

Documentation

[HOME](#)[CONTENTS](#)[PREVIOUS](#)[NEXT](#)[GLOSSARY](#)[FEEDBACK](#)[SEARCH](#)[HELP](#)

Table of Contents



[TCP Intercept](#)

[Description](#)

[Platforms](#)

[Prerequisites](#)

[Configuration Tasks](#)

[Enable TCP Intercept](#)[Set the TCP Intercept Mode](#)[Set the TCP Intercept Drop Mode](#)[Change the TCP Intercept Timers](#)[Change the TCP Intercept Aggressive Thresholds](#)[Monitor and Maintain TCP Intercept](#)

[Configuration Example](#)

[Command Reference](#)

[ip tcp intercept connection-timeout](#)[ip tcp intercept drop-mode](#)[ip tcp intercept finrst-timeout](#)[ip tcp intercept list](#)[ip tcp intercept max-incomplete high](#)[ip tcp intercept max-incomplete low](#)[ip tcp intercept mode](#)[ip tcp intercept one-minute high](#)[ip tcp intercept one-minute low](#)[ip tcp intercept watch-timeout](#)[show tcp intercept connections](#)[show tcp intercept statistics](#)

[Debug Command](#)

[debug ip tcp intercept](#)

TCP Intercept

Description

The TCP intercept feature implements software to protect TCP servers from TCP SYN-flooding attacks, which are a type of denial-of-service attack.

A SYN-flooding attack occurs when a hacker floods a server with a barrage of requests for connection. Since these messages have unreachable return addresses, the connections cannot be established. The resulting volume of unresolved open connections eventually overwhelms the server and can cause it to deny service to valid requests, thereby preventing legitimate users from connecting to a Web site, accessing e-mail, using FTP service, and so on.

The TCP intercept feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests. In intercept mode, the TCP intercept software intercepts TCP synchronization (SYN) packets from clients to servers that match an extended access list. The software establishes a connection with the client on behalf of the destination server, and if successful, establishes the connection with the server on behalf of the client and knits the two half-connections together transparently. Thus, connection attempts from unreachable hosts will never reach the server. The software continues to intercept and forward packets throughout the duration of the connection.

In the case of illegitimate requests, the software's aggressive timeouts on half-open connections and its thresholds on TCP connection requests protect destination servers while still allowing valid requests.

When establishing your security policy using TCP intercept, you can choose to intercept all requests or only those coming from specific networks or destined for specific servers. You can also configure the connection rate and threshold of outstanding connections.

You can choose to operate TCP intercept in watch mode, as opposed to intercept mode. In watch mode, the software passively watches the connection requests flowing through the router. If a connection fails to get established in a configurable interval, the software intervenes and terminates the connection attempt.

TCP options that are negotiated on handshake (such as RFC 1323 on window scaling, for example) will not be negotiated because the TCP intercept software does not know what the server can do or will negotiate.

Platforms

This feature is supported on these platforms:

- Cisco 4000-M, Cisco 4500-M, Cisco 4700
- Cisco 7200 series
- Cisco 7500 series

Prerequisites

Some type of IP routing must be configured so that the router is operating. This feature works on TCP flows.

Configuration Tasks

Perform the following tasks to configure the TCP intercept. The first task is required; the rest are optional.

- [Enable TCP Intercept](#)
- [Set the TCP Intercept Mode](#)

- [Set the TCP Intercept Drop Mode](#)
- [Change the TCP Intercept Timers](#)
- [Change the TCP Intercept Aggressive Thresholds](#)
- [Monitor and Maintain TCP Intercept](#)

Enable TCP Intercept

To enable TCP intercept, perform the following tasks in global configuration mode:

Task	Command
Step 1 Define an IP extended access list.	access-list <i>access-list-number</i> { deny permit } tcp any <i>destination destination-wildcard</i>
Step 2 Enable TCP intercept.	ip tcp intercept list <i>extended-access-list-number</i>

You can define an access list to intercept all requests or only those coming from specific networks or destined for specific servers. Typically the access list will define the source as **any** and define specific destination networks or servers. That is, you do not attempt to filter on the source addresses because you don't necessarily know who to intercept packets from. You identify the destination in order to protect destination servers.

