



CHAPTER 24

Preventing Network Attacks

This chapter describes how to prevent network attacks by configuring threat detection, TCP normalization, limiting of TCP and UDP connections, and many other protection features.

This chapter includes the following sections:

- [Configuring Threat Detection, page 24-1](#)
- [Configuring TCP Normalization, page 24-12](#)
- [Configuring Connection Limits and Timeouts, page 24-17](#)
- [Preventing IP Spoofing, page 24-21](#)
- [Configuring the Fragment Size, page 24-22](#)
- [Blocking Unwanted Connections, page 24-22](#)
- [Configuring IP Audit for Basic IPS Support, page 24-23](#)

Configuring Threat Detection

This section describes how to configure scanning threat detection and basic threat detection, and also how to use statistics to analyze threats. Threat detection is available in single mode only.

This section includes the following topics:

- [Configuring Basic Threat Detection, page 24-1](#)
- [Configuring Scanning Threat Detection, page 24-5](#)
- [Configuring and Viewing Threat Statistics, page 24-7](#)

Configuring Basic Threat Detection

Basic threat detection detects activity that might be related to an attack, such as a DoS attack. Basic threat detection is enabled by default.

This section includes the following topics:

- [Basic Threat Detection Overview, page 24-2](#)
- [Configuring Basic Threat Detection, page 24-2](#)
- [Managing Basic Threat Statistics, page 24-4](#)

Basic Threat Detection Overview

Using basic threat detection, the security appliance monitors the rate of dropped packets and security events due to the following reasons:

- Denial by access lists
- Bad packet format (such as invalid-ip-header or invalid-tcp-hdr-length)
- Connection limits exceeded (both system-wide resource limits, and limits set in the configuration)
- DoS attack detected (such as an invalid SPI, Stateful Firewall check failure)
- Basic firewall checks failed (This option is a combined rate that includes all firewall-related packet drops in this bulleted list. It does not include non-firewall-related drops such as interface overload, packets failed at application inspection, and scanning attack detected.)
- Suspicious ICMP packets detected
- Packets failed application inspection
- Interface overload
- Scanning attack detected (This option monitors scanning attacks; for example, the first TCP packet is not a SYN packet, or the TCP connection failed the 3-way handshake. Full scanning threat detection (see the “[Configuring Scanning Threat Detection](#)” section on page 24-5) takes this scanning attack rate information and acts on it by classifying hosts as attackers and automatically shunning them, for example.)
- Incomplete session detection such as TCP SYN attack detected or no data UDP session attack detected

When the security appliance detects a threat, it immediately sends a system log message (733100).

Basic threat detection affects performance only when there are drops or potential threats; even in this scenario, the performance impact is insignificant.

Configuring Basic Threat Detection

To configure basic threat detection, including enabling or disabling it and changing the default limits, perform the following steps:

-
- Step 1** To enable basic threat detection (if you previously disabled it), enter the following command:

```
hostname(config)# threat-detection basic-threat
```

By default, this command enables detection for certain types of security events, including packet drops and incomplete session detections. You can override the default settings for each type of event if desired.

If an event rate is exceeded, then the security appliance sends a system message. The security appliance tracks two types of rates: the average event rate over an interval, and the burst event rate over a shorter burst interval. The burst rate interval is 1/60th of the average rate interval or 10 seconds, whichever is higher. For each received event, the security appliance checks the average and burst rate limits; if both rates are exceeded, then the security appliance sends two separate system messages, with a maximum of one message for each rate type per burst period.

To disable basic threat detection, enter the **no threat-detection basic-threat** command.

[Table 24-1](#) lists the default settings. You can view all these default settings using the **show running-config all threat-detection** command.

Table 24-1 Basic Threat Detection Default Settings

Packet Drop Reason	Trigger Settings	
	Average Rate	Burst Rate
<ul style="list-style-type: none"> DoS attack detected Bad packet format Connection limits exceeded Suspicious ICMP packets detected 	100 drops/sec over the last 600 seconds.	400 drops/sec over the last 10 second period.
	80 drops/sec over the last 3600 seconds.	320 drops/sec over the last 60 second period.
Scanning attack detected	5 drops/sec over the last 600 seconds.	10 drops/sec over the last 10 second period.
	4 drops/sec over the last 3600 seconds.	8 drops/sec over the last 60 second period.
Incomplete session detected such as TCP SYN attack detected or no data UDP session attack detected (combined)	100 drops/sec over the last 600 seconds.	200 drops/sec over the last 10 second period.
	80 drops/sec over the last 3600 seconds.	160 drops/sec over the last 60 second period.
Denial by access lists	400 drops/sec over the last 600 seconds.	800 drops/sec over the last 10 second period.
	320 drops/sec over the last 3600 seconds.	640 drops/sec over the last 60 second period.
<ul style="list-style-type: none"> Basic firewall checks failed Packets failed application inspection 	400 drops/sec over the last 600 seconds.	1600 drops/sec over the last 10 second period.
	320 drops/sec over the last 3600 seconds.	1280 drops/sec over the last 60 second period.
Interface overload	2000 drops/sec over the last 600 seconds.	8000 drops/sec over the last 10 second period.
	1600 drops/sec over the last 3600 seconds.	6400 drops/sec over the last 60 second period.

Step 2 (Optional) To change the default settings for one or more type of event, enter the following command:

```
hostname(config)# threat-detection rate {acl-drop | bad-packet-drop | conn-limit-drop | dos-drop | fw-drop | icmp-drop | inspect-drop | interface-drop | scanning-threat | syn-attack} rate-interval rate_interval average-rate av_rate burst-rate burst_rate
```

For a description of each event type, see the “[Basic Threat Detection Overview](#)” section on page 24-2.

When you use this command with the **scanning-threat** keyword, it is also used in the scanning threat detection feature (see the “[Configuring Scanning Threat Detection](#)” section). The rates you set in this command determine when a host is considered to be an attacker or a target. If you do not set the rates using this command, the default values are used for the scanning threat detection feature as well as the basic threat detection feature. If you do not configure basic threat detection, you can still use this command with the **scanning-threat** keyword to configure the rate limits for scanning threat detection.

