

Configuring IP Services

This module describes how to configure optional IP services. For a complete description of the IP services commands in this chapter, refer to the *Cisco IOS IP Application Services Command Reference*. To locate documentation of other commands that appear in this module, use the master command list, or search online.

- Information About IP Services, on page 1
- How to Configure IP Services, on page 5
- Configuration Examples for IP Services, on page 14
- Additional References For IP Services, on page 17
- Feature Information for IP Services, on page 18

Information About IP Services

IP Source Routing

The Cisco IOS XE software examines IP header options on every packet. It supports the IP header options Strict Source Route, Loose Source Route, Record Route, and Time Stamp, which are defined in RFC 791. If the software finds a packet with one of these options enabled, it performs the appropriate action. If it finds a packet with an invalid option, it sends an Internet Control Message Protocol (ICMP) parameter problem message to the source of the packet and discards the packet.

IP provides a provision known as source routing that allows the source IP host to specify a route through the IP network. Source routing is specified as an option in the IP header. If source routing is specified, the software forwards the packet according to the specified source route. IP source routing is employed when you want to force a packet to take a certain route through the network. The default is to perform source routing. IP source routing is rarely used for legitimate purposes in networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash devices running these implementations by sending them datagrams with source routing options. Disable IP source routing whenever possible. Disabling IP source routing will cause a Cisco router to never forward an IP packet that carries a source routing option.

Note From Cisco IOS XE Release 17.1.1, IP source routing is disabled by default.

ICMP Overview

Originally created for the TCP/IP suite in RFC 792, the Internet Control Message Protocol (ICMP) was designed to report a small set of error conditions. ICMP can also report a wide variety of error conditions and provide feedback and testing capabilities. Each message uses a common format and is sent and received by using the same protocol rules.

ICMP enables IP to perform addressing, datagram packaging, and routing by allowing encapsulated messages to be sent and received between IP devices. These messages are encapsulated in IP datagrams just like any other IP message. When the message is generated, the original IP header is encapsulated in the ICMP message and these two pieces are encapsulated within a new IP header to be returned as an error report to the sending device.

ICMP messages are sent in several situations: when a datagram cannot reach its destination, when the gateway does not have the buffering capacity to forward a datagram, and when the gateway can direct the host to send traffic on a shorter route. To avoid the infinite regress of messages about messages, no ICMP messages are sent about ICMP messages.

ICMP does not make IP reliable or ensure the delivery of datagrams or the return of a control message. Some datagrams may be dropped without any report of their loss. The higher-level protocols that use IP must implement their own reliability procedures if reliable communication is required.

ICMP Unreachable Error Messages

Type 3 error messages are sent when a message cannot be delivered completely to the application at a destination host. Six codes contained in the ICMP header describe the unreachable condition as follows:

- 0—Network unreachable
- 1—Host unreachable
- 2—Protocol unreachable
- 3—Port unreachable
- 4—Fragmentation needed and the “don’t fragment﷯? (DF) bit is set
- 5—Source route failed

Cisco IOS XE software can suppress the generation of ICMP unreachable destination error messages, which is called rate-limiting. The default is no unreachable messages more often than once every half second. Separate intervals can be configured for code 4 and all other unreachable destination error messages. However, there is no method of displaying how many ICMP messages have not been sent.

The ICMP Unreachable Destination Counters feature provides a method to count and display the unsent Type 3 messages. This feature also provides console logging with error messages when there are periods of excessive rate limiting that would indicate a Denial of Service (DoS) attack against the router.

If the Cisco IOS XE software receives a nonbroadcast packet destined for itself that uses an unknown protocol, it sends an ICMP protocol unreachable message back to the source. Similarly, if the software receives a packet that it is unable to deliver to the final destination because it knows of no route to the destination address, it sends an ICMP host unreachable message to the source. This functionality is enabled by default.

Disable ICMP host unreachable messages whenever possible. ICMP supports IP traffic by relaying information about paths, routes, and network conditions. These messages can be used by an attacker to gain network mapping information.