If no access list match is found, the router allows the request to pass with no further action.

Set the TCP Intercept Mode

The TCP intercept can operate in either active intercept mode or passive watch mode. The default is intercept mode.

In intercept mode, the software actively intercepts each incoming connection request (SYN) and responds on behalf of the server with an ACK and SYN, then waits for an ACK of the SYN from the client. When that ACK is received, the original SYN is set to the server and the software performs a three-way handshake with the server. When this is complete, the two half-connections are joined.

In watch mode, connection requests are allowed to pass through the router to the server, but are watched until they become established. If they fail to become established by 30 seconds (configurable with the **ip tcp intercept watch-timeout** command), the software sends a Reset to the server to clear up its state.

To set the TCP intercept mode, perform the following task in global configuration mode:

Task	Command
Set the TCP intercept mode.	ip tcp intercept mode { intercept watch }

Set the TCP Intercept Drop Mode

When under attack, the TCP intercept feature becomes more aggressive in its protective behavior. If the number of incomplete connections exceeds 1100 or the number of connections arriving in the last one minute exceeds 1100, each new arriving connection causes the oldest partial connection to be deleted. Also, the initial retransmission timeout is reduced by half to 0.5 seconds (so the total time trying to establish a connection is cut in half).

By default, the software drops the oldest partial connection. Alternatively, you can configure the software to drop a random connection. To set the drop mode, perform the following task in global configuration mode:

Task	Command
Set the drop mode.	ip tcp intercept drop-mode { oldest random }

Change the TCP Intercept Timers

By default, the software waits for 30 seconds for a watched connection to reach established state before sending a Reset to the server. To change this value, perform the following task in global configuration mode:

Task	Command
Change the time allowed to reach established state.	ip tcp intercept watch-timeout <i>seconds</i>

By default, the software waits for 5 seconds from receipt of a reset or FIN-exchange before it ceases to manage the connection. To change this value, perform the following task in global configuration mode:

Task	Command
Change the time between receipt of a reset or FIN-exchange and dropping the connection.	ip tcp intercept finrst-timeout <i>seconds</i>

By default, the software still manages a connection for 24 hours after no activity. To change this value, perform the following task in global configuration mode:

Task	Command
Change the time the software will manage a connection after no activity.	ip tcp intercept connection-timeout <i>seconds</i>

Change the TCP Intercept Aggressive Thresholds

Two factors determine when aggressive behavior begins and ends: total incomplete connections and connection requests during the last one-minute sample period. Both thresholds have default values that can be redefined.

When a threshold is exceeded, the TCP intercept assumes the server is under attack and goes into aggressive mode. When in aggressive mode, the following occurs:

- Each new arriving connection causes the oldest partial connection to be deleted. (You can change to a random drop mode.)
- The initial retransmission timeout is reduced by half to 0.5 seconds (and so the total time trying to establish the connection is cut in half). (When not in aggressive mode, the code does exponential back-off on its retransmissions of SYN segments. The initial retransmission timeout is 1 second. The subsequent timeouts are 2 seconds, 4 seconds, 8 seconds, and 16 seconds. The code retransmits 4 times before giving up, so it gives up after 31 seconds of no acknowledgment.)
- If in watch mode, the watch timeout is reduced by half. (If the default is in place, the watch timeout becomes 15 seconds).

The drop strategy can be changed from the oldest connection to a random connection with the **ip tcp intercept drop-mode** command.

Note The two factors that determine aggressive behavior are related and work together. When *either* of the **high** values is exceeded, aggressive behavior begins. When *both* quantities fall below the **low** value, aggressive behavior ends.