The **rate-interface** *rate_interval* argument is between 600 seconds and 2592000 seconds (30 days). The rate interval is used to determine the length of time over which to average the drops. It also determines the burst threshold rate interval (see below).

The **average-rate** *av_rate* argument can be between 0 and 2147483647 in drops/sec.

The **burst-rate** *burst_rate* argument can be between 0 and 2147483647 in drops/sec. The burst rate is calculated as the average rate every *N* seconds, where *N* is the burst rate interval. The burst rate interval is 1/60th of the average rate interval or 10 seconds, whichever is larger.

You can configure up to three different rate intervals for each event type.

The following example enables basic threat detection, and changes the triggers for DoS attacks:

```
hostname(config)# threat-detection basic-threat
hostname(config)# threat-detection rate dos-drop rate-interval 600 average-rate 60
burst-rate 100
```

Managing Basic Threat Statistics

- To view basic threat statistics, enter the following command:

```
hostname# show threat-detection rate [min-display-rate min_display_rate] [acl-drop |
bad-packet-drop | conn-limit-drop | dos-drop | fw-drop | icmp-drop | inspect-drop |
interface-drop | scanning-threat | syn-attack]
```

where the **min-display-rate** *min_display_rate* argument limits the display to statistics that exceed the minimum display rate in events per second. You can set the *min_display_rate* between 0 and 2147483647.

For a description of each event type, see the [“Basic Threat Detection Overview” section on page 24-2](#).

The output shows the average rate in events/sec over two fixed time periods: the last 10 minutes and the last 1 hour. It also shows: the current burst rate in events/sec over the last completed burst interval, which is 1/60th of the average rate interval or 10 seconds, whichever is larger; the number of times the rates were exceeded (triggered); and the total number of events over the time periods.

The security appliance stores the count at the end of each burst period, for a total of 60 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output.

The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.

- To clear basic threat statistics, enter the following command:

```
hostname# clear threat-detection rate
```

The following is sample output from the **show threat-detection rate** command:

```
hostname# show threat-detection rate
```

	Average (eps)	Current (eps)	Trigger	Total events
10-min ACL drop:	0	0	0	16
1-hour ACL drop:	0	0	0	112
1-hour SYN attck:	5	0	2	21438
10-min Scanning:	0	0	29	193

1-hour Scanning:	106	0	10	384776
1-hour Bad pkts:	76	0	2	274690
10-min Firewall:	0	0	3	22
1-hour Firewall:	76	0	2	274844
10-min DoS attck:	0	0	0	6
1-hour DoS attck:	0	0	0	42
10-min Interface:	0	0	0	204
1-hour Interface:	88	0	0	318225

Configuring Scanning Threat Detection

A typical scanning attack consists of a host that tests the accessibility of every IP address in a subnet (by scanning through many hosts in the subnet or sweeping through many ports in a host or subnet). The scanning threat detection feature determines when a host is performing a scan. Unlike IPS scan detection that is based on traffic signatures, the security appliance scanning threat detection feature maintains an extensive database that contains host statistics that can be analyzed for scanning activity.

The host database tracks suspicious activity such as connections with no return activity, access of closed service ports, vulnerable TCP behaviors such as non-random IPID, and many more behaviors.

You can configure the security appliance to send system log messages about an attacker or you can automatically shun the host.



Caution

The scanning threat detection feature can affect the security appliance performance and memory significantly while it creates and gathers host- and subnet-based data structure and information.

This section includes the following topics:

- [Enabling Scanning Threat Detection, page 24-5](#)
- [Managing Shunned Hosts, page 24-6](#)
- [Viewing Attackers and Targets, page 24-7](#)

Enabling Scanning Threat Detection

To configure scanning threat detection, perform the following steps:

Step 1 To enable scanning threat detection, enter the following command:

```
hostname(config)# threat-detection scanning-threat [shun
[except {ip-address ip_address mask | object-group network_object_group_id}]]
```

By default, the system log message 730101 is generated when a host is identified as an attacker.

The **shun** keyword automatically terminates a host connection when the security appliance identifies the host as an attacker, in addition to sending the system log message.

You can except host IP addresses from being shunned by entering the **except ip-address** or **except object-group** keywords. Enter this command multiple times to identify multiple IP addresses or network object groups to exempt from shunning.

Step 2 (Optional) To set the duration of the shun for attacking hosts, enter the following command:

```
hostname(config)# threat-detection scanning-threat shun duration seconds
```

where the *seconds* argument is between 10 and 2592000 seconds. The default is 3600 seconds (1 hour).

- Step 3** (Optional) To change the default event limit for when the security appliance identifies a host as an attacker or as a target, enter the following command:

```
hostname(config)# threat-detection rate scanning-threat rate-interval rate_interval
average-rate av_rate burst-rate burst_rate
```

If the scanning threat rate is exceeded, then the security appliance sends a system message, and optionally shuns the attacker. The security appliance tracks two types of rates: the average event rate over an interval, and the burst event rate over a shorter burst interval. The burst event rate is 1/60th of the average rate interval or 10 seconds, whichever is higher. For each event detected that is considered to be part of a scanning attack, the security appliance checks the average and burst rate limits. If either rate is exceeded for traffic sent from a host, then that host is considered to be an attacker. If either rate is exceeded for traffic received by a host, then that host is considered to be a target.

If you already configured this command as part of the basic threat detection configuration (see the “[Configuring Basic Threat Detection](#)” section on page 24-1), then those settings are shared with the scanning threat detection feature; you cannot configure separate rates for each feature. If you do not set the rates using this command, the default values are used for both the scanning threat detection feature and the basic threat detection feature. The default values are:

Table 24-2 Default Rate Limits for Scanning Threat Detection

Average Rate	Burst Rate
5 drops/sec over the last 600 seconds.	10 drops/sec over the last 10 second period.
5 drops/sec over the last 3600 seconds.	10 drops/sec over the last 60 second period.

The *rate_interval* is between 300 seconds and 2592000 seconds (30 days). The rate interval is used to determine the length of time over which to average the events. It also determines the burst threshold rate interval (see below).

The **average-rate** *av_rate* argument can be between 0 and 2147483647 in drops/sec.

