



## Context-Based Access Control Commands

---

This chapter describes Context-based Access Control (CBAC) commands. CBAC intelligently filters TCP and User Datagram Protocol packets on the basis of application-layer protocol session information and can be used for intranets, extranets and internets. Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall's access lists to allow return traffic and additional data connections for permissible sessions (sessions that originated from within the protected internal network).

To find complete descriptions of other commands used when configuring CBAC, refer to the *Cisco IOS Command Reference Master Index* or search online.

For configuration information, refer to the chapter “Configuring Context-Based Access Control” in the *Cisco IOS Security Configuration Guide*.

# clear ip urlfilter cache

To clear the cache table, use the **clear ip urlfilter cache** command in EXEC mode.

```
clear ip urlfilter cache {ip-address | all}
```

Syntax Description		
	<i>ip-address</i>	Clears the cache table of a specified server IP address.
	<b>all</b>	Clears the cache table completely.

**Defaults** This command is not enabled.

**Command Modes** EXEC

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Usage Guidelines** The cache table consists of the most recently requested IP addresses and the respective authorization status for each IP address.

**Examples** The following example shows how to clear the cache table of IP address 172.18.139.21:

```
clear ip urlfilter cache 172.18.139.21
```

The following example shows how to clear the cache table of all IP addresses:

```
clear ip urlfilter cache all
```

Related Commands	Command	Description
	<a href="#">ip urlfilter cache</a>	Configures cache parameters.
	<a href="#">show ip urlfilter cache</a>	Displays the destination IP addresses that are cached into the cache table.

# ip inspect alert-off

To disable Context-based Access Control (CBAC) alert messages, which are displayed on the console, use the **ip inspect alert off** command in global configuration mode. To enable CBAC alert messages, use the **no** form of this command.

**ip inspect alert-off**

**no ip inspect alert-off**

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** Alert messages are displayed.

---

**Command Modes** Global configuration

---

Command History	Release	Modification
	12.0(5)T	This command was introduced.

---

---

**Usage Guidelines** Use the **ip inspect alert-off** command to disable alert messages.

---

**Examples** The following example turns on CBAC alert messages:

```
ip inspect alert-off
```

# ip inspect audit-trail

To turn on Context-based Access Control (CBAC) audit trail messages, which will be displayed on the console after each CBAC session closes, use the **ip inspect audit-trail** command in global configuration mode. To turn off CBAC audit trail message, use the **no** form of this command.

**ip inspect audit-trail**

**no ip inspect audit-trail**

**Syntax Description** This command has no arguments or keywords.

**Defaults** Audit trail messages are not displayed.

**Command Modes** Global configuration

Command History	Release	Modification
	11.2 P	This command was introduced.

**Usage Guidelines** Use this command to turn on CBAC audit trail messages.

**Examples** The following example turns on CBAC audit trail messages:

```
ip inspect audit-trail
```

Afterward, audit trail messages such as the following are displayed:

```
%FW-6-SESS_AUDIT_TRAIL: tcp session initiator (192.168.1.13:33192) sent 22 bytes --
responder (192.168.129.11:25) sent 208 bytes
%FW-6-SESS_AUDIT_TRAIL: ftp session initiator 192.168.1.13:33194) sent 336 bytes --
responder (192.168.129.11:21) sent 325 bytes
```

These messages are examples of audit trail messages. To determine which protocol was inspected, refer to the responder's port number. The port number follows the responder's IP address.

# ip inspect dns-timeout

To specify the Domain Name System (DNS) idle timeout (the length of time during which a DNS name lookup session will still be managed while there is no activity), use the **ip inspect dns-timeout** command in global configuration mode. To reset the timeout to the default of 5 seconds, use the **no** form of this command.

**ip inspect dns-timeout** *seconds*

**no ip inspect dns-timeout**

<b>Syntax Description</b>	<i>seconds</i>	Specifies the length of time in seconds, for which a DNS name lookup session will still be managed while there is no activity. The default is 5 seconds.
---------------------------	----------------	--

<b>Defaults</b>	5 seconds
-----------------	-----------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.2 P	This command was introduced.

**Usage Guidelines**

When the software detects a valid User Datagram Protocol packet for a new DNS name lookup session, if Context-based Access Control (CBAC) inspection is configured for UDP, the software establishes state information for the new DNS session.

If the software detects no packets for the DNS session for a time period defined by the DNS idle timeout, the software will not continue to manage state information for the session.

The DNS idle timeout applies to all DNS name lookup sessions inspected by CBAC.

The DNS idle timeout value overrides the global UDP timeout. The DNS idle timeout value also enters aggressive mode and overrides any timeouts specified for specific interfaces when you define a set of inspection rules with the **ip inspect name** command.

**Examples**

The following example sets the DNS idle timeout to 30 seconds:

```
ip inspect dns-timeout 30
```

The following example sets the DNS idle timeout back to the default (5 seconds):

```
no ip inspect dns-timeout
```

# ip inspect

To apply a set of inspection rules to an interface, use the **ip inspect** command in interface configuration mode. To remove the set of rules from the interface, use the **no** form of this command.

**ip inspect** *inspection-name* {**in** | **out**}

**no ip inspect** *inspection-name* {**in** | **out**}

## Syntax Description

<i>inspection-name</i>	Identifies which set of inspection rules to apply.
<b>in</b>	Applies the inspection rules to inbound traffic.
<b>out</b>	Applies the inspection rules to outbound traffic.

## Defaults

If no set of inspection rules is applied to an interface, no traffic will be inspected by CBAC.

## Command Modes

Interface configuration

## Command History

Release	Modification
11.2	This command was introduced.

## Usage Guidelines

Use this command to apply a set of inspection rules to an interface.

Typically, if the interface connects to the external network, you apply the inspection rules to outbound traffic; alternately, if the interface connects to the internal network, you apply the inspection rules to inbound traffic.

If you apply the rules to outbound traffic, then return inbound packets will be permitted if they belong to a valid connection with existing state information. This connection must be initiated with an outbound packet.

If you apply the rules to inbound traffic, then return outbound packets will be permitted if they belong to a valid connection with existing state information. This connection must be initiated with an inbound packet.

## Examples

The following example applies a set of inspection rules named “outboundrules” to an external interface’s outbound traffic. This causes inbound IP traffic to be permitted only if the traffic is part of an existing session, and to be denied if the traffic is not part of an existing session.

```
interface serial0
 ip inspect outboundrules out
```

## Related Commands

Command	Description
<a href="#">ip inspect name</a>	Defines a set of inspection rules.

# ip inspect hashtable

To change the size of the session hash table, use the **ip inspect hashtable** command in global configuration mode. To restore the size of the session hash table to the default, use the **no** form of this command.

**ip inspect hashtable** *number*

**no ip inspect hashtable** *number*

## Syntax Description

<i>number</i>	Size of the hash table in terms of buckets. Possible values for the hash table are 1024, 2048, 4096, and 8192; the default value is 1024.
---------------	---

## Defaults

1024 buckets

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(8)T	This command was introduced.

## Usage Guidelines

Use the **ip inspect hashtable** command to increase the size of the hash table when the number of concurrent sessions increases or to reduce the search time for the session. Collisions in a hash table result in poor hash function distribution because many entries are hashed into the same bucket for certain patterns of addresses. Even if a hash function distribution evenly dispenses the input across all of the buckets, a small hash table size will not scale well if there are a large number of sessions. As the number of sessions increase, the collisions increase, which increases the length of the linked lists, thereby, deteriorating the throughput performance.



### Note

You should increase the hash table size when the total number of sessions running through the context-based access control (CBAC) router is approximately twice the current hash size; decrease the hash table size when the total number of sessions is reduced to approximately half the current hash size. Essentially, try to maintain a 1:1 ratio between the number of sessions and the size of the hash table.

## Examples

The following example shows how to change the size of the session hash table to 2048 buckets:

```
ip inspect hashtable 2048
```

# ip inspect max-incomplete high

To define the number of existing half-open sessions that will cause the software to start deleting half-open sessions, use the **ip inspect max-incomplete high** command in global configuration mode. To reset the threshold to the default of 500 half-open sessions, use the **no** form of this command.