You can change the threshold for triggering aggressive mode based on the total number of incomplete connections. The default values for **low** and **high** are 900 and 1100 incomplete connections, respectively. To change these values, perform the following tasks in global configuration mode:

Task	Command
Set the threshold for stopping aggressive mode.	ip tcp intercept max-incomplete low <i>number</i>
Set the threshold for triggering aggressive mode.	ip tcp intercept max-incomplete high <i>number</i>

You can also change the threshold for triggering aggressive mode based on the number of connection requests received in the last 1-minute sample period. The default values for **low** and **high** are 900 and 1100 connection requests, respectively. To change these values, perform the following tasks in global configuration mode:

Task	Command
Set the threshold for stopping aggressive mode.	ip tcp intercept one-minute low <i>number</i>
Set the threshold for triggering aggressive mode.	ip tcp intercept one-minute high <i>number</i>

Monitor and Maintain TCP Intercept

To display TCP intercept information, perform either of the following tasks in EXEC mode:

Task	Command
Display incomplete connections and established connections.	show tcp intercept connections
Display TCP intercept statistics.	show tcp intercept statistics

Configuration Example

The following configuration defines extended IP access list 101, causing the software to intercept packets for all TCP servers on the 192.168.1.0/24 subnet:

```
ip tcp intercept list 101
!
access-list 101 permit tcp any 192.168.1.0 0.0.0.255
```

Command Reference

This section documents the following new commands:

- [ip tcp intercept connection-timeout](#)
- [ip tcp intercept drop-mode](#)

- [ip tcp intercept finrst-timeout](#)
- [ip tcp intercept list](#)
- [ip tcp intercept max-incomplete high](#)
- [ip tcp intercept max-incomplete low](#)
- [ip tcp intercept mode](#)
- [ip tcp intercept one-minute high](#)
- [ip tcp intercept one-minute low](#)
- [ip tcp intercept watch-timeout](#)
- [show tcp intercept connections](#)
- [show tcp intercept statistics](#)

ip tcp intercept connection-timeout

To change how long a TCP connection will still be managed by the TCP intercept after no activity, use the **ip tcp intercept connection-timeout** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept connection-timeout *seconds*
no ip tcp intercept connection-timeout [*seconds*]

Syntax Description

seconds Time (in seconds) that the software will still manage the connection after no activity. The minimum value is 1 second. The default is 86400 seconds (24 hours).

Default

86400 seconds (24 hours)

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

Example

The following example sets the software to manage the connection for 12 hours (43200 seconds) after no activity:

```
ip tcp intercept connection-timeout 43200
```

ip tcp intercept drop-mode

To set the TCP intercept drop mode, use the **ip tcp intercept drop-mode** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept drop-mode {oldest | random}
no ip tcp intercept drop-mode [oldest | random]

Syntax Description

oldest Software drops the oldest partial connection. This is the default.

random Software drops a randomly selected partial connection.

Default

oldest

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

If the number of incomplete connections exceeds 1100 or the number of connections arriving in the last 1 minute exceeds 1100, the TCP intercept feature becomes more aggressive. When this happens, each new arriving connection causes the oldest partial connection to be deleted, and the initial retransmission timeout is reduced by half to 0.5 seconds (and so the total time trying to establish the connection will be cut in half).

Note that the 1100 thresholds can be configured with the **ip tcp intercept max-incomplete high** and **ip tcp intercept one-minute high** commands.

Use the **ip tcp intercept drop-mode** command to change the dropping strategy from oldest to a random drop.

Example

The following example sets the drop mode to random:

```
ip tcp intercept drop-mode random
```

Related Commands

[ip tcp intercept max-incomplete high](#)

[ip tcp intercept max-incomplete low](#)

[ip tcp intercept one-minute high](#)

[ip tcp intercept one-minute low](#)

ip tcp intercept finrst-timeout

To change how long after receipt of a reset or FIN-exchange the software ceases to manage the connection, use the **ip tcp intercept finrst-timeout** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept finrst-timeout *seconds*
no ip tcp intercept finrst-timeout [*seconds*]

Syntax Description

seconds Time (in seconds) after receiving a reset or FIN-exchange that the software ceases to manage the connection. The minimum value is 1 second. The default is 5 seconds.

Default

5 seconds

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

Even after the two ends of the connection are joined, the software intercepts packets being sent back and forth. Use this command if you need to adjust how soon after receiving a reset or FIN-exchange the software stops intercepting packets.