The **burst-rate** *burst_rate* argument can be between 0 and 2147483647 in drops/sec. The burst rate is calculated as the average rate every *N* seconds, where *N* is the burst rate interval. The burst rate interval is 1/60th of the rate interval or 10 seconds, whichever is larger.

You can configure up to three commands with different rate intervals.

The following example enables scanning threat detection and automatically shuns hosts categorized as attackers, except for hosts on the 10.1.1.0 network. The default rate limits for scanning threat detection are also changed.

```
hostname(config)# threat-detection scanning-threat shun except ip-address 10.1.1.0
255.255.255.0
hostname(config)# threat-detection rate scanning-threat rate-interval 1200 average-rate 10
burst-rate 20
hostname(config)# threat-detection rate scanning-threat rate-interval 2400 average-rate 10
burst-rate 20
```

Managing Shunned Hosts

- To view the hosts that are currently shunned, enter the following command:

```
hostname# show threat-detection shun
```

- To release a host from being shunned, enter the following command:

```
hostname# clear threat-detection shun [ip_address [mask]]
```

If you do not specify an IP address, all hosts are cleared from the shun list.

The following is sample output from the **show threat-detection shun** command:

```
hostname# show threat-detection shun
Shunned Host List:
10.1.1.6
192.168.6.7
```

Viewing Attackers and Targets

To view the hosts that the security appliance decides are attackers (including hosts on the shun list), and to view the hosts that are the target of an attack, enter the following command:

```
hostname# show threat-detection scanning-threat [attacker | target]
```

If you do not enter an option, both attackers and target hosts are displayed.

The following is sample output from the **show threat-detection scanning-threat attacker** command:

```
hostname# show threat-detection scanning-threat attacker
10.1.2.3
10.8.3.6
209.165.200.225
```

Configuring and Viewing Threat Statistics

You can configure the security appliance to collect extensive statistics. Threat detection statistics show both allowed and dropped traffic rates. To view statistics for basic threat detection, see the [“Managing Basic Threat Statistics” section on page 24-4](#). By default, statistics for access lists are enabled.



Caution

Enabling statistics can affect the security appliance performance, depending on the type of statistics enabled. The **threat-detection statistics host** command affects performance in a significant way; if you have a high traffic load, you might consider enabling this type of statistics temporarily. The **threat-detection statistics port** command, however, has modest impact.

This section includes the following topics:

- [Configuring Threat Statistics, page 24-7](#)
- [Viewing Threat Statistics, page 24-8](#)

Configuring Threat Statistics

By default, statistics for access lists are enabled. To enable *all* statistics, enter the following command:

```
hostname(config)# threat-detection statistics
```

To enable only certain statistics, enter one or more of the following commands for each statistic type.

- Access lists—To enable statistics for access lists (if they were disabled previously), enter the following command:

```
hostname(config)# threat-detection statistics access-list
```

Access list statistics are only displayed using the **show threat-detection top access-list** command.

- Hosts—To enable statistics for hosts, enter the following command:

```
hostname(config)# threat-detection statistics host
```

The host statistics accumulate for as long as the host is active and in the scanning threat host database. The host is deleted from the database (and the statistics cleared) after 10 minutes of inactivity.

- TCP and UDP ports—To enable statistics for TCP and UDP ports, enter the following command:

```
hostname(config)# threat-detection statistics port
```

- Non-TCP/UDP IP ports—To enable statistics for non-TCP/UDP IP protocols, enter the following command:

```
hostname(config)# threat-detection statistics protocol
```

- TCP Intercept—To enable statistics for attacks intercepted by TCP Intercept (see the [“Configuring Connection Limits and Timeouts”](#) section on page 24-17 to enable TCP Intercept), enter the following command:

```
hostname(config)# threat-detection statistics tcp-intercept [rate-interval minutes]
[burst-rate attacks_per_sec] [average-rate attacks_per_sec]
```

where the **rate-interval** *minutes* argument sets the size of the history monitoring window, between 1 and 1440 minutes. The default is 30 minutes. The security appliance samples the number of attacks 60 times during the rate interval, so for the default 30 minute period, statistics are collected every 60 seconds.

The **burst-rate** *attacks_per_sec* argument sets the threshold for syslog message generation, between 25 and 2147483647. The default is 400 per second. When the burst rate is exceeded, syslog message 733104 is generated.

The **average-rate** *attacks_per_sec* argument sets the average rate threshold for syslog message generation, between 25 and 2147483647. The default is 200 per second. When the average rate is exceeded, syslog message 733105 is generated.

Viewing Threat Statistics

The display output shows the following:

- The average rate in events/sec over fixed time periods.
- The current burst rate in events/sec over the last completed burst interval, which is 1/60th of the average rate interval or 10 seconds, whichever is larger
- The number of times the rates were exceeded (for dropped traffic statistics only)
- The total number of events over the fixed time periods.

The security appliance stores the count at the end of each burst period, for a total of 60 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the **show** command at 3:00:25, then the last 5 seconds are not included in the output.

The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.

To view statistics, enter one of the following commands.

- To view the top 10 statistics, enter the following command:

```
hostname# show threat-detection statistics [min-display-rate min_display_rate] top
[[access-list | host | port-protocol] [rate-1 | rate-2 | rate-3] |
tcp-intercept [all] detail]]
```

where the **min-display-rate** *min_display_rate* argument limits the display to statistics that exceed the minimum display rate in events per second. You can set the *min_display_rate* between 0 and 2147483647.

If you do not enter any options, the top 10 statistics are shown for all categories.

To view the top 10 ACEs that match packets, including both permit and deny ACEs., use the **access-list** keyword. Permitted and denied traffic are not differentiated in this display. If you enable basic threat detection using the **threat-detection basic-threat** command, you can track access list denies using the **show threat-detection rate access-list** command.

To view only host statistics, use the **host** keyword.

To view statistics for ports and protocols, use the **port-protocol** keyword. The **port-protocol** keyword shows the combined statistics of TCP/UDP port and IP protocol types. TCP (protocol 6) and UDP (protocol 17) are not included in the display for IP protocols; TCP and UDP ports are, however, included in the display for ports. If you only enable statistics for one of these types, port or protocol, then you will only view the enabled statistics.