**ip inspect max-incomplete high** *number*

**no ip inspect max-incomplete high**

<b>Syntax Description</b>	<i>number</i>	Specifies the number of existing half-open sessions that will cause the software to start deleting half-open sessions. The default is 500 half-open sessions.
---------------------------	---------------	---

<b>Defaults</b>	500 half-open sessions
-----------------	------------------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.2 P	This command was introduced.

**Usage Guidelines**

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, “half-open” means that the session has not reached the established state. For User Datagram Protocol, “half-open” means that the firewall has detected traffic from one direction only.

Context-based Access Control (CBAC) measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are counted in the total number and rate measurements. Measurements are made once a minute.

When the number of existing half-open sessions rises above a threshold (the **max-incomplete high** number), the software will delete half-open sessions as required to accommodate new connection requests. The software will continue to delete half-open requests as necessary, until the number of existing half-open sessions drops below another threshold (the **max-incomplete low** number).

The global value specified for this threshold applies to all TCP and UDP connections inspected by CBAC.

**Examples**

The following example causes the software to start deleting half-open sessions when the number of existing half-open sessions rises above 900, and to stop deleting half-open sessions when the number drops below 800:

```
ip inspect max-incomplete high 900
ip inspect max-incomplete low 800
```

Related Commands	Command	Description
	<a href="#">ip inspect max-incomplete low</a>	Defines the number of existing half-open sessions that will cause the software to stop deleting half-open sessions.
	<a href="#">ip inspect one-minute high</a>	Defines the rate of new unestablished sessions that will cause the software to start deleting half-open sessions.
	<a href="#">ip inspect one-minute low</a>	Defines the rate of new unestablished TCP sessions that will cause the software to stop deleting half-open sessions.
	<a href="#">ip inspect tcp max-incomplete host</a>	Specifies the threshold and blocking time values for TCP host-specific denial-of-service detection and prevention.

# ip inspect max-incomplete low

To define the number of existing half-open sessions that will cause the software to stop deleting half-open sessions, use the **ip inspect max-incomplete low** command in global configuration mode. To reset the threshold to the default of 400 half-open sessions, use the **no** form of this command.

**ip inspect max-incomplete low** *number*

**no ip inspect max-incomplete low**

## Syntax Description

<i>number</i>	Specifies the number of existing half-open sessions that will cause the software to stop deleting half-open sessions. The default is 400 half-open sessions.
---------------	--

## Defaults

400 half-open sessions.

## Command Modes

Global configuration

## Command History

Release	Modification
11.2 P	This command was introduced.

## Usage Guidelines

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, “half-open” means that the session has not reached the established state. For User Datagram Protocol, “half-open” means that the firewall has detected traffic from one direction only.

Context-based Access Control (CBAC) measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are counted in the total number and rate measurements. Measurements are made once a minute.

When the number of existing half-open sessions rises above a threshold (the **max-incomplete high** number), the software will delete half-open sessions as required to accommodate new connection requests. The software will continue to delete half-open requests as necessary, until the number of existing half-open sessions drops below another threshold (the **max-incomplete low** number).

The global value specified for this threshold applies to all TCP and UDP connections inspected by CBAC.

## Examples

The following example causes the software to start deleting half-open sessions when the number of existing half-open sessions rises above 900, and to stop deleting half-open sessions when the number drops below 800:

```
ip inspect max-incomplete high 900
ip inspect max-incomplete low 800
```

Related Commands	Command	Description
	<a href="#">ip inspect max-incomplete high</a>	Defines the number of existing half-open sessions that will cause the software to start deleting half-open sessions.
	<a href="#">ip inspect one-minute high</a>	Defines the rate of new unestablished sessions that will cause the software to start deleting half-open sessions.
	<a href="#">ip inspect one-minute low</a>	Defines the rate of new unestablished TCP sessions that will cause the software to stop deleting half-open sessions.
	<a href="#">ip inspect tcp max-incomplete host</a>	Specifies the threshold and blocking time values for TCP host-specific denial-of-service detection and prevention.

# ip inspect name

To define a set of inspection rules, use the **ip inspect name** command in global configuration mode. To remove the inspection rule for a protocol or to remove the entire set of inspection rules, use the **no** form of this command.

```
ip inspect name inspection-name protocol [alert {on | off}] [audit-trail {on | off}]
[timeout seconds]
```

```
no ip inspect name [inspection-name protocol]
```

## HTTP Inspection Syntax

```
ip inspect name inspection-name http [urlfilter] [java-list access-list] [alert {on | off}]
[audit-trail {on | off}] [timeout seconds] (Java protocol only)
```

```
no ip inspect name inspection-name protocol (removes the inspection rule for a protocol)
```

## RPC Inspection Syntax

```
ip inspect name inspection-name rpc program-number number [wait-time minutes] [alert {on |
off}] [audit-trail {on | off}] [timeout seconds] (RPC protocol only)
```

```
no ip inspect name inspection-name protocol (removes the inspection rule for a protocol)
```

## Fragment Inspection Syntax

```
ip inspect name inspection-name fragment [max number timeout seconds]
```

```
no ip inspect name inspection-name fragment (removes fragment inspection for a rule)
```

## Syntax Description

<i>inspection-name</i>	Names the set of inspection rules. If you want to add a protocol to an existing set of rules, use the same <i>inspection-name</i> as the existing set of rules.  <b>Note</b> The <i>inspection-name</i> cannot exceed 16 characters; otherwise, the name will be truncated to the 16-character limit.
<i>protocol</i>	A protocol keyword listed in <a href="#">Table 19</a> or <a href="#">Table 20</a> .
<b>alert</b> { <b>on</b>   <b>off</b> }	(Optional) For each inspected protocol, the generation of alert messages can be set be <b>on</b> or <b>off</b> . If no option is selected, alerts are generated on the basis of the setting of the <b>ip inspect alert-off</b> command.
<b>audit-trail</b> { <b>on</b>   <b>off</b> }	(Optional) For each inspected protocol, <b>audit trail</b> can be set <b>on</b> or <b>off</b> . If no option is selected, an audit trail message are generated on the basis of the setting of the <b>ip inspect audit-trail</b> command.
<b>http</b>	Specifies the HTTP protocol for Java applet blocking.
<b>urlfilter</b>	(Optional) Associates URL filtering with HTTP inspection.

<b>timeout</b> <i>seconds</i>	(Optional) To override the global TCP or User Datagram Protocol (UDP), or Internet Control Message Protocol (ICMP) idle timeouts for the specified protocol, specify the number of seconds for a different idle timeout.  This timeout overrides the global TCP, UDP, or ICMP timeouts but will not override the global Domain Name System (DNS) timeout.
<b>java-list</b> <i>access-list</i>	(Optional) Specifies the numbered standard access list to use to determine “friendly” sites. This keyword is available only for the HTTP protocol, for Java applet blocking. Java blocking only works with numbered standard access lists.
<b>rpc program-number</b> <i>number</i>	Specifies the program number to permit. This keyword is available only for the remote-procedure call protocol.
<b>wait-time</b> <i>minutes</i>	(Optional) Specifies the number of minutes to keep a small hole in the firewall to allow subsequent connections from the same source address and to the same destination address and port. The default wait-time is zero minutes. This keyword is available only for the RPC protocol.
<b>fragment</b>	Specifies fragment inspection for the named rule.
<b>max</b> <i>number</i>	(Optional) Specifies the maximum number of unassembled packets for which state information (structures) is allocated by Cisco IOS software. Unassembled packets are packets that arrive at the router interface before the initial packet for a session. The acceptable range is 50 through 10000. The default is 256 state entries.  Memory is allocated for the state structures, and setting this value to a larger number may cause memory resources to be exhausted.
<b>timeout</b> <i>seconds</i> (fragmentation)	(Optional) Configures the number of seconds that a packet state structure remains active. When the timeout value expires, the router drops the unassembled packet, freeing that structure for use by another packet. The default timeout value is one second.  If this number is set to a value greater than one second, it will be automatically adjusted by the Cisco IOS software when the number of free state structures goes below certain thresholds: when the number of free states is less than 32, the timeout will be divided by 2. When the number of free states is less than 16, the timeout will be set to 1 second.

