on port B. If a packet enters the device through port A and is switched to port B, both incoming and outgoing packets are sent to the destination port. Both packets are the same unless a Layer 3 rewrite occurs, in which case the packets are different because of the packet modification.

Source Ports

A source port (also called a monitored port) is a switched or routed port that you monitor for network traffic analysis.

In a local SPAN session or RSPAN source session, you can monitor source ports or VLANs for traffic in one or both directions.

The device supports any number of source ports (up to the maximum number of available ports on the device) and any number of source VLANs (up to the maximum number of VLANs supported).

You cannot mix ports and VLANs in a single session.

A source port has these characteristics:

- A source port can be used in only one monitor session.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor.
- It can be any port type (for example, EtherChannel, Gigabit Ethernet, and so forth).
- For EtherChannel sources, you can monitor traffic for the entire EtherChannel or individually on a physical port as it participates in the port channel.
- It can be an access port, trunk port, routed port, or voice VLAN port.
- It cannot be a destination port.

- Source ports can be in the same or different VLANs.
- You can monitor multiple source ports in a single session.

Source VLANs

VLAN-based SPAN (VSPAN) is the monitoring of the network traffic in one or more VLANs. The SPAN or RSPAN source interface in VSPAN is a VLAN ID, and traffic is monitored on all the ports for that VLAN.

VSPAN has these characteristics:

- All active ports in the source VLAN are included as source ports and can be monitored in either or both directions.
- On a given port, only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.
- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.

VLAN Filtering

When you monitor a trunk port as a source port, by default, all VLANs active on the trunk are monitored. You can limit SPAN traffic monitoring on trunk source ports to specific VLANs by using VLAN filtering.

- VLAN filtering applies only to trunk ports or to voice VLAN ports.
- VLAN filtering applies only to port-based sessions and is not allowed in sessions with VLAN sources.
- When a VLAN filter list is specified, only those VLANs in the list are monitored on trunk ports or on voice VLAN access ports.
- SPAN traffic coming from other port types is not affected by VLAN filtering; that is, all VLANs are allowed on other ports.
- VLAN filtering affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic.

Destination Port

Each local SPAN session or RSPAN destination session must have a destination port (also called a monitoring port) that receives a copy of traffic from the source ports or VLANs and sends the SPAN packets to the user, usually a network analyzer.

A destination port has these characteristics:

- For a local SPAN session, the destination port must reside on the same device or device stack as the source port. For an RSPAN session, it is located on the device containing the RSPAN destination session. There is no destination port on a device or device stack running only an RSPAN source session.
- When a port is configured as a SPAN destination port, the configuration overwrites the original port configuration. When the SPAN destination configuration is removed, the port reverts to its previous

configuration. If a configuration change is made to the port while it is acting as a SPAN destination port, the change does not take effect until the SPAN destination configuration had been removed.

Note When QoS is configured on the SPAN destination port, QoS takes effect immediately.

- If the port was in an EtherChannel group, it is removed from the group while it is a destination port. If it was a routed port, it is no longer a routed port.
- It can be any Ethernet physical port.
- It cannot be a secure port.
- It cannot be a source port.
- It can participate in only one SPAN session at a time (a destination port in one SPAN session cannot be a destination port for a second SPAN session).
- When it is active, incoming traffic is disabled. The port does not transmit any traffic except that required for the SPAN session. Incoming traffic is never learned or forwarded on a destination port.
- If ingress traffic forwarding is enabled for a network security device, the destination port forwards traffic at Layer 2.
- It does not participate in any of the Layer 2 protocols (STP, VTP, CDP, DTP, PagP).
- A destination port that belongs to a source VLAN of any SPAN session is excluded from the source list and is not monitored.
- The maximum number of destination ports in a device or device stack is 64.

Local SPAN and RSPAN destination ports function differently with VLAN tagging and encapsulation:

- For local SPAN, if the **encapsulation replicate** keywords are specified for the destination port, these packets appear with the original encapsulation (untagged, ISL, or IEEE 802.1Q). If these keywords are not specified, packets appear in the untagged format. Therefore, the output of a local SPAN session with **encapsulation replicate** enabled can contain a mixture of untagged, ISL, or IEEE 802.1Q-tagged packets.
- For RSPAN, the RSPAN VLAN ID is inserted into the packet while preserving the original VLAN ID.

RSPAN VLAN

The RSPAN VLAN carries SPAN traffic between RSPAN source and destination sessions. RSPAN VLAN has these special characteristics:

- All traffic in the RSPAN VLAN is always flooded.
- No MAC address learning occurs on the RSPAN VLAN.
- RSPAN VLAN traffic only flows on trunk ports.
- RSPAN VLANs must be configured in VLAN configuration mode by using the **remote-span** VLAN configuration mode command.
- STP can run on RSPAN VLAN trunks but not on SPAN destination ports.

- An RSPAN VLAN cannot be a private-VLAN primary or secondary VLAN.

For VLANs 1 to 1005 that are visible to VLAN Trunking Protocol (VTP), the VLAN ID and its associated RSPAN characteristic are propagated by VTP. If you assign an RSPAN VLAN ID in the extended VLAN range (1006 to 4094), you must manually configure all intermediate devices.