To view TCP Intercept statistics, use the **tcp-intercept** keyword. The display includes the top 10 protected servers under attack. The **all** keyword to shows the history data of all the traced servers. The **detail** keyword shows history sampling data. The security appliance samples the number of attacks 60 times during the rate interval, so for the default 30 minute period, statistics are collected every 60 seconds.

The **rate-1** keyword shows the statistics for the smallest fixed rate intervals available in the display; **rate-2** shows the next largest rate interval; and **rate-3**, if you have three intervals defined, shows the largest rate interval. For example, the display shows statistics for the last 1 hour, 8 hours, and 24 hours. If you set the **rate-1** keyword, the security appliance shows only the 1 hour time interval.

- To view statistics for all hosts or for a specific host or subnet, enter the following command:

```
hostname# show threat-detection statistics [min-display-rate min_display_rate] host
[ip_address [mask]]
```

- To view statistics for all ports or for a specific port or range of ports, enter the following command:

```
hostname# show threat-detection statistics [min-display-rate min_display_rate] port
[start_port[-end_port]]
```

- To view statistics for all IP protocols or for a specific protocol, enter the following command:

```
hostname# show threat-detection statistics [min-display-rate min_display_rate]
protocol [protocol_number | ah | eigrp | esp | gre | icmp | igmp | igmp | igmp | ip | ipinip
| ipsec | nos | ospf | pcp | pim | pptp | snp | tcp | udp]
```

where the *protocol_number* argument is an integer between 0 and 255.

The following is sample output from the **show threat-detection statistics host** command:

```
hostname# show threat-detection statistics host
```

```

                Average(eps)    Current(eps) Trigger          Total events
Host:10.0.0.1: tot-ses:289235 act-ses:22571 fw-drop:0 insp-drop:0 null-ses:21438 bad-acc:0
  1-hour Sent byte:           2938                0      0          10580308
  8-hour Sent byte:            367                0      0          10580308
 24-hour Sent byte:            122                0      0          10580308
  1-hour Sent pkts:            28                 0      0           104043
  8-hour Sent pkts:             3                 0      0           104043
 24-hour Sent pkts:             1                 0      0           104043
 20-min Sent drop:             9                 0      1            10851
  1-hour Sent drop:            3                 0      1            10851
  1-hour Recv byte:           2697                0      0          9712670
  8-hour Recv byte:            337                0      0          9712670
 24-hour Recv byte:            112                0      0          9712670
  1-hour Recv pkts:            29                 0      0          104846
  8-hour Recv pkts:             3                 0      0          104846
 24-hour Recv pkts:             1                 0      0          104846
 20-min Recv drop:             42                 0      3            50567
  1-hour Recv drop:            14                 0      1            50567
Host:10.0.0.0: tot-ses:1 act-ses:0 fw-drop:0 insp-drop:0 null-ses:0 bad-acc:0
  1-hour Sent byte:             0                 0      0             614
  8-hour Sent byte:             0                 0      0             614
 24-hour Sent byte:             0                 0      0             614
  1-hour Sent pkts:             0                 0      0              6
  8-hour Sent pkts:             0                 0      0              6
 24-hour Sent pkts:             0                 0      0              6
 20-min Sent drop:             0                 0      0              4
  1-hour Sent drop:             0                 0      0              4
  1-hour Recv byte:             0                 0      0             706
  8-hour Recv byte:             0                 0      0             706
 24-hour Recv byte:             0                 0      0             706
  1-hour Recv pkts:             0                 0      0              7

```

Table 24-3 shows each field description.

Table 24-3 show threat-detection statistics host Fields

Field	Description
Host	Shows the host IP address.
tot-ses	Shows the total number of sessions for this host since it was added to the database.
act-ses	Shows the total number of active sessions that the host is currently involved in.
fw-drop	Shows the number of firewall drops. Firewall drops is a combined rate that includes all firewall-related packet drops tracked in basic threat detection, including access list denials, bad packets, exceeded connection limits, DoS attack packets, suspicious ICMP packets, TCP SYN attack packets, and no data UDP attack packets. It does not include non-firewall-related drops such as interface overload, packets failed at application inspection, and scanning attack detected.
insp-drop	Shows the number of packets dropped because they failed application inspection.
null-ses	Shows the number of null sessions, which are TCP SYN sessions that did not complete within the 3-second timeout, and UDP sessions that did not have any data sent by its server 3 seconds after the session starts.

Table 24-3 *show threat-detection statistics host Fields (continued)*

Field	Description
bad-acc	Shows the number of bad access attempts to host ports that are in a closed state. When a port is determined to be in a null session (see above), the port state of the host is set to HOST_PORT_CLOSE. Any client accessing the port of the host is immediately classified as a bad access without the need to wait for a timeout.
Average(eps)	Shows the average rate in events/sec over each time period. The security appliance stores the count at the end of each burst period, for a total of 60 completed burst intervals. The unfinished burst interval presently occurring is not included in the average rate. For example, if the average rate interval is 20 minutes, then the burst interval is 20 seconds. If the last burst interval was from 3:00:00 to 3:00:20, and you use the show command at 3:00:25, then the last 5 seconds are not included in the output. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.
Current(eps)	Shows the current burst rate in events/sec over the last completed burst interval, which is 1/60th of the average rate interval or 10 seconds, whichever is larger. For the example specified in the Average(eps) description, the current rate is the rate from 3:19:30 to 3:20:00
Trigger	Shows the number of times the dropped packet rate limits were exceeded. For valid traffic identified in the sent and received bytes and packets rows, this value is always 0, because there are no rate limits to trigger for valid traffic.
Total events	Shows the total number of events over each rate interval. The unfinished burst interval presently occurring is not included in the total events. The only exception to this rule is if the number of events in the unfinished burst interval already exceeds the number of events in the oldest burst interval (#1 of 60) when calculating the total events. In that case, the security appliance calculates the total events as the last 59 complete intervals, plus the events so far in the unfinished burst interval. This exception lets you monitor a large increase in events in real time.
20-min, 1-hour, 8-hour, and 24-hour	Shows statistics for these fixed rate intervals.
Sent byte	Shows the number of successful bytes sent from the host.
Sent pkts	Shows the number of successful packets sent from the host.
Sent drop	Shows the number of packets sent from the host that were dropped because they were part of a scanning attack.
Recv byte	Shows the number of successful bytes received by the host.
Recv pkts	Shows the number of successful packets received by the host.
Recv drop	Shows the number of packets received by the host that were dropped because they were part of a scanning attack.