Example

The following example sets the software to wait for 10 seconds before forgetting about the connection:

```
ip tcp intercept finrst-timeout 10
```

ip tcp intercept list

To enable TCP intercept, use the **ip tcp intercept list** global configuration command. To disable TCP intercept, use the **no** form of this command.

ip tcp intercept list *access-list-number*
no ip tcp intercept list *access-list-number*

Syntax Description

access-list-number Extended access list number in the range 100 to 199.

Default

Disabled

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

The TCP intercept feature intercepts TCP connection attempts and shields servers from TCP SYN-flood attacks, also known as denial-of-service attacks.

TCP packets matching the access list are presented to the TCP intercept code for processing, as determined by the [ip tcp intercept mode](#) command. The TCP intercept code either intercepts or watches the connections.

To have all TCP connection attempts submitted to the TCP intercept code, have the access list match everything.

Example

The following configuration defines access list 101, causing the software to intercept packets for all TCP servers on the 192.168.1.0/24 subnet:

```
ip tcp intercept list 101
!
access-list 101 permit tcp any 192.168.1.0 0.0.0.255
```

Related Commands

access-list (extended)

[ip tcp intercept mode](#)

[show tcp intercept connections](#)

[show tcp intercept statistics](#)

ip tcp intercept max-incomplete high

To define the maximum number of incomplete connections allowed before the software behaves aggressively, use the **ip tcp intercept max-incomplete high** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept max-incomplete high *number*

no ip tcp intercept max-incomplete high [*number*]

Syntax Description

number Defines the number of incomplete connections allowed, above which the software behaves aggressively. The range is 1 to 2147483647. The default is 1100.

Default

1100 incomplete connections

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

If the number of incomplete connections exceeds the *number* configured, the TCP intercept feature becomes aggressive. These are the characteristics of aggressive mode:

- Each new arriving connection causes the oldest partial connection to be deleted.
- The initial retransmission timeout is reduced by half to 0.5 seconds (and so the total time trying to establish the connection is cut in half).
- The watch-timeout is cut in half (from 30 seconds to 15 seconds).

You can change the drop strategy from the oldest connection to a random connection with the **ip tcp intercept drop-mode** command.

Note The two factors that determine aggressive behavior (connection requests and incomplete connections) are related and work together. When the value of *either* [ip tcp intercept one-minute high](#) or [ip tcp intercept max-incomplete high](#) is exceeded, aggressive behavior begins. When *both* connection requests and incomplete connections fall below the values of [ip tcp intercept one-minute low](#) and [ip tcp intercept max-incomplete low](#), aggressive behavior ends.

The software will back off from its aggressive behavior when the number of incomplete connections falls below the number specified by the [ip tcp intercept max-incomplete low](#) command.

Example

The following example allows 1500 incomplete connections before the software takes its aggressive steps:

```
ip tcp intercept max-incomplete high 1500
```

Related Commands

[ip tcp intercept drop-mode](#)
[ip tcp intercept max-incomplete low](#)
[ip tcp intercept one-minute high](#)
[ip tcp intercept one-minute low](#)

ip tcp intercept max-incomplete low

To define the number of incomplete connections below which the software stops behaving aggressively, use the **ip tcp intercept max-incomplete low** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept max-incomplete low *number*
no ip tcp intercept max-incomplete low [*number*]

Syntax Description

number Defines the number of incomplete connections below which the software stops behaving aggressively. The range is 1 to 2147483647. The default is 900.

Default

900 incomplete connections

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

When *both* connection requests and incomplete connections fall below the values of [ip tcp intercept one-minute low](#) and [ip tcp intercept max-incomplete low](#), the TCP intercept feature stops behaving aggressively.

Note The two factors that determine aggressive behavior (connection requests and incomplete connections) are related and work together. When the value of *either* [ip tcp intercept one-minute high](#) or [ip tcp intercept max-incomplete high](#) is exceeded, aggressive behavior begins. When *both* connection requests and incomplete connections fall below the values of [ip tcp intercept one-minute low](#) and [ip tcp intercept max-incomplete low](#), aggressive behavior ends.