Because the null interface is a packet sink, packets forwarded there will always be discarded and, unless disabled, will generate host unreachable messages. In that case, if the null interface is being used to block a Denial-of-Service attack, these messages flood the local network with these messages. Disabling these messages prevents this situation. In addition, because all blocked packets are forwarded to the null interface, an attacker receiving host unreachable messages could use those messages to determine Access Control List (ACL) configuration. If the “null 0 interface is configured on your router, disable ICMP host unreachable messages for discarded packets or packets routed to the null interface.

ICMP Mask Reply Messages

Occasionally, network devices must know the subnet mask for a particular subnetwork in the internetwork. To obtain this information, such devices can send ICMP mask request messages. ICMP mask reply messages are sent in reply from devices that have the requested information. The Cisco IOS XE software can respond to ICMP mask request messages if this function is enabled.

These messages can be used by an attacker to gain network mapping information.

ICMP Redirect Messages

Routes are sometimes less than optimal. For example, it is possible for the router to be forced to resend a packet through the same interface on which it was received. If the router resends a packet through the same interface on which it was received, the Cisco IOS XE software sends an ICMP redirect message to the originator of the packet telling the originator that the router is on a subnet directly connected to the receiving device, and that it must forward the packet to another system on the same subnet. The software sends an ICMP redirect message to the originator of the packet because the originating host presumably could have sent that packet to the next hop without involving this device at all. The redirect message instructs the sender to remove the receiving device from the route and substitute a specified device representing a more direct path. This functionality is enabled by default.

In a properly functioning IP network, a router will send redirects only to hosts on its own local subnets, no end node will ever send a redirect, and no redirect will ever be traversed more than one network hop. However, an attacker may violate these rules; some attacks are based on this. Disabling ICMP redirects will cause no operational impact to the network, and it eliminates this possible method of attack.

Denial of Service Attack

Denial of service has become a growing concern, especially when considering the associated costs of such an attack. DoS attacks can decrease the performance of networked devices, disconnect the devices from the network, and cause system crashes. When network services are unavailable, enterprises and service providers suffer the loss of productivity and sales.

The objective of a DoS attack is to deprive a user or organization access to services or resources. If a Website is compromised by a DoS attack, millions of users could be denied access to the site. DoS attacks do not typically result in intrusion or the illegal theft of information. Instead of providing access to unauthorized users, DoS attacks can cause much aggravation and cost to the target customer by preventing authorized access. Distributed DoS (DDoS) attacks amplify DoS attacks in that a multitude of compromised systems coordinate to flood targets with attack packets, thereby causing denial of service for users of the targeted systems.

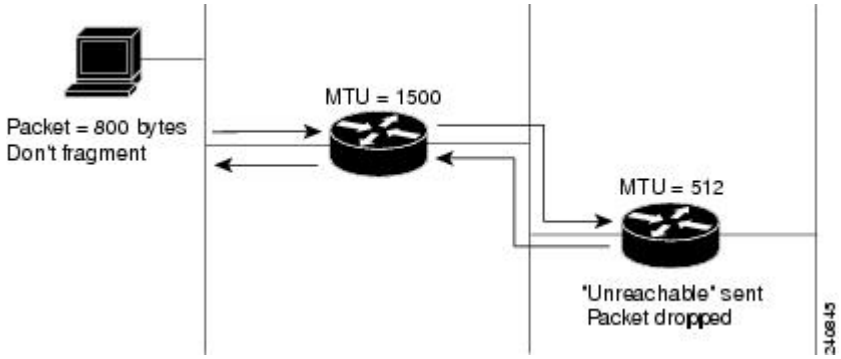
A DoS attack occurs when a stream of ICMP echo requests (pings) are broadcast to a destination subnet. The source addresses of these requests are falsified to be the source address of the target. For each request sent by

the attacker, many hosts on the subnet will respond flooding the target and wasting bandwidth. The most common DoS attack is called a “smurf” attack, named after an executable program and is in the category of network-level attacks against hosts. DoS attacks can be easily detected when error-message logging of the ICMP Unreachable Destination Counters feature is enabled.