**Table 19 Protocol Keywords—Transport-Layer Protocols**

Protocol	Keyword
ICMP	<b>icmp</b>
TCP	<b>tcp</b>
UDP	<b>udp</b>

**Table 20 Protocol Keywords—Application-Layer Protocols**

Protocol	Keyword
CU-SeeMe	cuseeme
FTP	ftp
Java	http
H.323	h323
Microsoft NetShow	netshow
RealAudio	realaudio
remote-procedure call (RPC)	rpc
Session Initiation Protocol (SIP)	sip
simple mail transfer protocol (SMTP)	smtp
Structured Query Language*Net (SQL*Net)	sqlnet
StreamWorks	streamworks
TFTP	tftp
UNIX R commands (rlogin, rexec, rsh)	rcmd
VDOLive	vdolive

**Defaults**

No inspection rules are defined until you define them using this command.

**Command Modes**

Global configuration

**Command History**

Release	Modification
11.2 P	This command was introduced.
12.0(5)T	Introduced configurable alert and audit trail, IP fragmentation checking, and NetShow protocol support.
12.2(11)YU	Support was added for ICMP and SIP protocols and the <b>urlfilter</b> keyword was added to the HTTP inspection syntax.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Usage Guidelines**

To define a set of inspection rules, enter this command for each protocol that you want the Cisco IOS firewall to inspect, using the same *inspection-name*. Give each set of inspection rules a unique *inspection-name*, which should not exceed the 16-character limit. Define either one or two sets of rules per interface—you can define one set to examine both inbound and outbound traffic, or you can define two sets: one for outbound traffic and one for inbound traffic.

To define a single set of inspection rules, configure inspection for all the desired application-layer protocols, and for TCP, UDP, or ICMP as desired. This combination of TCP, UDP, and application-layer protocols join together to form a single set of inspection rules with a unique name. (There are no application-layer protocols associated with ICMP.)

To remove the inspection rule for a protocol, use the **no** form of this command with the specified inspection name and protocol; to remove the entire set of inspection rules, use the **no** form of this command only; that is, do not list any inspection names or protocols.

In general, when inspection is configured for a protocol, return traffic entering the internal network will be permitted only if the packets are part of a valid, existing session for which state information is being maintained.

### TCP and UDP Inspection

You can configure TCP and UDP inspection to permit TCP and UDP packets to enter the internal network through the firewall, even if the application-layer protocol is not configured to be inspected. However, TCP and UDP inspection do not recognize application-specific commands, and therefore might not permit all return packets for an application, particularly if the return packets have a different port number from the previous exiting packet.

Any application-layer protocol that is inspected will take precedence over the TCP or UDP packet inspection. For example, if inspection is configured for FTP, all control channel information will be recorded in the state table, and all FTP traffic will be permitted back through the firewall if the control channel information is valid for the state of the FTP session. The fact that TCP inspection is configured is irrelevant.

With TCP and UDP inspection, packets entering the network must exactly match an existing session: the entering packets must have the same source or destination addresses and source or destination port numbers as the exiting packet (but reversed). Otherwise, the entering packets will be blocked at the interface.

### ICMP Inspection

An ICMP inspection session is on the basis of the source address of the inside host that originates the ICMP packet. Dynamic access control lists (ACLs) are created for return ICMP packets of the allowed types (echo-reply, time-exceeded, destination unreachable, and timestamp reply) for each session. There are no port numbers associated with an ICMP session, and the permitted IP address of the return packet is wild-carded in the ACL. The wild-card address is because the IP address of the return packet cannot be known in advance for time-exceeded and destination-unreachable replies. These replies can come from intermediate devices rather than the intended destination.

### Application-Layer Protocol Inspection

In general, if you configure inspection for an application-layer protocol, packets for that protocol should be permitted to exit the firewall (by configuring the correct access control list), and packets for that protocol will only be allowed back in through the firewall if they belong to a valid existing session. Each protocol packet is inspected to maintain information about the session state.

Java, H.323, RPC, SIP, and SMTP inspection have additional information, described in the next five sections.

### Java Inspection

Java inspection enables Java applet filtering at the firewall. Java applet filtering distinguishes between trusted and untrusted applets by relying on a list of external sites that you designate as “friendly.” If an applet is from a friendly site, the firewall allows the applet through. If the applet is not from a friendly site, the applet will be blocked. Alternately, you could permit applets from all sites except sites specifically designated as “hostile.”

**Note**

Before you configure Java inspection, you must configure a numbered standard access list that defines “friendly” and “hostile” external sites. You configure this numbered standard access list to permit traffic from friendly sites, and to deny traffic from hostile sites. If you do not configure a numbered standard access list, but use a “placeholder” access list in the **ip inspect name inspection-name http** command, all Java applets will be blocked.

**Note**

Java blocking forces a strict order on TCP packets. To properly verify that Java applets are not in the response, a firewall will drop any TCP packet that is out of order. Because the network—not the firewall—determines how packets are routed, the firewall cannot control the order of the packets; the firewall can only drop and retransmit all TCP packets that are not in order.

**Caution**

CBAC does not detect or block encapsulated Java applets. Therefore, Java applets that are wrapped or encapsulated, such as applets in .zip or .jar format, are *not* blocked at the firewall. CBAC also does not detect or block applets loaded via FTP, gopher, or HTTP on a nonstandard port.

**H.323 Inspection**

If you want CBAC inspection to work with NetMeeting 2.0 traffic (an H.323 application-layer protocol), you must also configure inspection for TCP, as described in the chapter “Configuring Context-Based Access Control” in the *Cisco IOS Security Configuration Guide*. This requirement exists because NetMeeting 2.0 uses an additional TCP channel not defined in the H.323 specification.

**RPC Inspection**

RPC inspection allows the specification of various program numbers. You can define multiple program numbers by creating multiple entries for RPC inspection, each with a different program number. If a program number is specified, all traffic for that program number will be permitted. If a program number is not specified, all traffic for that program number will be blocked. For example, if you created an RPC entry with the NFS program number, all NFS traffic will be allowed through the firewall.

**SIP Inspection**

You can configure SIP inspection to permit media sessions associated with SIP-signaled calls to traverse the firewall. Because SIP is frequently used to signal both incoming and outgoing calls, it is often necessary to configure SIP inspection in both directions on a firewall (both from the protected internal network and from the external network). Because inspection of traffic from the external network is not done with most protocols, it may be necessary to create an additional inspection rule to cause only SIP inspection to be performed on traffic coming from the external network.

**SMTP Inspection**

SMTP inspection causes SMTP commands to be inspected for illegal commands. Any packets with illegal commands are dropped, and the SMTP session will hang and eventually time out. An illegal command is any command except for the following legal commands:

- DATA
- EXPN
- HELO
- HELP
- MAIL

- NOOP
- QUIT
- RCPT
- RSET
- SAML
- SEND
- SOML
- VRFY

#### Use of the `urlfilter` Keyword

If you specify the `urlfilter` keyword, the Cisco IOS firewall will interact with a URL filtering software to control web traffic for a given host or user on the basis of a specified security policy.



#### Note

Enabling HTTP inspection with or without any option triggers the Java applet scanner, which is CPU intensive. The only way to stop the Java applet scanner is to specify the `java-list access-list` option. Configuring URL filtering without enabling the `java-list access-list` option will severely impact performance.

#### Use of the `timeout` Keyword

If you specify a timeout for any of the transport-layer or application-layer protocols, the timeout will override the global idle timeout for the interface to which the set of inspection rules is applied.

If the protocol is TCP or a TCP application-layer protocol, the timeout will override the global TCP idle timeout. If the protocol is UDP or a UDP application-layer protocol, the timeout will override the global UDP idle timeout.

If you do not specify a timeout for a protocol, the timeout value applied to a new session of that protocol will be taken from the corresponding TCP or UDP global timeout value valid at the time of session creation.

The default ICMP timeout is deliberately short (10 seconds) due to the security hole that is opened by allowing ICMP packets with a wild-carded source address back into the inside network. The timeout will occur 10 seconds after the last outgoing packet from the originating host. For example, if you send a set of 10 ping packets spaced one second apart, the timeout will expire in 20 seconds or 10 seconds after the last outgoing packet. However, the timeout is not extended for return packets. If a return packet is not seen within the timeout window, the hole will be closed and the return packet will not be allowed in. Although the default timeout can be made longer if desired, it is recommended that this value be kept relatively short.