Configuring TCP Normalization

The TCP normalization feature identifies abnormal packets that the security appliance can act on when they are detected; for example, the security appliance can allow, drop, or clear the packets. TCP normalization helps protect the security appliance from attacks. This section includes the following topics:

- [TCP Normalization Overview, page 24-12](#)
- [Enabling the TCP Normalizer, page 24-12](#)

TCP Normalization Overview

The TCP normalizer includes non-configurable actions and configurable actions. Typically, non-configurable actions that drop or clear connections apply to packets that are always bad. Configurable actions (as detailed in [“Enabling the TCP Normalizer” section on page 24-12](#)) might need to be customized depending on your network needs.

See the following guidelines for TCP normalization:

- The normalizer does not protect from SYN floods. The security appliance includes SYN flood protection in other ways.
- The normalizer always sees the SYN packet as the first packet in a flow unless the security appliance is in loose mode due to failover.

Enabling the TCP Normalizer

This feature uses Modular Policy Framework, so that implementing TCP normalization consists of identifying traffic, specifying the TCP normalization actions, and activating TCP normalization on an interface. See [Chapter 16, “Using Modular Policy Framework,”](#) for more information.

To configure TCP normalization, perform the following steps:

-
- Step 1** To specify the TCP normalization criteria that you want to look for, create a TCP map by entering the following command:

```
hostname(config)# tcp-map tcp-map-name
```

For each TCP map, you can customize one or more settings.

- Step 2** (Optional) Configure the TCP map criteria by entering one or more of the following commands (see [Table 24-4](#)). If you want to use the default settings for all criteria, you do not need to enter any commands for the TCP map. If you want to customize some settings, then the defaults are used for any commands you do not enter. The default configuration includes the following settings:

```
no check-retransmission
no checksum-verification
exceed-mss allow
queue-limit 0 timeout 4
reserved-bits allow
syn-data allow
synack-data drop
invalid-ack drop
seq-past-window drop
tcp-options range 6 7 clear
tcp-options range 9 255 clear
```

```

tcp-options selective-ack allow
tcp-options timestamp allow
tcp-options window-scale allow
ttl-evasion-protection
urgent-flag clear
window-variation allow-connection

```

Table 24-4 *tcp-map Commands*

Command	Notes
check-retransmission	Prevents inconsistent TCP retransmissions.
checksum-verification	Verifies the checksum.
exceed-mss {allow drop}	<p>Sets the action for packets whose data length exceeds the TCP maximum segment size.</p> <p>(Default) The allow keyword allows packets whose data length exceeds the TCP maximum segment size.</p> <p>The drop keyword drops packets whose data length exceeds the TCP maximum segment size.</p>
invalid-ack {allow drop}	<p>Sets the action for packets with an invalid ACK. You might see invalid ACKs in the following instances:</p> <ul style="list-style-type: none"> • In the TCP connection SYN-ACK-received status, if the ACK number of a received TCP packet is not exactly same as the sequence number of the next TCP packet sending out, it is an invalid ACK. • Whenever the ACK number of a received TCP packet is greater than the sequence number of the next TCP packet sending out, it is an invalid ACK. <p>The allow keyword allows packets with an invalid ACK.</p> <p>(Default) The drop keyword drops packets with an invalid ACK.</p> <p>Note TCP packets with an invalid ACK are automatically allowed for WAAS connections.</p>

Table 24-4 *tcp-map Commands (continued)*

Command	Notes
queue-limit <i>pkt_num</i> [timeout <i>seconds</i>]	<p>Sets the maximum number of out-of-order packets that can be buffered and put in order for a TCP connection, between 1 and 250 packets. The default is 0, which means this setting is disabled and the default system queue limit is used depending on the type of traffic:</p> <ul style="list-style-type: none"> • Connections for application inspection (the inspect command), IPS (the ips command), and TCP check-retransmission (the TCP map check-retransmission command) have a queue limit of 3 packets. If the security appliance receives a TCP packet with a different window size, then the queue limit is dynamically changed to match the advertised setting. • For other TCP connections, out-of-order packets are passed through untouched. <p>If you set the queue-limit command to be 1 or above, then the number of out-of-order packets allowed for all TCP traffic matches this setting. For application inspection, IPS, and TCP check-retransmission traffic, any advertised settings are ignored. For other TCP traffic, out-of-order packets are now buffered and put in order instead of passed through untouched.</p> <p>The timeout <i>seconds</i> argument sets the maximum amount of time that out-of-order packets can remain in the buffer, between 1 and 20 seconds; if they are not put in order and passed on within the timeout period, then they are dropped. The default is 4 seconds. You cannot change the timeout for any traffic if the <i>pkt_num</i> argument is set to 0; you need to set the limit to be 1 or above for the timeout keyword to take effect.</p>
reserved-bits { allow clear drop }	<p>Sets the action for reserved bits in the TCP header.</p> <p>(Default) The allow keyword allows packets with the reserved bits in the TCP header.</p> <p>The clear keyword clears the reserved bits in the TCP header and allows the packet.</p> <p>The drop keyword drops the packet with the reserved bits in the TCP header.</p>
seq-past-window { allow drop }	<p>Sets the action for packets that have past-window sequence numbers, namely the sequence number of a received TCP packet is greater than the right edge of the TCP receiving window.</p> <p>The allow keyword allows packets that have past-window sequence numbers. This action is only allowed if the queue-limit command is set to 0 (disabled).</p> <p>(Default) The drop keyword drops packets that have past-window sequence numbers.</p>