Path MTU Discovery

The Cisco IOS XE software supports the IP Path MTU Discovery mechanism, as defined in RFC 1191. IP Path MTU Discovery allows a host to dynamically discover and cope with differences in the maximum allowable maximum transmission unit (MTU) size of the various links along the path. Sometimes a router is unable to forward a datagram because it requires fragmentation (the packet is larger than the MTU you set for the interface with the **ip mtu** interface configuration command), but the “don’t fragment?? (DF) bit is set. The Cisco IOS XE software sends a message to the sending host, alerting it to the problem. The host will need to fragment packets for the destination so that they fit the smallest packet size of all the links along the path. This technique is shown in the figure below.

Figure 1: IP Path MTU Discovery



IP Path MTU Discovery is useful when a link in a network goes down, forcing the use of another, different MTU-sized link (and different routers). As shown in the figure above, suppose a router is sending IP packets over a network where the MTU in the first router is set to 1500 bytes, but the second router is set to 512 bytes. If the “don’t fragment?? bit of the datagram is set, the datagram would be dropped because the 512-byte router is unable to forward it. All packets larger than 512 bytes are dropped in this case. The second router returns an ICMP destination unreachable message to the source of the datagram with its Code field indicating “Fragmentation needed and DF set.?? To support IP Path MTU Discovery, it would also include the MTU of the next hop network link in the low-order bits of an unused header field.

IP Path MTU Discovery is also useful when a connection is being established and the sender has no information at all about the intervening links. It is always advisable to use the largest MTU that the links will bear; the larger the MTU, the fewer packets the host must send.

Note IP Path MTU Discovery is a process initiated by end hosts. If an end host does not support IP Path MTU Discovery, the receiving device will have no mechanism to avoid fragmenting datagrams generated by the end host.

If a router that is configured with a small MTU on an outbound interface receives packets from a host that is configured with a large MTU (for example, receiving packets from a Token Ring interface and forwarding them to an outbound Ethernet interface), the router fragments received packets that are larger than the MTU

of the outbound interface. Fragmenting packets slows the performance of the router. To keep routers in your network from fragmenting received packets, run IP Path MTU Discovery on all hosts and routers in your network, and always configure the largest possible MTU for each router interface type.

Show and Clear Commands for IOS Sockets

The Show and Clear Commands for IOS Sockets feature introduces the **show udp**, **show sockets**, and **clear sockets**commands. These new commands are useful for monitoring and managing the Cisco IOS Socket library.

In Cisco IOS software, sockets are a per process entity. This means that the maximum number of sockets is per process and all sockets are managed on a per process basis. For example, each Cisco IOS process could have a socket with file descriptor number 1. This is unlike UNIX or other operating systems that have per system file descriptor allocations.

The **show**and **clear** commands operate on a per process basis to be consistent with the current functionality. Thus, any action taken by the commands will be applicable only to a particular process at a time as selected by the process ID entered on the CLI.

Many applications have a need for **show** and **clear**commands, which primarily aid in debugging. The following scenarios provide examples of when these commands might be useful:

- The application H.323 is using sockets for voice calls. According to the current number of calls, there is still space for more sockets. However, no more sockets can be opened. You can now use the the **show sockets** command to find out if the socket space is indeed exhausted or if there are unused sockets available.
- An application is waiting for a particular socket event to happen. A UDP segment was seen, but the application never became active. You can use the **show udp** command to display the list of events being monitored to determine if a UDP socket event is being monitored or if the socket library failed to activate the application.
- An application wants to forcibly close all the sockets for a particular process. You can use the **clear sockets** command to close both the sockets and the underlying TCP or UDP connection or Stream Control Transmission Protocol (SCTP) association.

How to Configure IP Services

Protecting Your Network from DOS Attacks

ICMP supports IP traffic by relaying information about paths, routes, and network conditions. ICMP messages can be used by an attacker to gain network mapping information. IP source routing allows the source IP host to specify a route through the IP network and is rarely used for legitimate purposes in networks. Some older IP implementations do not process source-routed packets properly, and it may be possible to crash devices running these implementations by sending them datagrams with source routing options.

Whenever possible, ICMP messages and IP source routing should be disabled.