#### IP Fragmentation Inspection

CBAC inspection rules can help protect hosts against certain denial-of-service attacks involving fragmented IP packets. Even though the firewall keeps an attacker from making actual connections to a given host, the attacker may still be able to disrupt services provided by that host. This is done by sending many non-initial IP fragments or by sending complete fragmented packets through a router with an ACL that filters the first fragment of a fragmented packet. These fragments can tie up resources on the target host as it tries to reassemble the incomplete packets.

Using fragmentation inspection, the firewall maintains an *interfragment state* (structure) for IP traffic. Non-initial fragments are discarded unless the corresponding initial fragment was permitted to pass through the firewall. Non-initial fragments received before the corresponding initial fragments are discarded.

**Note**

Fragmentation inspection can have undesirable effects in certain cases, because it can result in the firewall discarding any packet whose fragments arrive out of order. There are many circumstances that can cause out-of-order delivery of legitimate fragments. Apply fragmentation inspection in situations where legitimate fragments, which are likely to arrive out of order, might have a severe performance impact.

Because routers running Cisco IOS software are used in a very large variety of networks, and because the CBAC feature is often used to isolate parts of internal networks from one another, the fragmentation inspection feature is not enabled by default. Fragmentation detection must be explicitly enabled for an inspection rule using the **ip inspect name** command. Unfragmented traffic is never discarded because it lacks a fragment state. Even when the system is under heavy attack with fragmented packets, legitimate fragmented traffic, if any, will still get some fraction of the firewall's fragment state resources, and legitimate, unfragmented traffic can flow through the firewall unimpeded.

**Examples**

The following example causes the software to inspect TCP sessions and UDP sessions, and to specifically allow CU-SeeMe, FTP, and RPC traffic back through the firewall for existing sessions only. For UDP traffic, audit-trail is on. For FTP traffic, the idle timeout is set to override the global TCP idle timeout. For RPC traffic, program numbers 100003, 100005, and 100021 are permitted.

```
ip inspect name myrules tcp
ip inspect name myrules udp audit-trail on
ip inspect name myrules cuseeme
ip inspect name myrules ftp timeout 120
ip inspect name myrules rpc program-number 100003
ip inspect name myrules rpc program-number 100005
ip inspect name myrules rpc program-number 100021
```

The following example adds fragment checking to software inspection of TCP and UDP sessions for the rule named "myrules." In this example, the firewall software will allocate 100 state structures, and the timeout value for dropping unassembled packets is set to 4 seconds. If 100 initial fragments for 100 different packets are sent through the router, all of the state structures will be used up. The initial fragment for packet 101 will be dropped. Additionally, if the number of free state structures (structures available for use by unassembled packets) drops below the threshold values, 32 or 16, the timeout value is automatically reduced to 2 or 1, respectively. Changing the timeout value frees up packet state structures more quickly.

```
ip inspect name myrules tcp
ip inspect name myrules udp audit-trail on
ip inspect name myrules cuseeme
ip inspect name myrules ftp timeout 120
ip inspect name myrules rpc program-number 100003
ip inspect name myrules rpc program-number 100005
ip inspect name myrules rpc program-number 100021
ip inspect name myrules fragment max 100 timeout 4
```

The following firewall and SIP example shows how to allow outside-initiated calls and internal calls. For outside-initiated calls, an ACL needs to be punched to allow for the traffic from the initial signaling packet from outside. Subsequent signaling and media channels will be allowed by the inspection module.

```

ip inspect name voip sip
interface FastEthernet0/0
  ip inspect voip in
!
!
interface FastEthernet0/1
  ip inspect voip in
  ip access-group 100 in
!
!
access-list 100 permit udp host <gw ip> any eq 5060
access-list 100 permit udp host <proxy ip> any eq 5060
access-list deny ip any any

```

The following example shows two configured inspections named “fw\_only” and “fw\_urlf”; URL filtering will work only on the traffic that is inspected by fw\_urlf. Note that the **java-list access-list** option has been enabled, which disables java scanning.

```

ip inspect name fw_only http java-list 51 timeout 30
interface e0
  ip inspect fw_only in
!
ip inspect name fw_urlf http urlfilter java-list 51 timeout 30
interface e1
  ip inspect fw_urlf in

```

#### Related Commands

Command	Description
<a href="#">ip inspect</a>	Applies a set of inspection rules to an interface.
<a href="#">ip inspect alert-off</a>	Disables CBAC alert messages.
<a href="#">ip inspect audit-trail</a>	Turns on CBAC audit trail messages, which will be displayed on the console after each CBAC session close.

# ip inspect one-minute high

To define the rate of new unestablished sessions that will cause the software to start deleting half-open sessions, use the **ip inspect one-minute high** command in global configuration mode. To reset the threshold to the default of 500 half-open sessions, use the **no** form of this command.

**ip inspect one-minute high** *number*

**no ip inspect one-minute high**

<b>Syntax Description</b>	<i>number</i>	Specifies the rate of new unestablished TCP sessions that will cause the software to start deleting half-open sessions. The default is 500 half-open sessions.
---------------------------	---------------	--

<b>Defaults</b>	500 half-open sessions
-----------------	------------------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.2 P	This command was introduced.

**Usage Guidelines**

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, “half-open” means that the session has not reached the established state. For User Datagram Protocol, “half-open” means that the firewall has detected traffic from one direction only.

Context-based Access Control (CBAC) measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are included in the total number and rate measurements. Measurements are made once a minute.

When the rate of new connection attempts rises above a threshold (the **one-minute high** number), the software will delete half-open sessions as required to accommodate new connection attempts. The software will continue to delete half-open sessions as necessary, until the rate of new connection attempts drops below another threshold (the **one-minute low** number). The rate thresholds are measured as the number of new session connection attempts detected in the last one-minute sample period. (The rate is calculated as an exponentially-decayed rate.)

The global value specified for this threshold applies to all TCP and UDP connections inspected by CBAC.

**Examples**

The following example causes the software to start deleting half-open sessions when more than 1000 session establishment attempts have been detected in the last minute, and to stop deleting half-open sessions when fewer than 950 session establishment attempts have been detected in the last minute:

```
ip inspect one-minute high 1000
ip inspect one-minute low 950
```

Related Commands	Command	Description
	<a href="#">ip inspect one-minute low</a>	Defines the rate of new unestablished TCP sessions that will cause the software to stop deleting half-open sessions.
	<a href="#">ip inspect max-incomplete high</a>	Defines the number of existing half-open sessions that will cause the software to start deleting half-open sessions.
	<a href="#">ip inspect max-incomplete low</a>	Defines the number of existing half-open sessions that will cause the software to stop deleting half-open sessions.
	<a href="#">ip inspect tcp max-incomplete host</a>	Specifies the threshold and blocking time values for TCP host-specific denial-of-service detection and prevention.

# ip inspect one-minute low

To define the rate of new unestablished TCP sessions that will cause the software to stop deleting half-open sessions, use the **ip inspect one-minute low** command in global configuration mode. To reset the threshold to the default of 400 half-open sessions, use the **no** form of this command

**ip inspect one-minute low** *number*

**no ip inspect one-minute low**

<b>Syntax Description</b>	<i>number</i>	Specifies the rate of new unestablished TCP sessions that will cause the software to stop deleting half-open sessions. The default is 400 half-open sessions.
---------------------------	---------------	---

<b>Defaults</b>	400 half-open sessions
-----------------	------------------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.2 P	This command was introduced.

**Usage Guidelines**

An unusually high number of half-open sessions (either absolute or measured as the arrival rate) could indicate that a denial-of-service attack is occurring. For TCP, “half-open” means that the session has not reached the established state. For User Datagram Protocol, “half-open” means that the firewall has detected traffic from one direction only.

Context-based Access Control (CBAC) measures both the total number of existing half-open sessions and the rate of session establishment attempts. Both TCP and UDP half-open sessions are included in the total number and rate measurements. Measurements are made once a minute.