See the [ip tcp intercept max-incomplete high](#) command for a description of aggressive behavior.

Example

The following example sets the software to stop behaving aggressively when the number of incomplete connections falls below 1000:

```
ip tcp intercept max-incomplete low 1000
```

Related Commands

[ip tcp intercept drop-mode](#)

[ip tcp intercept max-incomplete high](#)

[ip tcp intercept one-minute high](#)

[ip tcp intercept one-minute low](#)

ip tcp intercept mode

To change the TCP intercept mode, use the **ip tcp intercept mode** global configuration command. To restore the default value, use the **no** form of this command.

```
ip tcp intercept mode {intercept | watch}
no ip tcp intercept mode [intercept | watch]
```

Syntax Description

- intercept** Active mode in which the TCP intercept software intercepts TCP packets from clients to servers that match the configured access list and performs intercept duties. This is the default.
- watch** Monitoring mode in which the software allows connection attempts to pass through the router and watches them until they are established.

Default

intercept

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

When TCP intercept is enabled, it operates in intercept mode by default. In intercept mode, the software actively intercepts TCP SYN packets from clients to servers that match the specified access list. For each SYN, the software responds on behalf of the server with an ACK and SYN, and waits for an ACK of the SYN from the client. When that ACK is received, the original SYN is sent to the server, and the code then performs a three-way handshake with the server. Then the two half-connections are joined.

In watch mode, the software allows connection attempts to pass through the router, but watches them until they become established. If they fail to become established in 30 seconds (or the value set by the **ip tcp intercept watch-timeout** command), a Reset is sent to the server to clear up its state.

Example

The following example sets the mode to watch mode:

```
ip tcp intercept mode watch
```

Related Command

[ip tcp intercept watch-timeout](#)

ip tcp intercept one-minute high

To define the number of connection requests received in the last one-minute sample period before the software behaves aggressively, use the **ip tcp intercept one-minute high** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept one-minute high *number*
no ip tcp intercept one-minute high [*number*]

Syntax Description

number Specifies the number of connection requests that can be received in the last one-minute sample period before the software behaves aggressively. The range is 1 to 2147483647. The default is 1100.

Default

1100 connection requests

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

If the number of connection requests exceeds the *number* value configured, the TCP intercept feature becomes aggressive. These are the characteristics of aggressive mode:

- Each new arriving connection causes the oldest partial connection to be deleted.
- The initial retransmission timeout is reduced by half to 0.5 seconds (and so the total time trying to establish the connection is cut in half).
- The watch-timeout is cut in half (from 30 seconds to 15 seconds).

You can change the drop strategy from the oldest connection to a random connection with the **ip tcp intercept drop-mode** command.

Note The two factors that determine aggressive behavior (connection requests and incomplete connections) are related and work together. When the value of *either* [ip tcp intercept one-minute high](#) or **ip tcp intercept max-incomplete high** is exceeded, aggressive behavior begins. When *both* connection requests and incomplete connections fall below the values of [ip tcp intercept one-minute low](#) and [ip tcp intercept max-incomplete low](#), aggressive behavior ends.

Example

The following example allows 1400 connection requests before the software behaves aggressively:

```
ip tcp intercept one-minute high 1400
```

Related Commands

[ip tcp intercept drop-mode](#)
[ip tcp intercept max-incomplete high](#)
[ip tcp intercept max-incomplete low](#)
[ip tcp intercept one-minute low](#)

ip tcp intercept one-minute low

To define the number of connection requests below which the software stops behaving aggressively, use the **ip tcp intercept one-minute low** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept one-minute low *number*
no ip tcp intercept one-minute low [*number*]

Syntax Description

number Defines the number of connection requests in the last one-minute sample period below which the software stops behaving aggressively. The range is 1 to 2147483647. The default is 900.

Default

900 connection requests

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

When *both* connection requests and incomplete connections fall below the values of [ip tcp intercept one-minute low](#) and [ip tcp intercept max-incomplete low](#), the TCP intercept feature stops behaving aggressively.