Note From Cisco IOS XE Release 17.1.1, IP source routing is disabled by default.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip source-route**
4. **interface** *type/number/slot*
5. **no ip unreachables**
6. **no ip redirects**
7. **no ip mask-reply**

DETAILED STEPS

Procedure

	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 ip source-route** **Example:** `Device(config)# no ip source-route`	Disables IP source routing. **Note** From Cisco IOS XE Release 17.1.1, IP sour
Step 4	**interface** *type/number/slot* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to configure and enters interface configuration mode.
Step 5	**no ip unreachables** **Example:** `Device(config-if)# no ip unreachables`	Disables the sending of ICMP protocol unreachable and host unreachable messages. This command is enabled by default.

	Command or Action	Purpose
		 **Note** Disabling the unreachable messages also having the Cisco IOS XE software send
Step 6	**no ip redirects** **Example:** `Device(config-if)# no ip redirects`	Disables the sending of ICMP redirect messages to learn routes. This command is enabled by default.
Step 7	**no ip mask-reply** **Example:** `Device(config-if)# no ip mask-reply`	Disables the sending of ICMP mask reply messages.

Configuring ICMP Unreachable Rate Limiting User Feedback

Perform this task to clear all of the unreachable destination packet statistics and to specify an interval number for unreachable destination messages. This task also configures a packet counter (threshold) and interval to trigger a logging message to a console. This task is beneficial to begin a new log after the thresholds have been set.

SUMMARY STEPS

1. **enable**
2. **clear ip icmp rate-limit** [*interface-type interface-number*]
3. **configure terminal**
4. **ip icmp rate-limit unreachable** [**df**] [*ms*] [**log** [*packets*] [*interval-ms*]]
5. **exit**
6. **show ip icmp rate-limit** [*interface-type interface-number*]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ip icmp rate-limit** [*interface-type interface-number*] **Example:**	Clears all current ICMP unreachable statistics for all configured interfaces. The optional *interface-type* and *interface-number* arguments clear the statistics for only one interface.

	Command or Action	Purpose
	`Router# clear ip icmp rate-limit ethernet 2/3`	
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 4	**ip icmp rate-limit unreachable** [**df**] [*ms*] [**log** [*packets*] [*interval-ms*]]
	Example:
	`Router(config)# ip icmp rate-limit unreachable df log 1100 12000`
		Specifies the rate limitation of ICMP unreachable destination messages and the error message log threshold for generating a message. The default is no unreachable messages are sent more often than once every half second.
		The arguments and keywords are as follows:
		• **df** --(Optional) When "don't fragment" (DF) bit is set in the ICMP header, a datagram cannot be fragmented. If the **df** keyword is not specified, all other types of destination unreachable messages are sent.
		• *ms* --(Optional) Interval at which unreachable messages are generated. The valid range is from 1 to 4294967295.
		• **log** --(Optional) List of error messages. The arguments are as follows:
		  • *packets*--(Optional) Number of packets that determine a threshold for generating a log. The default is 1000.
		  • *interval-ms*--(Optional) Time limit for an interval for which a logging message is triggered. The default is 60000, which is 1 minute.
		Note Counting begins as soon as this command is

Step 5	**exit** **Example:** `Router# exit`	Exits to privileged EXEC mode.
Step 6	**show ip icmp rate-limit** [*interface-type interface-number*] **Example:** `Router# show ip icmp rate-limit ethernet 2/3`	(Optional) Displays all current ICMP unreachable statistics for all configured interfaces. The optional *interface-type* and *interface-number* arguments display the statistics for only one interface.

Example

The following output using the**show ip icmp rate-limit** command displays the unreachable destinations by interface:

```
Router# show ip icmp rate-limit
                          DF bit unreachables       All other unreachables
Interval (millisecond)    500                       500
Interface                 # DF bit unreachables     # All other unreachables
---------                 ---------------------     ------------------------
Ethernet0/0               0                         0
Ethernet0/2               0                         0
Serial3/0/3               0                         19
The greatest number of unreachables is on serial interface 3/0/3.
```

Setting the MTU Packet Size

All interfaces have a default MTU packet size. You can adjust the IP MTU size so that the Cisco IOS XE software will fragment any IP packet that exceeds the MTU set for an interface.