It is normal to have multiple RSPAN VLANs in a network at the same time with each RSPAN VLAN defining a network-wide RSPAN session. That is, multiple RSPAN source sessions anywhere in the network can contribute packets to the RSPAN session. It is also possible to have multiple RSPAN destination sessions throughout the network, monitoring the same RSPAN VLAN and presenting traffic to the user. The RSPAN VLAN ID separates the sessions.

SPAN and RSPAN Interaction with Other Features

SPAN interacts with these features:

- Routing—SPAN does not monitor routed traffic. VSPAN only monitors traffic that enters or exits the , not traffic that is routed between VLANs. For example, if a VLAN is being Rx-monitored and the routes traffic from another VLAN to the monitored VLAN, that traffic is not monitored and not received on the SPAN destination port.

- STP—A destination port does not participate in STP while its SPAN or RSPAN session is active. The destination port can participate in STP after the SPAN or RSPAN session is disabled. On a source port, SPAN does not affect the STP status. STP can be active on trunk ports carrying an RSPAN VLAN.

- CDP—A SPAN destination port does not participate in CDP while the SPAN session is active. After the SPAN session is disabled, the port again participates in CDP.

- VTP—You can use VTP to prune an RSPAN VLAN between .

- VLAN and trunking—You can modify VLAN membership or trunk settings for source or destination ports at any time. However, changes in VLAN membership or trunk settings for a destination port do not take effect until you remove the SPAN destination configuration. Changes in VLAN membership or trunk settings for a source port immediately take effect, and the respective SPAN sessions automatically adjust accordingly.

- EtherChannel—You can configure an EtherChannel group as a source port a SPAN destination port. When a group is configured as a SPAN source, the entire group is monitored.

  If a physical port is added to a monitored EtherChannel group, the new port is added to the SPAN source port list. If a port is removed from a monitored EtherChannel group, it is automatically removed from the source port list.

  A physical port that belongs to an EtherChannel group can be configured as a SPAN source port and still be a part of the EtherChannel. In this case, data from the physical port is monitored as it participates in the EtherChannel. However, if a physical port that belongs to an EtherChannel group is configured as a SPAN destination, it is removed from the group. After the port is removed from the SPAN session, it rejoins the EtherChannel group. Ports removed from an EtherChannel group remain members of the group, but they are in the inactive or suspended state.

  If a physical port that belongs to an EtherChannel group is a destination port and the EtherChannel group is a source, the port is removed from the EtherChannel group and from the list of monitored ports.

- Multicast traffic can be monitored. For egress and ingress port monitoring, only a single unedited packet is sent to the SPAN destination port. It does not reflect the number of times the multicast packet is sent.

- A private-VLAN port cannot be a SPAN destination port.

- A secure port cannot be a SPAN destination port.

  For SPAN sessions, do not enable port security on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable port security on any ports with monitored egress.

- An IEEE 802.1x port can be a SPAN source port. You can enable IEEE 802.1x on a port that is a SPAN destination port; however, IEEE 802.1x is disabled until the port is removed as a SPAN destination.

  For SPAN sessions, do not enable IEEE 802.1x on ports with monitored egress when ingress forwarding is enabled on the destination port. For RSPAN source sessions, do not enable IEEE 802.1x on any ports that are egress monitored.

SPAN and RSPAN and Device Stacks

Because the stack of represents one logical , local SPAN source ports and destination ports can be in different in the stack. Therefore, the addition or deletion of in the stack can affect a local SPAN session, as well as an RSPAN source or destination session. An active session can become inactive when a is removed from the stack or an inactive session can become active when a is added to the stack.

Flow-Based SPAN

You can control the type of network traffic to be monitored in SPAN or RSPAN sessions by using flow-based SPAN (FSPAN) or flow-based RSPAN (FRSPAN), which apply access control lists (ACLs) to the monitored traffic on the source ports. The FSPAN ACLs can be configured to filter IPv4, IPv6, and non-IP monitored traffic.

You apply an ACL to a SPAN session through the interface. It is applied to all the traffic that is monitored on all interfaces in the SPAN session.The packets that are permitted by this ACL are copied to the SPAN destination port. No other packets are copied to the SPAN destination port.

The original traffic continues to be forwarded, and any port, VLAN, and router ACLs attached are applied. The FSPAN ACL does not have any effect on the forwarding decisions. Similarly, the port, VLAN, and router ACLs do not have any effect on the traffic monitoring. If a security input ACL denies a packet and it is not forwarded, the packet is still copied to the SPAN destination ports if the FSPAN ACL permits it. But if the security output ACL denies a packet and it is not sent, it is not copied to the SPAN destination ports. However, if the security output ACL permits the packet to go out, it is only copied to the SPAN destination ports if the FSPAN ACL permits it. This is also true for an RSPAN session.

You can attach three types of FSPAN ACLs to the SPAN session:

- IPv4 FSPAN ACL— Filters only IPv4 packets.
- IPv6 FSPAN ACL— Filters only IPv6 packets.
- MAC FSPAN ACL— Filters only non-IP packets.

If a VLAN-based FSPAN session configured on a stack cannot fit in the hardware memory on one or more devices, it is treated as unloaded on those devices, and traffic meant for the FSPAN ACL and sourcing on that device is not copied to the SPAN destination ports. The FSPAN ACL continues to be correctly applied, and traffic is copied to the SPAN destination ports on the devices where the FSPAN ACL fits in the hardware memory.