Table 24-4 *tcp-map Commands (continued)*

Command	Notes
synack-data { allow drop }	<p>Sets the action for TCP SYNACK packets that contain data.</p> <p>The allow keyword allows TCP SYNACK packets that contain data.</p> <p>(Default) The drop keyword drops TCP SYNACK packets that contain data.</p>
syn-data { allow drop }	<p>Sets the action for SYN packets with data.</p> <p>(Default) The allow keyword allows SYN packets with data.</p> <p>The drop keyword drops SYN packets with data.</p>
tcp-options { selective-ack timestamp window-scale } { allow clear } Or tcp-options range <i>lower upper</i> { allow clear drop }	<p>Sets the action for packets with TCP options, including the selective-ack, timestamp, or window-scale TCP options.</p> <p>(Default) The allow keyword allows packets with the specified option.</p> <p>(Default for range) The clear keyword clears the option and allows the packet.</p> <p>The drop keyword drops the packet with the specified option.</p> <p>The selective-ack keyword sets the action for the SACK option.</p> <p>The timestamp keyword sets the action for the timestamp option. Clearing the timestamp option disables PAWS and RTT.</p> <p>The window-scale keyword sets the action for the window scale mechanism option.</p> <p>The range keyword specifies a range of options. The <i>lower</i> argument sets the lower end of the range as 6, 7, or 9 through 255.</p> <p>The <i>upper</i> argument sets the upper end of the range as 6, 7, or 9 through 255.</p>
ttl-evasion-protection	<p>Disables the TTL evasion protection. Do not enter this command if you want to prevent attacks that attempt to evade security policy.</p> <p>For example, an attacker can send a packet that passes policy with a very short TTL. When the TTL goes to zero, a router between the security appliance and the endpoint drops the packet. It is at this point that the attacker can send a malicious packet with a long TTL that appears to the security appliance to be a retransmission and is passed. To the endpoint host, however, it is the first packet that has been received by the attacker. In this case, an attacker is able to succeed without security preventing the attack.</p>

Table 24-4 tcp-map Commands (continued)

Command	Notes
urgent-flag {allow clear}	<p>Sets the action for packets with the URG flag. The URG flag is used to indicate that the packet contains information that is of higher priority than other data within the stream. The TCP RFC is vague about the exact interpretation of the URG flag, therefore end systems handle urgent offsets in different ways, which may make the end system vulnerable to attacks.</p> <p>The allow keyword allows packets with the URG flag.</p> <p>(Default) The clear keyword clears the URG flag and allows the packet.</p>
window-variation {allow drop}	<p>Sets the action for a connection that has changed its window size unexpectedly. The window size mechanism allows TCP to advertise a large window and to subsequently advertise a much smaller window without having accepted too much data. From the TCP specification, “shrinking the window” is strongly discouraged. When this condition is detected, the connection can be dropped.</p> <p>(Default) The allow keyword allows connections with a window variation.</p> <p>The drop keyword drops connections with a window variation.</p>

- Step 3** To identify the traffic, add a class map using the **class-map** command. See the “[Creating a Layer 3/4 Class Map for Through Traffic](#)” section on page 16-5 for more information.

For example, you can match all traffic using the following commands:

```
hostname(config)# class-map TCPNORM
hostname(config-cmap)# match any
```

To match specific traffic, you can match an access list:

```
hostname(config)# access list TCPNORM extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map TCP_norm_class
hostname(config-cmap)# match access-list TCPNORM
```

- Step 4** To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from [Step 1](#).

For example:

```
hostname(config)# policy-map TCP_norm_policy
hostname(config-pmap)# class TCP_norm_class
hostname(config-pmap-c)#
```

- Step 5** Apply the TCP map to the class map by entering the following command.

```
hostname(config-pmap-c)# set connection advanced-options tcp-map-name
```

- Step 6** To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy polycmap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. Only one global policy is allowed. Interface service policies take precedence over the global service policy for a given feature. For example, if you have a global policy with inspections, and an interface policy with TCP normalization, then both inspections and TCP normalization are applied to the interface. However, if you have a global policy with inspections, and an interface policy with inspections, then only the interface policy inspections are applied to that interface.

For example, to allow urgent flag and urgent offset packets for all traffic sent to the range of TCP ports between the well known FTP data port and the Telnet port, enter the following commands:

```
hostname(config)# tcp-map tmap
hostname(config-tcp-map)# urgent-flag allow
hostname(config-tcp-map)# class-map urg-class
hostname(config-cmap)# match port tcp range ftp-data telnet
hostname(config-cmap)# policy-map pmap
hostname(config-pmap)# class urg-class
hostname(config-pmap-c)# set connection advanced-options tmap
hostname(config-pmap-c)# service-policy pmap global
```

Configuring Connection Limits and Timeouts

This section describes how to set maximum TCP and UDP connections, maximum embryonic connections, maximum per-client connections, connection timeouts, dead connection detection, and how to disable TCP sequence randomization. You can set limits for connections that go through the security appliance, or for management connections to the security appliance. This section contains the following topics:

- [Connection Limit Overview, page 24-17](#)
- [Enabling Connection Limits and Timeouts, page 24-19](#)



Note

You can also configure maximum connections, maximum embryonic connections, and TCP sequence randomization in the NAT configuration. If you configure these settings for the same traffic using both methods, then the security appliance uses the lower limit. For TCP sequence randomization, if it is disabled using either method, then the security appliance disables TCP sequence randomization.

Connection Limit Overview

This section describes why you might want to limit connections, and includes the following topics:

- [TCP Intercept Overview, page 24-18](#)
- [Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility, page 24-18](#)
- [Dead Connection Detection \(DCD\) Overview, page 24-18](#)
- [TCP Sequence Randomization Overview, page 24-18](#)

TCP Intercept Overview

Limiting the number of embryonic connections protects you from a DoS attack. The security appliance uses the per-client limits and the embryonic connection limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets. An embryonic connection is a connection request that has not finished the necessary handshake between source and destination. TCP Intercept uses the SYN cookies algorithm to prevent TCP SYN-flooding attacks. A SYN-flooding attack consists of a series of SYN packets usually originating from spoofed IP addresses. The constant flood of SYN packets keeps the server SYN queue full, which prevents it from servicing connection requests. When the embryonic connection threshold of a connection is crossed, the security appliance acts as a proxy for the server and generates a SYN-ACK response to the client SYN request. When the security appliance receives an ACK back from the client, it can then authenticate the client and allow the connection to the server.

To view TCP Intercept statistics, including the top 10 servers under attack, see the [“Configuring and Viewing Threat Statistics”](#) section on page 24-7.