Changing the MTU value (with the **mtu** interface configuration command) can affect the IP MTU value. If the current IP MTU value is the same as the MTU value and you change the MTU value, the IP MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the **mtu** interface configuration command.

All devices on a physical medium must have the same protocol MTU in order to operate.

Perform this task to set the MTU packet size for a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/number/slot*
4. **ip mtu** *bytes*
5. **end**

DETAILED STEPS

Procedure

	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	**interface** *type/number/slot* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 300`	Sets the IP MTU packet size for an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring IP Accounting With NetFlow

IP Accounting collects the number of bytes and packets processed by the network element based on the source or destination IP address, or the configured IP precedence. The information collected can be used to identify users for network usage billing, monitoring, and troubleshooting.

Cisco ASR 1000 Series Aggregation Services Routers do not support the IP Accounting feature; however, support Flexible Netflow as the recommended method to collect network information. For more information on Flexible NetFlow configuration see the Flexible NetFlow Configuration Guide.

The following steps are performed in this task:

1. Create a flow record based on the IP address and define the counters to be collected.
2. Create a flow record based on IP precedence and define the counters to be collected.
3. Create a flow monitor, define the monitor parameters, and link it with the IP address-based flow record.
4. Create a flow monitor, define the monitor parameters, and link it with IP precedence-based flow record.
5. Attach the IP address-based flow monitor and IP precedence-based flow monitor to an interface where the traffic is monitored.
6. Monitor the flow cache and statistics.
7. Clean the flow cache and statistics.
8. Export the flow cache to external source in .csv format.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*

4. **match ipv4 source address**
5. **match ipv4 destination address**
6. **collect counter packets long**
7. **exit**
8. **flow record** *record-name*
9. **match ipv4precedence**
10. **collect counter packets long**
11. **exit**
12. **flow monitor** *flow-monitor-name*
13. **record** *record-name*
14. **cache timeout active** *seconds*
15. **cache entries** *number*
16. **exit**
17. **flow monitor** *flow-monitor-name*
18. **record** *record-name*
19. **cache timeout active** *seconds*
20. **cache entries** *number*
21. **exit**
22. **interface** *type number*
23. **ip flow monitor** *monitor-name* **input**
24. **ip flow monitor** *monitor-name* **input**
25. **exit**
26. **show flow monitor** *monitor-name* **cache**
27. **show flow monitor** *monitor-name* **cache**
28. **clear flow monitor** *monitor-name* **cache**
29. **clear flow monitor** *monitor-name* **statistics**
30. **show flow monitor** *monitor-name* **cache format csv**

DETAILED STEPS

Procedure

	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 ip-acct`	Creates or modifies an existing Flexible NetFlow flow record, and enters Flexible NetFlow flow record configuration mode.

	Command or Action	Purpose
Step 4	**match ipv4 source address** **Example:** `Device(config-flow-record)# match ipv4 source address`	Configures the IPv4 source address as a key field for a flow record.
Step 5	**match ipv4 destination address** **Example:** `Device(config-flow-record)# match ipv4 destination address`	Configures the IPv4 destination address as a key field for a flow record.
Step 6	**collect counter packets long** **Example:** `Device(config-flow-record)# collect counter packets long`	Configures a 64-bit counter that is incremented for each packet seen in the flow.
Step 7	**exit** **Example:** `Device(config-flow-record)# exit`	Exits Flexible NetFlow flow record configuration mode and returns to global configuration mode.
Step 8	**flow record** *record-name* **Example:** `Device(config)# flow record prec-acct`	Creates or modifies an existing Flexible NetFlow flow record, and enters Flexible NetFlow flow record configuration mode.
Step 9	**match ipv4precedence** **Example:** `Device(config-flow-record) match ipv4 precedence`	Configures the IPv4 precedence (part of type of service) as a key field.
Step 10	**collect counter packets long** **Example:** `Device(config-flow-record)# collect counter packets long`	Configures a 64-bit counter that is incremented for each packet seen in the flow.
Step 11	**exit** **Example:** `Device(config-flow-record)# exit`	Exits Flexible NetFlow flow record configuration mode and returns to global configuration mode.
Step 12	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor ip-acct`	Creates or modifies an existing Flexible NetFlow flow monitor and enters Flexible NetFlow flow monitor configuration mode.
Step 13	**record** *record-name* **Example:** `Device(config-flow-monitor)# record ip-acct`	Configures a user-defined flow record that was previously configured for a Flexible NetFlow flow monitor.