When an empty FSPAN ACL is attached, some hardware functions copy all traffic to the SPAN destination ports for that ACL. If sufficient hardware resources are not available, even an empty FSPAN ACL can be unloaded.

Default SPAN and RSPAN Configuration

Table 4: Default SPAN and RSPAN Configuration

Feature	Default Setting
SPAN state (SPAN and RSPAN)	Disabled.
Source port traffic to monitor	Both received and sent traffic (**both**).
Encapsulation type (destination port)	Native form (untagged packets).
Ingress forwarding (destination port)	Disabled.
VLAN filtering	On a trunk interface used as a source port, all VLANs are monitored.
RSPAN VLANs	None configured.

Configuring SPAN and RSPAN

SPAN Configuration Guidelines

- To remove a source or destination port or VLAN from the SPAN session, use the **no monitor session** *session_number* **source interface** *interface-id* {**interface** *interface-id* | **vlan** *vlan-id*} global configuration command or the **no monitor session** *session_number* **destination interface** *interface-id* global configuration command. For destination interfaces, the **encapsulation** options are ignored with the **no** form of the command.
- To monitor all VLANs on the trunk port, use the **no monitor session** *session_number* **filter** global configuration command.

RSPAN Configuration Guidelines

- All the SPAN configuration guidelines apply to RSPAN.
- As RSPAN VLANs have special properties, you should reserve a few VLANs across your network for use as RSPAN VLANs; do not assign access ports to these VLANs.
- You can apply an output ACL to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source .
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple in your network.
- Access ports (including voice VLAN ports) on the RSPAN VLAN are put in the inactive state.
- You can configure any VLAN as an RSPAN VLAN as long as these conditions are met:
    - The same RSPAN VLAN is used for an RSPAN session in all the .

- All participating support RSPAN.

FSPAN and FRSPAN Configuration Guidelines

- When at least one FSPAN ACL is attached, FSPAN is enabled.
- When you attach at least one FSPAN ACL that is not empty to a SPAN session, and you have not attached one or more of the other FSPAN ACLs (for instance, you have attached an IPv4 ACL that is not empty, and have not attached IPv6 and MAC ACLs), FSPAN blocks the traffic that would have been filtered by the unattached ACLs. Therefore, this traffic is not monitored.

How to Configure SPAN and RSPAN

The following sections provide information on how to configure SPAN and RSPAN.

Creating a Local SPAN Session

Follow these steps to create a SPAN session and specify the source (monitored) ports or VLANs and the destination (monitoring) ports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**][**ingress**{**vlan** *vlan-id*}] }
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}	Removes any existing SPAN configuration for the session.

	Command or Action	Purpose
	Example: `Device(config)# no monitor session all`	• For *session_number*, the range is 1 to 66. • **all**—Removes all SPAN sessions. • **local**—Removes all local sessions. • **remote**—Removes all remote SPAN sessions.
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Device(config)# monitor session 1 source interface gigabitethernet1/1`	Specifies the SPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • For *interface-id*, specify the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • **Note** A single session can include multiple sources (ports or VLANs) defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx**—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. • **both**—Monitors both received and sent traffic. • **rx**—Monitors received traffic. • **tx**—Monitors sent traffic. **Note** You can use the **monitor session** *session_number* **source** command multiple times to configure multiple source ports.

Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**][**ingress**{**vlan** *vlan-id*}] }
	Example:
	`Device(config)# monitor session 1 destination`
		Note For local SPAN, you must use the same session number for the source and destination interfaces.
		• For *session_number*, specify the session number entered in step 4.

	Command or Action	Purpose
	`interface gigabitethernet1/2 encapsulation replicate ingress vlan 10`	• For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). • **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type: • **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN. **Note** You can use **monitor session** *session_number* **destination** command multiple times to configure multiple destination ports.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating a Local SPAN Session and Configuring Incoming Traffic

Follow these steps to create a SPAN session, to specify the source ports or VLANs and the destination ports, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress**{**vlan** *vlan-id*}]}
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device#` **`configure terminal`**	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)#` **`no monitor session all`**
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
	Example:
	`Device(config)#` **`monitor session 2 source`**
	`gigabitethernet1/1 rx`
		Specifies the SPAN session and the source port (monitored port).

Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress**{**vlan** *vlan-id*}]}
	Example:
	`Device(config)#` **`monitor session 2 destination`**
	`interface gigabitethernet1/2 encapsulation`
	`replicate ingress vlan 10`
		Specifies the SPAN session, the destination port, the packet encapsulation, and the ingress VLAN and encapsulation.
		• For *session_number*, specify the session number entered in Step 4.
		• For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN.

	Command or Action	Purpose
		• (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma or hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). • **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type: • **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Specifying VLANs to Filter

Follow these steps to limit SPAN source traffic to specific VLANs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**]
6. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**][**ingress**{**vlan** *vlan-id*}]}
7. **end**

8. **show running-config**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session all`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Device(config)# monitor session 2 source interface gigabitethernet1/2 rx`	Specifies the characteristics of the source port (monitored port) and SPAN session. • For *session_number*, the range is 1 to 66. • For *interface-id*, specify the source port to monitor. The interface specified must already be configured as a trunk port.
Step 5	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] **Example:** `Device(config)# monitor session 2 filter vlan 1 - 5 , 9`	Limits the SPAN source traffic to specific VLANs. • For *session_number*, enter the session number specified in Step 4. • For *vlan-id*, the range is 1 to 4094. • (Optional) Use a comma (**,**) to specify a series of VLANs, or use a hyphen (**-**) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.
Step 6	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**][**ingress**{**vlan** *vlan-id*}]} **Example:**	Specifies the SPAN session and the destination port (monitoring port). • For *session_number*, specify the session number entered in Step 4.