Note The two factors that determine aggressive behavior (connection requests and incomplete connections) are related and work together. When the value of *either* [ip tcp intercept one-minute high](#) or [ip tcp intercept max-incomplete high](#) is exceeded, aggressive behavior begins. When *both* connection requests and incomplete connections fall below the values of [ip tcp intercept one-minute low](#) and [ip tcp intercept max-incomplete low](#), aggressive behavior ends.

See the [ip tcp intercept one-minute high](#) command for a description of aggressive behavior.

Example

The following example sets the software to stop behaving aggressively when the number of connection requests falls below 1000:

```
ip tcp intercept one-minute low 1000
```

Related Commands

[ip tcp intercept drop-mode](#)
[ip tcp intercept max-incomplete high](#)
[ip tcp intercept max-incomplete low](#)
[ip tcp intercept one-minute high](#)

ip tcp intercept watch-timeout

To define how long the software will wait for a watched TCP intercept connection to reach established state before sending a reset to the server, use the **ip tcp intercept watch-timeout** global configuration command. To restore the default value, use the **no** form of this command.

ip tcp intercept watch-timeout *seconds*
no ip tcp intercept watch-timeout [*seconds*]

Syntax Description

seconds Time (in seconds) that the software waits for a watched connection to reach established state before sending a Reset to the server. The minimum value is 1 second. The default is 30 seconds.

Default

30 seconds

Command Mode

Global configuration

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

Use this command if you have set the TCP intercept to passive watch mode and you want to change the default time the connection is watched. During aggressive behavior, the watch timeout time is cut in half.

Example

The following example sets the software to wait 60 seconds for a watched connection to reach established state before sending a Reset to the server:

```
ip tcp intercept watch-timeout 60
```

Related Command

[ip tcp intercept mode](#)

show tcp intercept connections

To display TCP incomplete connections or established connections, use the **show tcp intercept connections EXEC** command.

show tcp intercept connections

Syntax Description

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

Sample Display

The following is sample output from the **show tcp intercept connections** command:

```
Router# show tcp intercept connections
Incomplete:
Client                Server                State  Create  Timeout  Mode
172.19.160.17:58190   10.1.1.30:23         SYNRCVD 00:00:09 00:00:05 I
172.19.160.17:57934   10.1.1.30:23         SYNRCVD 00:00:09 00:00:05 I

Established:
Client                Server                State  Create  Timeout  Mode
171.69.232.23:1045   10.1.1.30:23         ESTAB   00:00:08 23:59:54 I
```

[Table 19](#) describes significant fields shown in the display.

Table 19: Show TCP Intercept Connections Field Descriptions

Field	Description
Incomplete:	Rows of information under "Incomplete" indicate connections that are not yet established.
Client	IP address and port of the client.
Server	IP address and port of the server being protected by TCP intercept.
State	SYNRCVD--establishing with client. SYNSENT--establishing with server. ESTAB--established with both, passing data.
Create	Hours:minutes:seconds since the connection was created.
Timeout	Hours:minutes:seconds until the retransmission timeout.
Mode	I--intercept mode. W--watch mode.
Established:	Rows of information under "Established" indicate connections that are established. The fields are the same as those under "Incomplete" except for the Timeout field described below.
Timeout	Hours:minutes:seconds until the connection will timeout, unless the software sees a FIN exchange, in which case this indicates the hours:minutes:seconds until the FIN or RESET timeout.

Related Commands[ip tcp intercept connection-timeout](#)[ip tcp intercept finrst-timeout](#)[ip tcp intercept list](#)[show tcp intercept statistics](#)**show tcp intercept statistics**

To display TCP intercept statistics, use the **show tcp intercept statistics EXEC** command.

show tcp intercept statistics**Syntax Description**

This command has no arguments or keywords.

Command Mode

EXEC

Usage Guidelines

This command first appeared in Cisco IOS Release 11.2 F.