When the rate of new connection attempts rises above a threshold (the **one-minute high** number), the software will delete half-open sessions as required to accommodate new connection attempts. The software will continue to delete half-open sessions as necessary, until the rate of new connection attempts drops below another threshold (the **one-minute low** number). The rate thresholds are measured as the number of new session connection attempts detected in the last one-minute sample period. (The rate is calculated as an exponentially decayed rate.)

The global value specified for this threshold applies to all TCP and UDP connections inspected by CBAC.

**Examples**

The following example causes the software to start deleting half-open sessions when more than 1000 session establishment attempts have been detected in the last minute, and to stop deleting half-open sessions when fewer than 950 session establishment attempts have been detected in the last minute:

```
ip inspect one-minute high 1000
ip inspect one-minute low 950
```

Related Commands	Command	Description
	<a href="#">ip inspect max-incomplete high</a>	Defines the number of existing half-open sessions that will cause the software to start deleting half-open sessions.
	<a href="#">ip inspect max-incomplete low</a>	Defines the number of existing half-open sessions that will cause the software to stop deleting half-open sessions.
	<a href="#">ip inspect one-minute high</a>	Defines the rate of new unestablished sessions that will cause the software to start deleting half-open sessions.
	<a href="#">ip inspect tcp max-incomplete host</a>	Specifies the threshold and blocking time values for TCP host-specific denial-of-service detection and prevention.

# ip inspect tcp finwait-time

To define how long a TCP session will still be managed after the firewall detects a FIN-exchange, use the **ip inspect tcp finwait-time** command in global configuration mode. To reset the timeout to the default of 5 seconds, use the **no** form of this command.

**ip inspect tcp finwait-time** *seconds*

**no ip inspect tcp finwait-time**

## Syntax Description

<i>seconds</i>	Specifies how long a TCP session will be managed after the firewall detects a FIN-exchange. The default is 5 seconds.
----------------	---

## Defaults

5 seconds

## Command Modes

Global configuration

## Command History

Release	Modification
11.2 P	This command was introduced.

## Usage Guidelines

When the software detects a valid TCP packet that is the first in a session, and if Context-based Access Control (CBAC) inspection is configured for the packet's protocol, the software establishes state information for the new session.

Use this command to define how long TCP session state information will be maintained after the firewall detects a FIN-exchange for the session. The FIN-exchange occurs when the TCP session is ready to close.

The global value specified for this timeout applies to all TCP sessions inspected by CBAC.

The timeout set with this command is referred to as the "finwait" timeout.



### Note

If the **-n** option is used with **rsh**, and the commands being executed do not produce output before the "finwait" timeout, the session will be dropped and no further output will be seen.

## Examples

The following example changes the "finwait" timeout to 10 seconds:

```
ip inspect tcp finwait-time 10
```

The following example changes the "finwait" timeout back to the default (5 seconds):

```
no ip inspect tcp finwait-time
```

# ip inspect tcp idle-time

To specify the TCP idle timeout (the length of time a TCP session will still be managed while there is no activity), use the **ip inspect tcp idle-time** command in global configuration mode. To reset the timeout to the default of 3600 seconds (1 hour), use the **no** form of this command.

**ip inspect tcp idle-time** *seconds*

**no ip inspect tcp idle-time**

## Syntax Description

*seconds* Specifies the length of time, in seconds, for which a TCP session will still be managed while there is no activity. The default is 3600 seconds (1 hour).

## Defaults

3600 seconds (1 hour)

## Command Modes

Global configuration

## Command History

Release	Modification
11.2 P	This command was introduced.

## Usage Guidelines

When the software detects a valid TCP packet that is the first in a session, and if Context-based Access Control (CBAC) inspection is configured for the packet's protocol, the software establishes state information for the new session.

If the software detects no packets for the session for a time period defined by the TCP idle timeout, the software will not continue to manage state information for the session.

The global value specified for this timeout applies to all TCP sessions inspected by CBAC. This global value can be overridden for specific interfaces when you define a set of inspection rules with the **ip inspect name** (global configuration) command.



### Note

This command does not affect any of the currently defined inspection rules that have explicitly defined timeouts. Sessions created based on these rules still inherit the explicitly defined timeout value. If you change the TCP idle timeout with this command, the new timeout will apply to any new inspection rules you define or to any existing inspection rules that do not have an explicitly defined timeout. That is, new sessions based on these rules (having no explicitly defined timeout) will inherit the global timeout value.

## Examples

The following example sets the global TCP idle timeout to 1800 seconds (30 minutes):

```
ip inspect tcp idle-time 1800
```

The following example sets the global TCP idle timeout back to the default of 3600 seconds (one hour):

```
no ip inspect tcp idle-time
```

# ip inspect tcp max-incomplete host

To specify threshold and blocking time values for TCP host-specific denial-of-service detection and prevention, use the **ip inspect tcp max-incomplete host** command in global configuration mode. To reset the threshold and blocking time to the default values, use the **no** form of this command.

**ip inspect tcp max-incomplete host** *number* **block-time** *minutes*

**no ip inspect tcp max-incomplete host**

## Syntax Description

<i>number</i>	Specifies how many half-open TCP sessions with the same host destination address can exist at a time, before the software starts deleting half-open sessions to the host. Use a number from 1 to 250. The default is 50 half-open sessions.
<b>block-time</b>	Specifies blocking of connection initiation to a host.
<i>minutes</i>	Specifies how long the software will continue to delete new connection requests to the host. The default is 0 minutes.

## Defaults

50 half-open sessions and 0 minutes

## Command Modes

Global configuration

## Command History

Release	Modification
11.2 P	This command was introduced.

## Usage Guidelines

An unusually high number of half-open sessions with the same destination host address could indicate that a denial-of-service attack is being launched against the host. For TCP, “half-open” means that the session has not reached the established state.

Whenever the number of half-open sessions with the same destination host address rises above a threshold (the **max-incomplete host** number), the software will delete half-open sessions according to one of the following methods:

- If the **block-time** *minutes* timeout is 0 (the default):  
The software will delete the oldest existing half-open session for the host for every new connection request to the host. This ensures that the number of half-open sessions to a given host will never exceed the threshold.
- If the **block-time** *minutes* timeout is greater than 0:  
The software will delete all existing half-open sessions for the host, and then block all new connection requests to the host. The software will continue to block all new connection requests until the **block-time** expires.

The software also sends syslog messages whenever the **max-incomplete host** number is exceeded and when blocking of connection initiations to a host starts or ends.

The global values specified for the threshold and blocking time apply to all TCP connections inspected by Context-based Access Control (CBAC).

### Examples

The following example changes the **max-incomplete host** number to 40 half-open sessions, and changes the **block-time** timeout to 2 minutes:

```
ip inspect tcp max-incomplete host 40 block-time 2
```

The following example resets the defaults (50 half-open sessions and 0 minutes):

```
no ip inspect tcp max-incomplete host
```

### Related Commands

Command	Description
<a href="#">ip inspect max-incomplete high</a>	Defines the number of existing half-open sessions that will cause the software to start deleting half-open sessions.
<a href="#">ip inspect max-incomplete low</a>	Defines the number of existing half-open sessions that will cause the software to stop deleting half-open sessions.
<a href="#">ip inspect one-minute high</a>	Defines the rate of new unestablished sessions that will cause the software to start deleting half-open sessions.
<a href="#">ip inspect one-minute low</a>	Defines the rate of new unestablished TCP sessions that will cause the software to stop deleting half-open sessions.

# ip inspect tcp synwait-time

To define how long the software will wait for a TCP session to reach the established state before dropping the session, use the **ip inspect tcp synwait-time** command in global configuration mode. To reset the timeout to the default of 30 seconds, use the **no** form of this command.

**ip inspect tcp synwait-time** *seconds*

**no ip inspect tcp synwait-time**

<b>Syntax Description</b>	<i>seconds</i>	Specifies how long, in seconds, the software will wait for a TCP session to reach the established state before dropping the session. The default is 30 seconds.
---------------------------	----------------	---

<b>Defaults</b>	30 seconds
-----------------	------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.2 P	This command was introduced.

**Usage Guidelines** Use this command to define how long Cisco IOS software will wait for a TCP session to reach the established state before dropping the session. The session is considered to have reached the established state after the session's first SYN bit is detected.

The global value specified for this timeout applies to all TCP sessions inspected by Context-based Access Control (CBAC).