	Command or Action	Purpose
	`Device(config)# monitor session 2 destination interface gigabitethernet1/1 ingress vlan 6`	- For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN. - (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type: - **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

Step 7	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a VLAN as an RSPAN VLAN

Follow these steps to create a new VLAN, then configure it to be the RSPAN VLAN for the RSPAN session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **remote-span**
5. **end**

6. **show running-config**
7. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vlan** *vlan-id* **Example:** `Device(config)# vlan 100`	Enters a VLAN ID to create a VLAN, or enters the VLAN ID of an existing VLAN, and enters VLAN configuration mode. The range is 2 to 1001 and 1006 to 4094. The RSPAN VLAN cannot be VLAN 1 (the default VLAN) or VLAN IDs 1002 through 1005 (reserved for Token Ring and FDDI VLANs).
Step 4	**remote-span** **Example:** `Device(config-vlan)# remote-span`	Configures the VLAN as an RSPAN VLAN.

Step 5	**end**
	Example:
	`Device(config-vlan)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

You must create the RSPAN VLAN in all devices that will participate in RSPAN. If the RSPAN VLAN-ID is in the normal range (lower than 1005) and VTP is enabled in the network, you can create the RSPAN VLAN in one device, and VTP propagates it to the other devices in the VTP domain. For extended-range VLANs (greater than 1005), you must configure RSPAN VLAN on both source and destination devices and any intermediate devices.

Use VTP pruning to get an efficient flow of RSPAN traffic, or manually delete the RSPAN VLAN from all trunks that do not need to carry the RSPAN traffic.

To remove the remote SPAN characteristic from a VLAN and convert it back to a normal VLAN, use the **no remote-span** VLAN configuration command.

To remove a source port or VLAN from the SPAN session, use the **no monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} global configuration command. To remove the RSPAN VLAN from the session, use the **no monitor session** *session_number* **destination remote vlan** *vlan-id*.

Creating an RSPAN Source Session

Follow these steps to create and start an RSPAN source session and to specify the monitored source and the destination RSPAN VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination remote vlan** *vlan-id*
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session 1`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

	Command or Action	Purpose
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Device(config)# monitor session 1 source interface gigabitethernet1/1 tx`	Specifies the RSPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • Enter a source port or source VLAN for the RSPAN session: • For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • A single session can include multiple sources (ports or VLANs), defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx**—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. • **both**—Monitors both received and sent traffic. • **rx**—Monitors received traffic. • **tx**—Monitors sent traffic.
Step 5	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Device(config)# monitor session 1 destination remote vlan 100`	Specifies the RSPAN session, the destination RSPAN VLAN, and the destination-port group. • For *session_number*, enter the number defined in Step 4. • For *vlan-id*, specify the source RSPAN VLAN to monitor.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 7	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	`Device# show running-config`	
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Specifying VLANs to Filter

Follow these steps to configure the RSPAN source session to limit RSPAN source traffic to specific VLANs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**]
6. **monitor session** *session_number* **destination remote vlan** *vlan-id*
7. **end**
8. **show running-config**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session 2`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

	Command or Action	Purpose
Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Device(config)# monitor session 2 source interface gigabitethernet1/2 rx`	Specifies the characteristics of the source port (monitored port) and SPAN session. • For *session_number*, the range is 1 to 66. • For *interface-id*, specify the source port to monitor. The interface specified must already be configured as a trunk port. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **both** | **rx** | **tx**—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. • **both**—Monitors both received and sent traffic. • **rx**—Monitors received traffic. • **tx**—Monitors sent traffic.
Step 5	**monitor session** *session_number* **filter vlan** *vlan-id* [**,** | **-**] **Example:** `Device(config)# monitor session 2 filter vlan 1 - 5 , 9`	Limits the SPAN source traffic to specific VLANs. • For *session_number*, enter the session number specified in step 4. • For *vlan-id*, the range is 1 to 4094. • (Optional) **,** | **-** Use a comma (**,**) to specify a series of VLANs or use a hyphen (**-**) to specify a range of VLANs. Enter a space before and after the comma; enter a space before and after the hyphen.
Step 6	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Device(config)# monitor session 2 destination remote vlan 902`	Specifies the RSPAN session and the destination remote VLAN (RSPAN VLAN). • For *session_number*, enter the session number specified in Step 4. • For *vlan-id*, specify the RSPAN VLAN to carry the monitored traffic to the destination port.

Step 7	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating an RSPAN Destination Session

You configure an RSPAN destination session on a different device or device stack; that is, not the device or device stack on which the source session was configured.