	Command or Action	Purpose
Step 14	**cache timeout active** *seconds* **Example:** `Device(config-flow-monitor)# cache timeout active 604800`	Specifies the active flow timeout, in seconds for the flow monitor. **Note** Cisco IOS XE Releases do not support p this command.
Step 15	**cache entries** *number* **Example:** `Device(config-flow-monitor)# cache entries 200000`	Specifies the maximum number of entries in the flow monitor cache.
Step 16	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 17	**flow monitor** *flow-monitor-name* **Example:** `Device(config)# flow monitor prec-acct`	Create or modifies an existing Flexible NetFlow flow monitor, and enters Flexible NetFlow flow monitor configuration mode.
Step 18	**record** *record-name* **Example:** `Device(config-flow-monitor)# record prec-acct`	Configures a user-defined flow record that was previously configured for a Flexible NetFlow flow monitor.
Step 19	**cache timeout active** *seconds* **Example:** `Device(config-flow-monitor)# cache timeout active 604800`	Specifies the active flow timeout, in seconds for the flow monitor. **Note** Cisco IOS XE Releases do not support p this command.
Step 20	**cache entries** *number* **Example:** `Device(config-flow-monitor)# cache entries 200000`	Specifies the maximum number of entries in the flow monitor cache.
Step 21	**exit** **Example:** `Device(config-flow-monitor)# exit`	Exits Flexible NetFlow flow monitor configuration mode and returns to global configuration mode.
Step 22	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/4`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 23	**ip flow monitor** *monitor-name* **input** **Example:** `Device(config-if)# ip flow monitor ip-acct input`	Enables a Flexible NetFlow flow monitor for IPv4 traffic that the router is transmitting.
Step 24	**ip flow monitor** *monitor-name* **input** **Example:** `Device(config-if)# ip flow monitor prec-acct input`	Enables a Flexible NetFlow flow monitor for IPv4 traffic that the router is transmitting.
Step 25	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 26	**show flow monitor** *monitor-name* **cache** **Example:** `Device# show flow monitor prec-acct cache`	Displays the contents of the cache for the flow monitor record that was previously configured.
Step 27	**show flow monitor** *monitor-name* **cache** **Example:** `Device# show flow monitor ip-acct cache`	Displays the contents of the cache for the flow monitor record that was previously configured.
Step 28	**clear flow monitor** *monitor-name* **cache** **Example:** `Device# clear flow monitor ip-acct cache`	Clears the flow monitor cache information.
Step 29	**clear flow monitor** *monitor-name* **statistics** **Example:** `Device# clear flow monitor ip-acct statistics`	Clears the flow monitor statistics.
Step 30	**show flow monitor** *monitor-name* **cache format csv** **Example:** `Device# show flow monitor ip-acct cache format csv | append bootflash:ip-acct`	Exports the flow monitor cache contents to an external source in comma separated variables (CSV) format.

Configuration Examples for IP Services

Example: Protecting Your Network from DOS Attacks

The following example shows how to change some of the ICMP defaults for Gigabit Ethernet interface 0/0/0 to prevent ICMP from relaying information about paths, routes, and network conditions, which can be used by an attacker to gain network mapping information.

Disabling the unreachable messages will have a secondary effect: it will also disable IP Path MTU Discovery, because path discovery works by having the Cisco IOS XE software send Unreachable messages. If you have

a network segment with a small number of devices and an absolutely reliable traffic pattern—which could easily happen on a segment with a small number of rarely used user devices—you would be disabling options that your device would be unlikely to use anyway.

```
Device(config)# no ip source-route
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# no ip unreachables
Device(config-if)# no ip redirects
Device(config-if)# no ip mask-reply
```