**Examples** The following example changes the "synwait" timeout to 20 seconds:

```
ip inspect tcp synwait-time 20
```

The following example changes the "synwait" timeout back to the default (30 seconds):

```
no ip inspect tcp synwait-time
```

# ip inspect udp idle-time

To specify the User Datagram Protocol idle timeout (the length of time for which a UDP “session” will still be managed while there is no activity), use the **ip inspect udp idle-time** command in global configuration model. To reset the timeout to the default of 30 seconds, use the **no** form of this command.

**ip inspect udp idle-time** *seconds*

**no ip inspect udp idle-time**

<b>Syntax Description</b>	<i>seconds</i>	Specifies the length of time a UDP “session” will still be managed while there is no activity. The default is 30 seconds.
---------------------------	----------------	---

<b>Defaults</b>	30 seconds
-----------------	------------

<b>Command Modes</b>	Global configuration
----------------------	----------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	11.2 P	This command was introduced.

**Usage Guidelines**

When the software detects a valid UDP packet, if Context-based Access Control (CBAC) inspection is configured for the packet’s protocol, the software establishes state information for a new UDP “session.” Because UDP is a connectionless service, there are no actual sessions, so the software approximates sessions by examining the information in the packet and determining if the packet is similar to other UDP packets (for example, it has similar source or destination addresses) and if the packet was detected soon after another similar UDP packet.

If the software detects no UDP packets for the UDP session for the a period of time defined by the UDP idle timeout, the software will not continue to manage state information for the session.

The global value specified for this timeout applies to all UDP sessions inspected by CBAC. This global value can be overridden for specific interfaces when you define a set of inspection rules with the **ip inspect name** command.



**Note**

This command does not affect any of the currently defined inspection rules that have explicitly defined timeouts. Sessions created based on these rules still inherit the explicitly defined timeout value. If you change the UDP idle timeout with this command, the new timeout will apply to any new inspection rules you define or to any existing inspection rules that do not have an explicitly defined timeout. That is, new sessions based on these rules (having no explicitly defined timeout) will inherit the global timeout value.

---

**Examples**

The following example sets the global UDP idle timeout to 120 seconds (2 minutes):

```
ip inspect udp idle-time 120
```

The following example sets the global UDP idle timeout back to the default of 30 seconds:

```
no ip inspect udp idle-time
```

# ip urlfilter alert

To enable URL filtering system alert messages, use the **ip urlfilter alert** command in global configuration mode. To disable the system alert, use the **no** form of this command.

**ip urlfilter alert**

**no ip urlfilter alert**

**Syntax Description** This command has no arguments or keywords.

**Defaults** URL filtering messages are enabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Usage Guidelines** Use the **ip urlfilter alert** command to display system messages, such as a server entering allow mode, a server going down, or a URL that is too long for the lookup request.

**Examples** The following example shows how to enable URL filtering alert messages:

```
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor websense 192.168.3.1
```

Afterward, system alert messages such as the following are displayed:

```
%URLF-3-SERVER_DOWN:Connection to the URL filter server 10.92.0.9 is down
```

This level three LOG\_ERR-type message is displayed when a configured URL filter server (UFS) goes down. When this happens, the firewall will mark the configured server as secondary and try to bring up one of the other secondary servers and mark that server as the primary server. If there is no other server configured, the firewall will enter into allow mode and display the URLF-3-ALLOW\_MODE message described.

```
%URLF-3-ALLOW_MODE:Connection to all URL filter servers are down and ALLOW MODE is OFF
```

This LOG\_ERR type message is displayed when all UFSs are down and the system enters into allow mode.

**Note**

Whenever the system goes into allow mode (all filter servers are down), a periodic keepalive timer will be triggered that will try to bring up a server by opening a TCP connection.

```
%URLF-5-SERVER_UP:Connection to an URL filter server 10.92.0.9 is made, the system is returning from ALLOW MODE
```

This LOG\_NOTICE-type message is displayed when the UFSs are detected as being up and the system is returning from allow mode.

```
%URLF-4-URL_TOO_LONG:URL too long (more than 3072 bytes), possibly a fake packet?
```

This LOG\_WARNING-type message is displayed when the URL in a lookup request is too long; any URL longer than 3K will be dropped.

```
%URLF-4-MAX_REQ:The number of pending request exceeds the maximum limit <1000>
```

This LOG\_WARNING-type message is displayed when the number of pending requests in the system exceeds the maximum limit and all further requests are dropped.

# ip urlfilter allowmode

To turn on the default mode (allow mode) of the filtering algorithm, use the **ip urlfilter allowmode** command in global configuration mode. To disable the default mode, use the **no** form of this command.

**ip urlfilter allowmode [on | off]**

**no ip urlfilter allowmode [on | off]**

## Syntax Description

<b>on</b>	(Optional) Allow mode is on.
<b>off</b>	(Optional) Allow mode is off.

## Defaults

Allow mode is off.

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(11)YU	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

## Usage Guidelines

The system will go into allow mode when connections to all vendor servers (Websense or N2H2) are down. The system will return to normal mode when a connection to at least one web vendor server is up. Allow mode directs your system to forward or drop all packets on the basis of the configurable allow mode setting: if allow mode is on and the vendor servers are down, the HTTP requests will be allowed to pass; if allow mode is off and the vendor servers are down, the HTTP requests will be forbidden.

## Examples

The following example shows how to enable allow mode on your system:

```
ip urlfilter allowmode on
```

Afterward, the following alert message will be displayed when the system goes into allow mode:

```
%URLF-3-ALLOW_MODE: Connection to all URL filter servers are down and ALLOW MODE is OFF
```

The following alert message will be displayed when the system returns from allow mode:

```
%URLF-5-SERVER_UP: Connection to an URL filter server 12.0.0.3 is made, the system is returning from allow mode
```

# ip urlfilter audit-trail

To log messages into the syslog server or router, use the **ip urlfilter audit-trail** command in global configuration mode. To disable this functionality, use the **no** form of this command.

**ip urlfilter audit-trail**

**no ip urlfilter audit-trail**

**Syntax Description** This command has no arguments or keywords.

**Defaults** This command is disabled.

**Command Modes** Global configuration

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Usage Guidelines** Use the **ip urlfilter audit-trail** command to log messages such as URL request status (allow or deny) into your syslog server.

**Examples** The following example shows how to enable syslog message logging:

```
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor websense 192.168.3.1
```

Afterward, audit trail messages such as the following are displayed and logged into the log server:

```
%URLF-6-SITE_ALLOWED:Client 11.0.0.2:12543 accessed server 10.76.82.21:8080
```

This message is logged for each request whose destination IP address is found in the cache. It includes the source IP address, source port number, destination IP address, and destination port number. The URL is not logged in this case because the IP address of the request is found in the cache; thus, parsing the request and extracting the URL is a waste of time.

```
%URLF-4-SITE-BLOCKED: Access denied for the site 'www.sports.com'; client  
12.54.192.6:34557 server 64.124.50.12:80
```

This message is logged when a request finds a match against one of the blocked domains in the exclusive-domain list or the corresponding entry in the IP cache.

```
%URLF-6-URL_ALLOWED:Access allowed for URL http://www.N2H2.com/; client 12.54.192.6:54123  
server 192.168.0.1:80
```

This message is logged for each URL request that is allowed by the vendor server (Websense or N2H2). It includes the allowed URL, source IP address, source port number, destination IP address, and destination port number. Longer URLs will be truncated to 300 bytes and then logged.

```
%URLF-6-URL_BLOCKED:Access denied URL http://www.google.com; client 12.54.192.6:54678  
server 64.192.14.2:80
```

This message is logged for each URL request that is blocked by the vendor server. It includes the blocked URL, source IP address, source port number, destination IP address, and destination port number. Longer URLs will be truncated to 300 bytes and then logged.

# ip urlfilter cache

To configure cache parameters, use the **ip urlfilter cache** command in global configuration mode. To clear the configuration, use the **no** form of this command.

**ip urlfilter cache** *number*

**no ip urlfilter cache** *number*

## Syntax Description

<i>number</i>	Maximum number of destination IP addresses that can be cached into the cache table. The default value is 5000.
---------------	--

## Defaults

Maximum number of destination IP addresses is 5000.