Follow these steps to define the RSPAN VLAN on that device, to create an RSPAN destination session, and to specify the source RSPAN VLAN and the destination port.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vlan** *vlan-id*
4. **remote-span**
5. **exit**
6. **no monitor session** {*session_number* | **all** | **local** | **remote**}
7. **monitor session** *session_number* **source remote vlan** *vlan-id*
8. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**][**ingress**{**vlan** *vlan-id*}] }
9. **end**
10. **show running-config**
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**vlan** *vlan-id* **Example:** `Device(config)# vlan 901`	Specifies the VLAN ID of the RSPAN VLAN created from the source device, and enters VLAN configuration mode. If both devices are participating in VTP and the RSPAN VLAN ID is from 2 to 1005, Steps 3 through 5 are not required because the RSPAN VLAN ID is propagated through the VTP network.
Step 4	**remote-span** **Example:** `Device(config-vlan)# remote-span`	Identifies the VLAN as the RSPAN VLAN.
Step 5	**exit** **Example:** `Device(config-vlan)# exit`	Returns to global configuration mode.

Step 6	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session 1`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

Step 7	**monitor session** *session_number* **source remote vlan** *vlan-id* **Example:** `Device(config)# monitor session 1 source remote vlan 901`	Specifies the RSPAN session and the source RSPAN VLAN. • For *session_number*, the range is 1 to 66. • For *vlan-id*, specify the source RSPAN VLAN to monitor.

Step 8	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**][**ingress**{**vlan** *vlan-id*}] }
	Example:
	`Device(config)# monitor session 1 destination interface gigabitethernet1/2 encapsulation replicate ingress vlan 10`
		Specifies the RSPAN session and the destination interface.
		• For *session_number*, enter the number defined in Step 7.
		In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port.
		• For *interface-id*, specify the destination interface. The destination interface must be a physical interface.

	Command or Action	Purpose
		• (Optional) **[, | -]** Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged). • **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type: • **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN. • You can use **monitor session** *session_number* **destination** command multiple times to configure multiple destination ports.

Step 9	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 10	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Creating an RSPAN Destination Session and Configuring Incoming Traffic

Follow these steps to create an RSPAN destination session, to specify the source RSPAN VLAN and the destination port, and to enable incoming traffic on the destination port for a network security device (such as a Cisco IDS Sensor Appliance).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}

4. **monitor session** *session_number* **source remote vlan** *vlan-id*
5. **monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress**{**vlan** *vlan-id*}]}
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session 2`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

Step 4	**monitor session** *session_number* **source remote vlan** *vlan-id* **Example:** `Device(config)# monitor session 2 source remote vlan 901`	Specifies the RSPAN session and the source RSPAN VLAN. • For *session_number*, the range is 1 to 66. • For *vlan-id*, specify the source RSPAN VLAN to monitor.

Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress**{**vlan** *vlan-id*}]}
	Example:
	`Device(config)# monitor session 2 destination interface gigabitethernet1/2 encapsulation replicate ingress vlan 10`
		Specifies the SPAN session, the destination port, the packet encapsulation, and the ingress VLAN and encapsulation.
		• For *session_number*, enter the number defined in Step 5.
		In an RSPAN destination session, you must use the same session number for the source RSPAN VLAN and the destination port.
		• For *interface-id*, specify the destination interface. The destination interface must be a physical interface.
		• Though visible in the command-line help string, **encapsulation replicate** is not supported for RSPAN.

	Command or Action	Purpose
		The original VLAN ID is overwritten by the RSPAN VLAN ID, and all packets appear on the destination port as untagged. • (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • Enter **ingress** with additional keywords to enable forwarding of incoming traffic on the destination port and to specify the encapsulation type: • **vlan** *vlan-id*—Forwards incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.

Step 6	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring an FSPAN Session

Follow these steps to create a SPAN session, specify the source (monitored) ports or VLANs and the destination (monitoring) ports, and configure FSPAN for the session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [, | -] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination** {**interface** *interface-id* [, | -] [**encapsulation replicate**] [**ingress**{**vlan** *vlan-id*}]}
6. **monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}

7. **end**
8. **show running-config**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session 2`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Device(config)# monitor session 2 source interface gigabitethernet1/1`	Specifies the SPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) [**both** | **rx** | **tx**]—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the SPAN monitors both sent and received traffic. • **both**—Monitors both sent and received traffic. This is the default. • **rx**—Monitors received traffic. • **tx**—Monitors sent traffic.

	Command or Action	Purpose
		Note You can use the **monitor session** *session_number* **source** command multiple times to configure multiple source ports.

Step 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress**{**vlan** *vlan-id*}]}
	Example:
	`Device(config)# monitor session 2 destination interface gigabitethernet1/2 encapsulation replicate ingress vlan 50`
		Specifies the SPAN session and the destination port (monitoring port).
		- For *session_number*, specify the session number entered in Step 4.
		- For **destination**, specify the following parameters:
		  - For *interface-id*, specify the destination port. The destination interface must be a physical port; it cannot be an EtherChannel, and it cannot be a VLAN.
		  - (Optional) [**,** | **-**] Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen.
		  - (Optional) **encapsulation replicate** specifies that the destination interface replicates the source interface encapsulation method. If not selected, the default is to send packets in native form (untagged).
		  - **ingress** enables forwarding of incoming traffic on the destination port and to specify the encapsulation type:
		    - **vlan** *vlan-id*—Accepts incoming packets with untagged encapsulation type with the specified VLAN as the default VLAN.
		Note For local SPAN, you must use the same session number for the source and destination interfaces.
		You can use **monitor session** *session_number* **destination** command multiple times to configure multiple destination ports.