Sample Display

The following is sample output from the **show tcp intercept statistics** command:

```
Router# show tcp intercept statistics
intercepting new connections using access-list 101
2 incomplete, 1 established connections (total 3)
1 minute connection request rate 2 requests/sec
```

Related Commands

[ip tcp intercept connection-timeout](#)

[ip tcp intercept finrst-timeout](#)

[ip tcp intercept list](#)

[show tcp intercept connections](#)

Debug Command

This section describes the command for debugging the TCP intercept.

debug ip tcp intercept

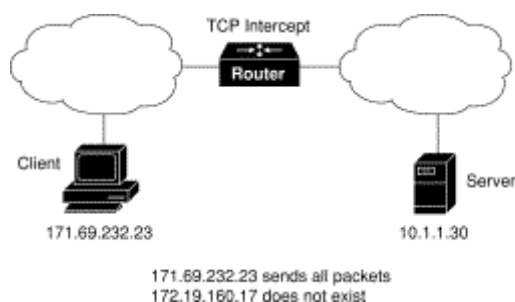
To display TCP intercept statistics, use the **debug ip tcp intercept EXEC** command.

[no] debug ip tcp intercept

Sample Display

[Figure 19](#) illustrates a scenario in which a router configured with TCP intercept operates between a client and a server.

Figure 19: TCP Intercept Debug Scenario



[Figure 20](#) shows sample **debug ip tcp intercept** output, with interspersed commentary. The sample output is based on [Figure 19](#).

Figure 20: Sample Debug IP TCP Intercept Output

```
Router# debug ip tcp intercept
```

[a connection attempt arrives]

```
INTERCEPT: new connection (172.19.160.17:61774) => (10.1.1.30:23)
INTERCEPT: 172.19.160.17:61774 <- ACK+SYN (10.1.1.30:61774)
```

[a second connection attempt arrives]

```
INTERCEPT: new connection (172.19.160.17:62030) => (10.1.1.30:23)
INTERCEPT: 172.19.160.17:62030 <- ACK+SYN (10.1.1.30:62030)
```

[retransmitting to both apparent clients]

```
INTERCEPT: retransmit 2 (172.19.160.17:61774) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmit 2 (172.19.160.17:62030) <- (10.1.1.30:23) SYNRCVD
```

[a third connection attempt arrives]

```
INTERCEPT: new connection (171.69.232.23:1048) => (10.1.1.30:23)
INTERCEPT: 171.69.232.23:1048 <- ACK+SYN (10.1.1.30:1048)
```

[more retransmissions trying to establish with the apparent clients]

```
INTERCEPT: retransmit 4 (172.19.160.17:61774) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmit 4 (172.19.160.17:62030) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmit 2 (171.69.232.23:1048) <- (10.1.1.30:23) SYNRCVD
```

[finished connection with third client, send and retransmit to server]

```
INTERCEPT: 1st half of connection is established (171.69.232.23:1048) => (10.1.1.30:23)
INTERCEPT: (171.69.232.23:1048) SYN -> 10.1.1.30:23
INTERCEPT: retransmit 2 (171.69.232.23:1048) -> (10.1.1.30:23) SYNSENT
```

[server responds, the connection is established, send final ACK]

```
INTERCEPT: 2nd half of connection established (171.69.232.23:1048) => (10.1.1.30:23)
INTERCEPT: (171.69.232.23:1048) ACK -> 10.1.1.30:23
```

[retransmit to first two apparent clients, time out, send resets]

```
INTERCEPT: retransmit 8 (172.19.160.17:61774) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmit 8 (172.19.160.17:62030) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmit 16 (172.19.160.17:61774) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmit 16 (172.19.160.17:62030) <- (10.1.1.30:23) SYNRCVD
INTERCEPT: retransmitting too long (172.19.160.17:61774) => (10.1.1.30:23) SYNRCVD
INTERCEPT: 172.19.160.17:61774 <- RST (10.1.1.30:23)
INTERCEPT: retransmitting too long (172.19.160.17:62030) => (10.1.1.30:23) SYNRCVD
INTERCEPT: 172.19.160.17:62030 <- RST (10.1.1.30:23)
```

[HOME](#) [CONTENTS](#) [PREVIOUS](#) [NEXT](#) [GLOSSARY](#) [FEEDBACK](#) [SEARCH](#) [HELP](#)

[Copyright 1989-1998 © Cisco Systems Inc.](#)