The cache table is cleared out every 12 hours.

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(11)YU	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

## Usage Guidelines

The cache table consists of the most recently requested IP addresses and respective authorization status for each IP address.

The caching algorithm involves three parameters—the maximum number of IP addresses that can be cached, an idle time, and an absolute time. The algorithm also involves two timers—idle timer and absolute timer. The idle timer is a small periodic timer (1 minute) that checks to see whether the number of cached IP addresses in the cache table exceeds 80 percent of the maximum limit. If the cached IP addresses have exceeded 80 percent, it will start removing idle entries; if it has not exceeded 80 percent, it will quit and wait for the next cycle. The absolute timer is a large periodic timer (1 hour) that is used to remove all of the elapsed entries. (The age of an elapsed entry is greater than the absolute time.) An elapsed entry will also be removed during cache lookup.

The idle time value is fixed at 10 minutes. The absolute time value is taken from the vendor server look-up response, which is often greater than 15 hours. The absolute value for cache entries made out of exclusive-domains is 12 hours. The maximum number of cache entries is configurable by enabling the **ip urlfilter cache** command.



### Note

The vendor server is not able to inform the Cisco IOS firewall of filtering policy changes in the database.

**Examples**

The following example shows how to configure the cache table to hold a maximum of five destination IP addresses:

```
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor websense 192.168.3.1
```

**Related Commands**

Command	Description
<a href="#">clear ip urlfilter cache</a>	Clears the cache table.
<a href="#">show ip urlfilter cache</a>	Displays the destination IP addresses that are cached into the cache table.

# ip urlfilter exclusive-domain

To add or remove a domain name to or from the exclusive domain list so that the firewall does not have to send lookup requests to the vendor server, use the **ip urlfilter exclusive-domain** command in global configuration mode. To remove a domain name from the exclusive domain name list, use the **no** form of this command.

**ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name*

**no ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name*

## Syntax Description

<b>permit</b>	Permits all traffic destined for the specified domain name.
<b>deny</b>	Blocks all traffic destined for the specified domain name.
<i>domain-name</i>	Domain name that is added or removed from the exclusive domain name list; for example, www.cisco.com.

## Defaults

This command is not enabled.

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(11)YU	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

## Usage Guidelines

The **ip urlfilter exclusive-domain** command allows you to specify a list of domain names (exclusive domains) so that the firewall will not create a lookup request for the HTTP traffic that is destined for one of the domains in the exclusive list. Thus, you can avoid sending look-up requests to the web server for HTTP traffic that is destined for a host that is completely allowed to all users.

Flexibility when entering domain names is also provided; that is, the user can enter the complete domain name or a partial domain name.

### Complete Domain Name

If the user adds a complete domain name, such as “www.cisco.com,” to the exclusive domain list, all HTTP traffic whose URLs are destined for this domain (such as www.cisco.com/news and www.cisco.com/index) will be excluded from the URL filtering policies of the vendor server (Websense or N2H2), and on the basis of the configuration, the URLs will be permitted or blocked (denied).

### Partial Domain Name

If the user adds only a partial domain name to the exclusive domain list, such as “.cisco.com,” all URLs whose domain names end with this partial domain name (such as www.cisco.com/products and www.cisco.com/eng) will be excluded from the URL filtering policies of the vendor server (Websense or N2H2), and on the basis of the configuration, the URLs will be permitted or blocked (denied).

---

**Examples**

The following example shows how to add the complete domain name “www.cisco.com” to the exclusive domain name list. This configuration will block all traffic destined to the www.cisco.com domain.

```
ip urlfilter exclusive-domain deny www.cisco.com
```

The following example shows how to add the partial domain name “.cisco.com” to the exclusive domain name list. This configuration will permit all traffic destined to domains that end with .cisco.com.

```
ip urlfilter exclusive-domain permit .cisco.com
```

# ip urlfilter max-request

To set the maximum number of outstanding requests that can exist at any given time, use the **ip urlfilter max-request** command in global configuration mode. To disable this function, use the **no** form of this command.

**ip urlfilter max-request** *number*

**no ip urlfilter max-request** *number*

## Syntax Description

*number* Maximum number of outstanding requests. The default value is 1000.

## Defaults

Maximum number of requests is 1000.

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(11)YU	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

## Usage Guidelines

If the specified maximum number of outstanding requests is exceeded, new requests will be dropped.



### Note

Allow mode is not considered because it should be used only when servers are down.

## Examples

The following example shows how to configure the maximum number of outstanding requests to 950:

```
ip inspect name url_filter http
ip urlfilter max-request 950
```

## Related Commands

Command	Description
<a href="#">ip inspect name</a>	Defines a set of inspection rules.
<a href="#">ip urlfilter server vendor</a>	Configures a vendor server for URL filtering.

# ip urlfilter max-resp-pak

To configure the maximum number of HTTP responses that the firewall can keep in its packet buffer, use the **ip urlfilter max-resp-pak** command in global configuration mode. To return to the default, use the **no** form of this command.

**ip urlfilter max-resp-pak** *number*

**no ip urlfilter max-resp-pak** *number*

## Syntax Description

<i>number</i>	Maximum number of HTTP responses that can be stored in the packet buffer of the firewall. After the maximum number has been reached, the firewall will drop further responses. The default, and absolute maximum, value is 200.
---------------	---

## Defaults

200 HTTP responses

## Command Modes

Global configuration

## Command History

Release	Modification
12.2(11)YU	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

## Usage Guidelines

When an HTTP request arrives at a Cisco IOS firewall, the firewall forwards the request to the web server while simultaneously sending a URL look-up request to the vendor server (Websense or N2H2). If the vendor server reply arrives before the HTTP response, the firewall will know whether to permit or block the HTTP response; if the HTTP response arrives before the vendor server reply, the firewall will not know whether to allow or block the response, so the firewall will drop the response until it hears from the vendor server. The **ip urlfilter max-resp-pak** command allows you to configure your firewall to store the HTTP responses in a buffer, which allows your firewall to store a maximum of 200 HTTP responses. Each response will remain in the buffer until an allow or deny message is received from the vendor server. If the vendor server reply allows the URL, the firewall will release the HTTP response from the buffer to the end user; if the vendor server reply denies the URL, the firewall will discard the HTTP response from the buffer and close the connection to both ends.

## Examples

The following example shows how to configure your firewall to hold 150 HTTP responses:

```
ip urlfilter max-resp-pak 150
```

# ip urlfilter server vendor

To configure a vendor server for URL filtering, use the **ip urlfilter server vendor** command in global configuration mode. To remove a server from your configuration, use the **no** form of this command.

```
ip urlfilter server vendor { websense | n2h2 } ip-address [port port-number] [timeout seconds]
[retransmit number]
```

```
no ip urlfilter server vendor { websense | n2h2 } ip-address [port port-number] [timeout seconds]
[retransmit number]
```

Syntax Description		
<b>websense</b>		Websense server will be used.
<b>n2h2</b>		N2H2 server will be used.
<i>ip-address</i>		IP address of the vendor server.
<b>port</b> <i>port-number</i>		(Optional) Port number that the vendor server listens on. The default port number is 15868.
<b>timeout</b> <i>seconds</i>		(Optional) Length of time, in seconds, that the Cisco IOS firewall will wait for a response from the vendor server. The default timeout is 5 seconds.
<b>retransmit</b> <i>number</i>		(Optional) Number of times the Cisco IOS firewall will retransmit the request when a response does not arrive for the request. The default value is two times.

**Defaults** A vendor server is not configured.

**Command Modes** Global configuration

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Usage Guidelines** Use the **ip urlfilter server vendor** command to configure a Websense or N2H2 server, which will interact with the Cisco IOS Firewall to filter HTTP requests on the basis of a specified policy—global filtering, user- or group-based filtering, keyword-based filtering, category-based filtering, or customized filtering.

If the firewall has not received a response from the vendor server within the time specified in the **timeout seconds** keyword and argument, the firewall will check the **retransmit number** keyword and argument configured for the vendor server. If the firewall *has not* exceeded the maximum retransmit tries allowed, it will resend the HTTP lookup request. If the firewall *has* exceeded the maximum retransmit tries allowed, it will delete the outstanding request from the queue and check the status of the allow mode value. The firewall will forward the request if the allow mode is on; otherwise, it will drop the request.