Step 6	**monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
	Example:
	`Device(config)# monitor session 2 filter ipv6 access-group 4`
		Specifies the SPAN session, the types of packets to filter, and the ACLs to use in an FSPAN session.
		- For *session_number*, specify the session number entered in Step 4.

	Command or Action	Purpose
		• For *access-list-number*, specify the ACL number that you want to use to filter traffic. • For *name*, specify the ACL name that you want to use to filter traffic.

Step 7	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring an FRSPAN Session

Follow these steps to start an RSPAN source session, specify the monitored source and the destination RSPAN VLAN, and configure FRSPAN for the session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination remote vlan** *vlan-id*
6. **vlan** *vlan-id*
7. **remote-span**
8. **exit**
9. **monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
10. **end**
11. **show running-config**
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**no monitor session** {*session_number* | **all** | **local** | **remote**}
	Example:
	`Device(config)# no monitor session 2`
		Removes any existing SPAN configuration for the session.
		• For *session_number*, the range is 1 to 66.
		• **all**—Removes all SPAN sessions.
		• **local**—Removes all local sessions.
		• **remote**—Removes all remote SPAN sessions.

Step 4	**monitor session** *session_number* **source** {**interface** *interface-id* } [**,** | **-**] [**both** | **rx** | **tx**] **Example:** `Device(config)# monitor session 2 source interface gigabitethernet1/1`	Specifies the SPAN session and the source port (monitored port). • For *session_number*, the range is 1 to 66. • For *interface-id*, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (**port-channel** *port-channel-number*). Valid port-channel numbers are 1 to 48. • For *vlan-id*, specify the source VLAN to monitor. The range is 1 to 4094 (excluding the RSPAN VLAN). **Note** A single session can include multiple sources (ports or VLANs) defined in a series of commands, but you cannot combine source ports and source VLANs in one session. • (Optional) [**,** | **-**]—Specifies a series or range of interfaces. Enter a space before and after the comma; enter a space before and after the hyphen. • (Optional) [**both** | **rx** | **tx**]—Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the SPAN monitors both sent and received traffic. • **both**—Monitors both sent and received traffic. This is the default.

	Command or Action	Purpose
		• **rx**—Monitors received traffic. • **tx**—Monitors sent traffic. **Note** You can use the **monitor session** *session_number* **source** command multiple times to configure multiple source ports.
Step 5	**monitor session** *session_number* **destination remote vlan** *vlan-id* **Example:** `Device(config)#` **`monitor session 2 destination remote vlan 5`**	Specifies the RSPAN session and the destination RSPAN VLAN. • For *session_number*, enter the number defined in Step 4. • For *vlan-id*, specify the destination RSPAN VLAN to monitor.
Step 6	**vlan** *vlan-id* **Example:** `Device(config)#` **`vlan 10`**	Enters the VLAN configuration mode. For *vlan-id*, specify the source RSPAN VLAN to monitor.
Step 7	**remote-span** **Example:** `Device(config-vlan)#` **`remote-span`**	Specifies that the VLAN you specified in Step 5 is part of the RSPAN VLAN.
Step 8	**exit** **Example:** `Device(config-vlan)#` **`exit`**	Returns to global configuration mode.

Step 9	**monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
	Example:
	`Device(config)#` **`monitor session 2 filter ip access-group 7`**
		Specifies the RSPAN session, the types of packets to filter, and the ACLs to use in an FRSPAN session.
		• For *session_number*, specify the session number entered in Step 4.
		• For *access-list-number*, specify the ACL number that you want to use to filter traffic.
		• For *name*, specify the ACL name that you want to use to filter traffic.

Step 10	**end**
	Example:
	`Device(config)#` **`end`**
		Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 11	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 12	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitoring SPAN and RSPAN Operations

The following table describes the command used to display SPAN and RSPAN operations configuration and results to monitor operations:

Table 5: Monitoring SPAN and RSPAN Operations

Command	Purpose
show monitor	Displays the current SPAN configuration.

Configuration Examples for SPAN and RSPAN

The following sections provide configuration examples for SPAN and RSPAN

Example: Configuring Local SPAN

This example shows how to set up SPAN session 1 for monitoring source port traffic to a destination port. First, any existing SPAN configuration for session 1 is deleted, and then bidirectional traffic is mirrored from source Gigabit Ethernet port 1 to destination Gigabit Ethernet port 2, retaining the encapsulation method.

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/1
Device(config)# monitor session 1 destination interface gigabitethernet1/2
encapsulation replicate ingress vlan 7
Device(config)# end
```

This example shows how to remove port 1 as a SPAN source for SPAN session 1:

```
Device> enable
Device# configure terminal
```

```
Device(config)# no monitor session 1 source interface gigabitethernet1/1
Device(config)# end
```

This example shows how to disable received traffic monitoring on port 1, which was configured for bidirectional monitoring:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/1 rx
```

The monitoring of traffic received on port 1 is disabled, but traffic sent from this port continues to be monitored.

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on all ports belonging to VLANs 1 through 3, and send it to destination Gigabit Ethernet port 2. The configuration is then modified to also monitor all traffic on all ports belonging to VLAN 10.