Disabling TCP Intercept for Management Packets for Clientless SSL Compatibility

By default, TCP management connections have TCP Intercept always enabled. When TCP Intercept is enabled, it intercepts the 3-way TCP connection establishment handshake packets and thus deprives the security appliance from processing the packets for clientless SSL. Clientless SSL requires the ability to process the 3-way handshake packets to provide selective ACK and other TCP options for clientless SSL connections. To disable TCP Intercept for management traffic, you can set the embryonic connection limit; only after the embryonic connection limit is reached is TCP Intercept enabled.

Dead Connection Detection (DCD) Overview

DCD detects a dead connection and allows it to expire, without expiring connections that can still handle traffic. You configure DCD when you want idle, but valid connections to persist.

When you enable DCD, idle timeout behavior changes. With idle timeout, DCD probes are sent to each of the two end-hosts to determine the validity of the connection. If an end-host fails to respond after probes are sent at the configured intervals, the connection is freed, and reset values, if configured, are sent to each of the end-hosts. If both end-hosts respond that the connection is valid, the activity timeout is updated to the current time and the idle timeout is rescheduled accordingly.

Enabling DCD changes the behavior of idle-timeout handling in the TCP normalizer. DCD probing resets the idle timeout on the connections seen in the **show conn** command. To determine when a connection that has exceeded the configured timeout value in the timeout command but is kept alive due to DCD probing, the **show service-policy** command includes counters to show the amount of activity from DCD.

TCP Sequence Randomization Overview

Each TCP connection has two ISNs: one generated by the client and one generated by the server. The security appliance randomizes the ISN of the TCP SYN passing in both the inbound and outbound directions.

Randomizing the ISN of the protected host prevents an attacker from predicting the next ISN for a new connection and potentially hijacking the new session.

TCP initial sequence number randomization can be disabled if required. For example:

- If another in-line firewall is also randomizing the initial sequence numbers, there is no need for both firewalls to be performing this action, even though this action does not affect the traffic.
- If you use eBGP multi-hop through the security appliance, and the eBGP peers are using MD5. Randomization breaks the MD5 checksum.
- You use a WAAS device that requires the security appliance not to randomize the sequence numbers of connections.

Enabling Connection Limits and Timeouts

To set connection limits and timeouts, perform the following steps:

- Step 1** To identify the traffic, add a class map using the **class-map** command. See the “[Creating a Layer 3/4 Class Map for Through Traffic](#)” section on page 16-5 or the “[Creating a Layer 3/4 Class Map for Management Traffic](#)” section on page 16-7 for more information.

For example, you can match all traffic using the following commands:

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
```

To match specific traffic, you can match an access list:

```
hostname(config)# access list CONNS extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map CONNS
hostname(config-cmap)# match access-list CONNS
```

- Step 2** To add or edit a policy map that sets the actions to take with the class map traffic, enter the following commands:

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

where the *class_map_name* is the class map from [Step 1](#).

For example:

```
hostname(config)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)#
```

- Step 3** To set maximum connection limits or whether TCP sequence randomization is enabled, enter the following command:

```
hostname(config-pmap-c)# set connection {[conn-max n] [embryonic-conn-max n]
[per-client-embryonic-max n] [per-client-max n] [random-sequence-number {enable |
disable}]}
```

where the **conn-max** *n* argument sets the maximum number of simultaneous TCP and/or UDP connections that are allowed, between 0 and 65535. The default is 0, which allows unlimited connections.

The **embryonic-conn-max** *n* argument sets the maximum number of simultaneous embryonic connections allowed, between 0 and 65535. The default is 0, which allows unlimited connections.

The **per-client-embryonic-max** *n* argument sets the maximum number of simultaneous embryonic connections allowed per client, between 0 and 65535. The default is 0, which allows unlimited connections.

The **per-client-max** *n* argument sets the maximum number of simultaneous connections allowed per client, between 0 and 65535. The default is 0, which allows unlimited connections.

The **random-sequence-number** {**enable** | **disable**} keyword enables or disables TCP sequence number randomization. See the “[TCP Sequence Randomization Overview](#)” section on page 24-18 section for more information.

You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The security appliance combines the command into one line in the running configuration.



Note For management traffic, you can only set the **conn-max** and **embryonic-conn-max** keywords.

Step 4 To set connection timeouts, enter the following command:

```
hostname(config-pmap-c)# set connection timeout {[embryonic hh:mm:ss] {tcp hh:mm:ss
[reset]} [half-closed hh:mm:ss] [dcd hh:mm:ss [max_retries]]}
```

where the **embryonic** *hh:mm:ss* keyword sets the timeout period until a TCP embryonic (half-open) connection is closed, between 0:0:5 and 1193:00:00. The default is 0:0:30. You can also set this value to 0, which means the connection never times out.

The **tcp** *hh:mm:ss* keyword sets the idle timeout between 0:5:0 and 1193:00:00. The default is 1:0:0. You can also set this value to 0, which means the connection never times out. The **reset** keyword sends a reset to TCP endpoints when the connection times out. The security appliance sends the reset packet only in response to a host sending another packet for the timed-out flow (on the same source and destination port). The host then removes the connection from its connection table after receiving the reset packet. The host application can then attempt to establish a new connection using a SYN packet.

The **half-closed** *hh:mm:ss* keyword sets the idle timeout between 0:5:0 and 1193:00:00. The default is 0:10:0. Half-closed connections are not affected by DCD. Also, the security appliance does not send a reset when taking down half-closed connections.

The **dcd** keyword enables DCD. DCD detects a dead connection and allows it to expire, without expiring connections that can still handle traffic. You configure DCD when you want idle, but valid connections to persist. After a TCP connection times out, the security appliance sends DCD probes to the end hosts to determine the validity of the connection. If one of the end hosts fails to respond after the maximum retries are exhausted, the security appliance frees the connection. If both end hosts respond that the connection is valid, the security appliance updates the activity timeout to the current time and reschedules the idle timeout accordingly. The *retry-interval* sets the time duration in *hh:mm:ss* format to wait after each unresponsive DCD probe before sending another probe, between 0:0:1 and 24:0:0. The default is 0:0:15. The *max-retries* sets the number of consecutive failed retries for DCD before declaring the connection as dead. The minimum value is 1 and the maximum value is 255. The default is 5.

You can enter this command all on one line (in any order), or you can enter each attribute as a separate command. The command is combined onto one line in the running configuration.



Note This command is not available for management traffic.

Step 5 To activate the policy map on one or more interfaces, enter the following command:

```
hostname(config)# service-policy polycmap_name {global | interface interface_name}
```

where *policy_map_name* is the policy map you configured in [Step 2](#). To apply the policy map to traffic on all the interfaces, use the **global** keyword. To apply the policy map to traffic on a specific interface, use the **interface** *interface_name* option, where *interface_name* is the name assigned to the interface with the **nameif** command.

Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.

The following example sets the connection limits and timeouts for all traffic:

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
hostname(config-cmap)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)# set connection conn-max 1000 embryonic-conn-max 3000
hostname(config-pmap-c)# set connection timeout tcp 2:0:0 embryonic 0:40:0 half-closed
0:20:0 dcd
hostname(config-pmap-c)# service-policy CONNS interface outside
```

You can enter **set connection** commands with multiple parameters or you can enter each parameter as a separate command. The security appliance combines the commands into one line in the running configuration. For example, if you entered the following two commands in class configuration mode:

```
hostname(config-pmap-c)# set connection conn-max 600
hostname(config-pmap-c)# set connection embryonic-conn-max 50
```

the output of the **show running-config policy-map** command would display the result of the two commands in a single, combined command:

```
set connection conn-max 600 embryonic-conn-max 50
```

Preventing IP Spoofing

This section lets you enable Unicast Reverse Path Forwarding on an interface. Unicast RPF guards against IP spoofing (a packet uses an incorrect source IP address to obscure its true source) by ensuring that all packets have a source IP address that matches the correct source interface according to the routing table.

Normally, the security appliance only looks at the destination address when determining where to forward the packet. Unicast RPF instructs the security appliance to also look at the source address; this is why it is called Reverse Path Forwarding. For any traffic that you want to allow through the security appliance, the security appliance routing table must include a route back to the source address. See RFC 2267 for more information.

For outside traffic, for example, the security appliance can use the default route to satisfy the Unicast RPF protection. If traffic enters from an outside interface, and the source address is not known to the routing table, the security appliance uses the default route to correctly identify the outside interface as the source interface.

If traffic enters the outside interface from an address that is known to the routing table, but is associated with the inside interface, then the security appliance drops the packet. Similarly, if traffic enters the inside interface from an unknown source address, the security appliance drops the packet because the matching route (the default route) indicates the outside interface.

Unicast RPF is implemented as follows:

- ICMP packets have no session, so each packet is checked.
- UDP and TCP have sessions, so the initial packet requires a reverse route lookup. Subsequent packets arriving during the session are checked using an existing state maintained as part of the session. Non-initial packets are checked to ensure they arrived on the same interface used by the initial packet.

To enable Unicast RPF, enter the following command:

```
hostname(config)# ip verify reverse-path interface interface_name
```

Configuring the Fragment Size

By default, the security appliance allows up to 24 fragments per IP packet, and up to 200 fragments awaiting reassembly. You might need to let fragments on your network if you have an application that routinely fragments packets, such as NFS over UDP. However, if you do not have an application that fragments traffic, we recommend that you do not allow fragments through the security appliance. Fragmented packets are often used as DoS attacks. To set disallow fragments, enter the following command:

```
hostname(config)# fragment chain 1 [interface_name]
```

Enter an interface name if you want to prevent fragmentation on a specific interface. By default, this command applies to all interfaces.

Blocking Unwanted Connections

If you know that a host is attempting to attack your network (for example, system log messages show an attack), then you can block (or shun) connections based on the source IP address and other identifying parameters. No new connections can be made until you remove the shun.



Note

If you have an IPS that monitors traffic, such as an AIP SSM, then the IPS can shun connections automatically.

To shun a connection manually, perform the following steps:

Step 1 If necessary, view information about the connection by entering the following command:

```
hostname# show conn
```

The security appliance shows information about each connection, such as the following:

```
TCP out 64.101.68.161:4300 in 10.86.194.60:23 idle 0:00:00 bytes 1297 flags UIO
```

Step 2 To shun connections from the source IP address, enter the following command:

```
hostname(config)# shun src_ip [dst_ip src_port dest_port [protocol]] [vlan vlan_id]
```

If you enter only the source IP address, then all future connections are shunned; existing connections remain active.

To drop an existing connection, as well as blocking future connections from the source IP address, enter the destination IP address, source and destination ports, and the protocol. By default, the protocol is 0 for IP.

For multiple context mode, you can enter this command in the admin context, and by specifying a VLAN ID that is assigned to an interface in other contexts, you can shun the connection in other contexts.

Step 3 To remove the shun, enter the following command:

```
hostname(config)# no shun src_ip [vlan vlan_id]
```

Configuring IP Audit for Basic IPS Support

The IP audit feature provides basic IPS support for a security appliance that does not have an AIP SSM. It supports a basic list of signatures, and you can configure the security appliance to perform one or more actions on traffic that matches a signature.

To enable IP audit, perform the following steps:

-
- Step 1** To define an IP audit policy for informational signatures, enter the following command:

```
hostname(config)# ip audit name name info [action [alarm] [drop] [reset]]
```

Where **alarm** generates a system message showing that a packet matched a signature, **drop** drops the packet, and **reset** drops the packet and closes the connection. If you do not define an action, then the default action is to generate an alarm.

- Step 2** To define an IP audit policy for attack signatures, enter the following command:

```
hostname(config)# ip audit name name attack [action [alarm] [drop] [reset]]
```

Where **alarm** generates a system message showing that a packet matched a signature, **drop** drops the packet, and **reset** drops the packet and closes the connection. If you do not define an action, then the default action is to generate an alarm.

- Step 3** To assign the policy to an interface, enter the following command:

```
ip audit interface interface_name policy_name
```

- Step 4** To disable signatures, or for more information about signatures, see the **ip audit signature** command in the *Cisco Security Appliance Command Reference*.
-