Example: Configuring ICMP Unreachable Destination Counters

The following example shows how to clear all of the unreachable destination packet statistics and to specify an interval number for unreachable destination messages. This example also shows how to configure a packet counter threshold and interval to trigger a logging message to a console.

```
Router# clear ip icmp rate-limit ethernet 0/0
Router# configure terminal
Router(config)# ip icmp rate-limit unreachable df log 1100 12000
```

Example: Setting the MTU Packet Size

The following example shows how to change the default MTU packet size for Gigabit Ethernet interface 0/0/0:

```
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip mtu 300
```

Example: Configuring IP Accounting with NetFlow

The following example shows how to use NetFlow for IP Accounting:

```
! Created flow record and flow monitor for IP address accounting
Device# configure terminal
Device(config)# flow record ip-acct
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# exit
Device(config)# flow monitor ip-acct
Device(config-flow-monitor)# record ip-acct
Device(config-flow-monitor)# cache timeout active 604800
Device(config-flow-monitor)# cache entries 200000
Device(config-flow-monitor)# exit

! Created flow record and flow monitor for precedence accounting
Device(config)# flow record prec-acct
Device(config-flow-record)# match ipv4 precedence
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# exit
Device(config)# flow monitor prec-acct
Device(config-flow-monitor)# record prec-acct
Device(config-flow-monitor)# cache timeout active 604800
Device(config-flow-monitor)# cache entries 200000
```

```
Device(config-flow-monitor)# exit

! Apply both ip-acct and prec-acct on an interface
Device(config)# interface GigabitEthernet 0/0/4
Device(config-if)# ip address 192.0.2.1 255.255.255.0
Device(config-if)# ip flow monitor ip-acct input
Device(config-if)# ip flow monitor prec-acct input
Device(config-if)# negotiation auto
Device(config-if)# end
```

Verifying IP Accounting with NetFlow

SUMMARY STEPS

1. **show flow monitor** *monitor-name***cache**
2. **show flow monitor** *monitor-name* **cache**
3. **clear flow monitor** *monitor-name* {**cache** | **force-export** | **statistics**}
4. **show flow monitor** *monitor-name* **format csv | append bootflash:***monitor-name*}

DETAILED STEPS

Procedure

Step 1 **show flow monitor** *monitor-name***cache**

Displays the contents of the cache for the flow monitor.

Example:

```
Device# show flow monitor prec-acct cache

  Cache type:                               Normal (Platform cache)
  Cache size:                               200000
  Current entries:                               3

  Flows added:                                   3
  Flows aged:                                    0

IP PREC             pkts long
=======  ====================
      0               8117679
      1               8118233
      2               8118761
```

Step 2 **show flow monitor** *monitor-name* **cache**

Displays the contents of the cache for the flow monitor.

Example:

```
Device# show flow monitor ip-acct cache

  Cache type:                               Normal (Platform cache)
```

```
  Cache size:                             200000
  Current entries:                            10

  Flows added:                                10
  Flows aged:                                  0

IPV4 SRC ADDR    IPV4 DST ADDR               pkts long
===============  ===============  ====================
192.168.0.1      192.168.2.2                5987314
192.168.0.1      192.168.3.2                5987314
192.168.0.1      192.168.10.2               5987354
192.168.0.1      192.168.1.2                5987363
192.168.0.1      192.168.8.2                5987384
192.168.0.1      192.168.7.2                5987387
192.168.0.1      192.168.6.2                5987420
192.168.0.1      192.168.9.2                5987606
192.168.0.1      192.168.5.2                5987645
192.168.0.1      192.168.2.2                5987659
```

Step 3 **clear flow monitor** *monitor-name* {**cache** | **force-export** | **statistics**}

Clears the flow monitor cache information.

Example:

```
Device# clear flow monitor ip-acct cache
```

Step 4 **show flow monitor** *monitor-name* **format csv | append bootflash:***monitor-name*}

Displays output of statistics from the flows in a flow monitor cache in comma-separated variables (CSV) format.

Example:

```
Device# show flow monitor ip-acct cache format csv | append bootflash:ip-acct
```

Additional References For IP Services

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IP application services commands	Cisco IOS IP Application Services Command Reference

Standards and RFCs

Standard	Title
RFC 1256	ICMP Router Discovery Messages

Technical Assistance

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

Feature Information for IP Services

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

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