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source vlan 1 - 3 rx

Device(config)# monitor session 2 destination interface gigabitethernet1/2
Device(config)# monitor session 2 source vlan 10
Device(config)# end
```

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on Gigabit Ethernet source port 1, and send it to destination Gigabit Ethernet port 2 with the same egress encapsulation type as the source port, and to enable ingress forwarding with VLAN 6 as the default ingress VLAN:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source gigabitethernet0/1 rx
Device(config)# monitor session 2 destination interface gigabitethernet0/2 encapsulation
replicate ingress vlan 6
Device(config)# end
```

This example shows how to remove any existing configuration on SPAN session 2, configure SPAN session 2 to monitor traffic received on Gigabit Ethernet trunk port 2, and send traffic for only VLANs 1 through 5 and VLAN 9 to destination Gigabit Ethernet port 1:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination interface gigabitethernet1/1 ingress vlan 10
Device(config)# end
```

Examples: Creating an RSPAN VLAN

This example shows how to create the RSPAN VLAN 901:

```
Device> enable
Device# configure terminal
```

```
Device(config)# vlan 901
Device(config-vlan)# remote span
Device(config-vlan)# end
```

This example shows how to remove any existing RSPAN configuration for session 1, configure RSPAN session 1 to monitor multiple source interfaces, and configure the destination as RSPAN VLAN 901:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/1 tx
Device(config)# monitor session 1 source interface gigabitethernet1/2 rx
Device(config)# monitor session 1 source interface port-channel 2
Device(config)# monitor session 1 destination remote vlan 901
Device(config)# end
```

This example shows how to remove any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor traffic received on trunk port 2, and send traffic for only VLANs 1 through 5 and 9 to destination RSPAN VLAN 902:

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination remote vlan 902
Device(config)# end
```

This example shows how to configure VLAN 901 as the source remote VLAN and port 1 as the destination interface:

```
Device> enable
Device# configure terminal
Device(config)# monitor session 1 source remote vlan 901
Device(config)# monitor session 1 destination interface gigabitethernet2/1 ingress vlan 10
Device(config)# end
```

This example shows how to configure VLAN 901 as the source remote VLAN in RSPAN session 2, to configure Gigabit Ethernet source port 2 as the destination interface, and to enable forwarding of incoming traffic on the interface with VLAN 6 as the default receiving VLAN:

```
Device> enable
Device# configure terminal
Device(config)# monitor session 2 source remote vlan 901
Device(config)# monitor session 2 destination interface gigabitethernet1/2 ingress vlan 6
Device(config)# end
```

CHAPTER 5

Configuring Industrial Asset Discovery

- Information About Industrial Asset Discovery, on page 73
- Guidelines and Limitations, on page 74
- Default Configuration, on page 74
- Configuring Industrial Asset Discovery, on page 75
- Verifying Industrial Asset Discovery Information, on page 75
- Feature History for Industrial Asset Discovery, on page 77

Information About Industrial Asset Discovery

The Industrial Asset Discovery (IAD) feature enables users to view details of directly connected end devices. IAD uses discovery messages included in industrial protocols such as Common Industrial Protocol (CIP) and Profinet to discover these details. Because the IE switches are using the same protocol messages as CIP and Profinet devices, there will be no impact to the end devices. End devices will respond normally.

Industrial networks include end devices such as programmable logic controllers (PLCs) and Intelligent Electronic Devices (IEDs) that are used for control process automation. These devices are connected to Supervisory Control and Data Acquisition (SCADA) applications that run protocols such as CIP and Profinet to monitor, control, and manage the end devices. Centralized CIP/Profinet controllers collect device information through broadcast discovery and maintain a device inventory database. However, this information does not include end device layer 2 network connectivity information such as switch, interface, location, and VLAN, which is useful for operators to physically locate and keep track of end devices.

IAD discovery allows detailed location and layer 2 connectivity information to be collected from end devices. This device information is processed and maintained in a local database on the switch. Information collected through CIP/Profinet discovery can be combined with IP device tracking information to provide detailed information about end devices.

Industrial Asset Discovery Operation

When IAD is enabled, after the IE switch boots up, IAD waits for a pre-defined period of time and then sends discovery messages to the industrial protocols that are enabled. Subsequently, the notification is sent at periodic intervals. You can configure this interval using the **iad refresh-interval** command. You can enable and disable discovery for any or all of the protocols in IAD: CIP, Profinet, IP Device Tracking (IPDT), Cisco Discovery Protocol (CDP), and Link Layer Discovery Protocol (LLDP).

The database is automatically refreshed if an interface goes down and comes back up. IAD sends a notification and waits for a pre-defined interval of time before the sending the next discovery message when a link flap event occurs. This helps to avoid sending too many discovery messages.

Device information received through CIP, Profinet, and IPDT or CDP and LLDP is collated and stored in a local database. Each access switch in the network maintains its own IAD database. The local database is dynamically refreshed based on configurable timer values. Information collected about end devices as part of IAD discovery includes:

- Interface Status
- IP-Address
- Mac Address
- Serial Number
- Device PID
- Vendor
- Device Type
- Software version
- Protocol
- Timestamp

The resulting output varies, depending on the network for which an end device has been configured.

Guidelines and Limitations

- IAD is supported on IE3200, IE3300, and IE3400/IE3400H platforms only.
- For CIP/Profinet, the assumption is that end devices are connected through access ports and switch to switch peer links are connected through trunk ports. End devices discovered on trunk interfaces are not added to the local database.
- When an interface goes down, all records pertaining to that interface are deleted. When the interface comes up again, discovery messages are initiated and records are collected.
- For CIP and Profinet discovery messages, an IP address must be assigned to Switched Virtual Interface (SVI) VLAN interface. Same vlan used for Profinet or CIP devices.
- A maximum of 100 records can be stored in the IAD database. There is no restriction on the number of records received on an interface.
- SNMP and YANG are not supported.

Default Configuration

IAD is disabled by default.

When IAD is enabled, these are the default settings:

- Discovery messages are sent for CIP, and Profinet. IPDT is also enabled. The protocol subsystems then send corresponding discovery messages and collect records.
- Records are received from CDP and LLDP.
- The default refresh interval for sending protocol notifications to update the local database is 6 hours.

Configuring Industrial Asset Discovery

Before you begin

Ensure that the protocols you want to enable for IAD are enabled at the switch level.

Procedure

	Command or Action	Purpose
Step 1	`configure terminal`	Enter global configuration mode.
Step 2	`iad enable [cdp | cip | ipdt | lldp | profinet]`	Enables IAD for the specified protocol. If no protocol is specified, IAD is enabled for all protocols.
Step 3	`iad refresh-interval` *interval*	Specify the rate at which CIP/Profinet discovery packets are sent (in seconds). The range is 60-86400. The default is 21,600 (6 hours).

Example

The following example show how to configure IAD for CIP and Profinet and sets the refresh interval to 3 hours:

```
IE3400_IAD#config t
IE3400_IAD(config)#iad enable cip
IE3400_IAD(config)#iad enable profinet
IE3400_IAD(config)#iad refresh-interval 10800
```

Verifying Industrial Asset Discovery Information

Use the following commands to display IAD inventory and configuration status.

Command	Description
show iad inventory [all | interface | protocol]	Display the IAD device inventory: • all—All device records in tabular format • interface—Filter records by interface • protocol—Filter records by protocol: • cdp • cip • ipdt • lldp • profinet
show iad status	Display present IAD configuration status

The following example shows you how to verify the IAD status:

```
IE3400#show iad status
IAD Information:
Status : Enabled
Send/Receive Notification to CDP : Enabled
Send/Receive Notification to CIP : Enabled
Send/Receive Notification to IPDT : Enabled
Send/Receive Notification to LLDP : Enabled
Send/Receive Notification to PROFINET : Enabled
Last discovery sent for CIP/Profinet : 14:41:47 UTC Wed Nov 15 2023
IAD Records Refresh Interval Rate : 10 secs
```

The following example shows you how to verify the IAD inventory:

```
IE3400#show iad inventory all
Capability codes:
    (R) Router, (B) Source Route Bridge, (T) Telephone, (H) Host
    (C) DOCSIS Cable Device (W) WLAN Access Point (P) Repeater
    (G) Trans Bridge, (F) Switch, (I) IGMP, (E) Phone, (S) Station
    (D) Remote, (A) CVTA, (M) Two-port Mac Relay, (O) Other
Interface  Status      IP-Address        Mac Addr        Serial No    Device PID      Vendor
         Device Type  SW ver       Protocol      Last Reported Time
---------- ------ ---------------   -----------------   -----------   ---------------
---------- ------------ ------------ ------------- --------------------
Gi2/5      UP     10.76.29.205      0C:75:BD:C8:68:29   Unknown      IE-3400-8P2S    Unknown
    B,R          17.14.202310 CDP,LLDP      16:39:56 UTC Tue Nov 21 2023
Gi2/2      UP     Unknown           AC:64:17:65:D4:A9   Unknown      ET200MP         SIEMENS
 A  IO           Unknown      PROFINET      16:39:49 UTC Tue Nov 21 2023
Gi1/7      UP     29.29.29.60       38:4B:24:6A:A6:48   Unknown      SCALANCE XC-200 SIEMENS
 A  IO           Unknown      PROFINET      16:39:49 UTC Tue Nov 21 2023
Gi1/6      UP     192.168.1.8       00:00:BC:D1:2E:DB   8467751      1756-EN2T/C     Rockwell
   EtherNet/IP  Unknown      CIP           16:39:48 UTC Tue Nov 21 2023
Gi2/8      UP     192.168.1.30      A4:53:0E:91:E9:61   1049749832   IE-3400H-24T-E  Cisco
 Sys  Switch       Unknown      CIP           16:39:48 UTC Tue Nov 21 2023
Gi2/4      UP     192.168.1.10      00:29:C2:3C:09:8B   943458688    IE-3200-8T2S    Cisco
 Sys  Switch       Unknown      CIP           16:39:48 UTC Tue Nov 21 2023
Gi2/3      UP     Unknown           D0:EC:35:58:53:04   Unknown      IE-3400-8T2S    Unknown
```

```
    R,F,I        17.13.202310 CDP          16:39:53 UTC Tue Nov 21 2023

Total entries displayed : 7
```

Feature History for Industrial Asset Discovery

Feature Name	Release	Description
Industrial Asset Discovery (IAD)	Cisco IOS XE 17.14.1	Initial release on IE3200, IE3300, and IE3400/IE3400H series switches