### Primary and Secondary Servers

When users configure multiple vendor servers, the firewall will use only one server at a time—the primary server; all other servers are called secondary servers. When the primary server becomes unavailable for any reason, it becomes a secondary server and one of the secondary servers becomes the primary server.

A firewall marks a primary server as down when sending a request to or receiving a response from the server fails. When a primary server goes down, the system will go to the beginning of the configured servers list and try to activate the first server on the list. If the first server on the list is unavailable, it will try the second server on the list; the system will keep trying to activate a server until it is successful or until it reaches the end of the server list. If the system reaches the end of the server list, it will set a flag indicating that all of the servers are down, and it will enter allow mode.

### Examples

The following example shows how to configure the Websense server for URL filtering:

```
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor websense 192.168.3.1
```

### Related Commands

Command	Description
<a href="#">ip urlfilter allowmode</a>	Turns on the default mode (allow mode) of the filtering algorithm.
<a href="#">ip urlfilter max-request</a>	Sets the maximum number of outstanding requests that can exist at any given time.

# ip urlfilter urlf-server-log

To enable the logging of system messages on the URL filtering server, use the **ip urlfilter urlf-server-log** command in global configuration mode. To disable the logging of system messages, use the **no** form of this command.

**ip urlfilter urlf-server-log**

**no ip urlfilter urlf-server-log**

**Syntax Description** This command has no arguments or keywords.

**Defaults** This command is disabled.

**Command Modes** Global configuration

## Command History

Release	Modification
12.2(11)YU	This command was introduced.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

## Usage Guidelines

Use the **ip urlfilter urlf-server-log** command to enable Cisco IOS to send a log request immediately after the URL lookup request. The firewall will not make a URL lookup request if the destination IP address is in the cache, but it will still make a log request to the server. (The log request contains the URL, host name, source IP address, and the destination IP address.) The server records the log request into its own log server so you can view this information as necessary.

## Examples

The following example shows how to enable system message logging on the URL filter server:

```
ip urlfilter urlf-server-log
```

# no ip inspect

To turn off Context-based Access Control (CBAC) completely at a firewall, use the **no ip inspect** command in global configuration mode.

## no ip inspect

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** No default behavior or values.

---

**Command Modes** Global configuration

---

Command History	Release	Modification
	11.2 P	This command was introduced.

---

---

**Usage Guidelines** Turn off CBAC with the **no ip inspect** global configuration command.



**Note**

---

The **no in inspect** command removes all CBAC configuration entries and resets all CBAC global timeouts and thresholds to the defaults. All existing sessions are deleted and their associated access lists are removed.

---

---

**Examples** The following example turns off CBAC at a firewall:

```
no ip inspect
```

# show ip inspect

To view Context-based Access Control (CBAC) configuration and session information, use the **show ip inspect** command in privileged EXEC mode.

**show ip inspect** { *name inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all** }

## Syntax Description

<b>name</b> <i>inspection-name</i>	Displays the configured inspection rule with the name <i>inspection-name</i> .
<b>config</b>	Displays the complete CBAC inspection configuration.
<b>interfaces</b>	Displays interface configuration with respect to applied inspection rules and access lists.
<b>session</b> [ <b>detail</b> ]	Displays existing sessions that are currently being tracked and inspected by CBAC. The optional <b>detail</b> keyword causes additional details about these sessions to be shown.
<b>all</b>	Displays all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.

## Command Modes

Privileged EXEC

## Command History

Release	Modification
11.2 P	This command was introduced.

## Usage Guidelines

Use this command to view the CBAC configuration and session information.

## Examples

The following example shows sample output for the **show ip inspect name myinspectionrule** command, where the inspection rule “myinspectionrule” is configured:

```
Inspection Rule Configuration
Inspection name myinspectionrule
  tcp timeout 3600
  udp timeout 30
  ftp timeout 3600
```

The output shows the protocols that should be inspected by CBAC and the corresponding idle timeouts for each protocol.

The following is sample output for the **show ip inspect config** command:

```
Session audit trail is disabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
  Inspection name myinspectionrule
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
```

The output shows CBAC configuration, including global timeouts, thresholds, and inspection rules.

The following is sample output for the **show ip inspect interfaces** command:

```
Interface Configuration
Interface Ethernet0
  Inbound inspection rule is myinspectionrule
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
  Outgoing inspection rule is not set
  Inbound access list is not set
  Outgoing access list is not set
```

The following is sample output for the **show ip inspect sessions** command:

```
Established Sessions
Session 25A3318 (10.0.0.1:20)=>(10.1.0.1:46068) ftp-data SIS_OPEN
Session 25A6E1C (10.1.0.1:46065)=>(10.0.0.1:21) ftp SIS_OPEN
```

The output shows the source and destination addresses and port numbers (separated by colons), and it indicates that the session is an FTP session.

The following is sample output for the **show ip inspect sessions detail** command:

```
Established Sessions
Session 25A335C (40.0.0.1:20)=>(30.0.0.1:46069) ftp-data SIS_OPEN
  Created 00:00:07, Last heard 00:00:00
  Bytes sent (initiator:responder) [0:3416064] acl created 1
  Inbound access-list 111 applied to interface Ethernet1
Session 25A6E1C (30.0.0.1:46065)=>(40.0.0.1:21) ftp SIS_OPEN
  Created 00:01:34, Last heard 00:00:07
  Bytes sent (initiator:responder) [196:616] acl created 1
  Inbound access-list 111 applied to interface Ethernet1
```

The output includes times, number of bytes sent, and which access list is applied.

The following is sample output for the **show ip inspect all** command:

```
Session audit trail is disabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
  Inspection name all
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
Interface Configuration
Interface Ethernet0
  Inbound inspection rule is all
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
  Outgoing inspection rule is not set
  Inbound access list is not set
  Outgoing access list is not set
Established Sessions
Session 25A6E1C (30.0.0.1:46065)=>(40.0.0.1:21) ftp SIS_OPEN
Session 25A34A0 (40.0.0.1:20)=>(30.0.0.1:46072) ftp-data SIS_OPEN
```

# show ip urlfilter cache

To display the maximum number of entries that can be cached into the cache table and the number of entries and the destination IP addresses that are cached into the cache table, use the **show ip urlfilter cache** command in EXEC mode.

**show ip urlfilter cache**

**Syntax Description** This command has no arguments or keywords.

**Defaults** This command is not enabled.

**Command Modes** EXEC

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Examples** The following example is sample output from the **show ip urlfilter cache** command:

```
Router# show ip urlfilter cache

Maximum number of entries allowed: 5000
Number of entries cached: 5
IP addresses cached ....
 10.64.128.54
 172.28.139.21
 10.76.82.25
 192.168.0.1
 10.0.1.2
```

[Table 21](#) describes the significant fields shown in the display.

**Table 21** *show ip urlfilter cache Field Descriptions*

Field	Description
Maximum number of entries allowed	Maximum number of destination IP addresses that can be cached into the cache table. This parameter can be configured using the <b>ip url filter cache</b> command. (The default is 5000.)
Number of entries cached	Number of entries that have already been cached into the cache table.
IP addresses cached	IP addresses that have already been cached into the cache table.

## ■ show ip urlfilter cache

Related Commands	Command	Description
	<a href="#">clear ip urlfilter cache</a>	Clears the cache table.
	<a href="#">ip urlfilter cache</a>	Configures cache parameters.

# show ip urlfilter config

To display the size of the cache, the maximum number of outstanding requests, the allow mode state, and the list of configured vendor servers, use the **show ip urlfilter config** command in EXEC mode.

## show ip urlfilter config

**Syntax Description** This command has no arguments or keywords.

**Defaults** This command is not enabled.

**Command Modes** EXEC

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Examples** The following example is sample output from the **show ip urlfilter config** command:

```
Router# show ip urlfilter config

URL filter is ENABLED

Primary Websense server configurations
=====
Websense server IP address: 10.0.0.3
Websense server port: 15868
Websense retransmit time out: 5 (seconds)
Websense number of retransmit:2

Secondary Websense server configurations:
=====
None.

Other configurations
=====
Allow mode: OFF
System Alert: ON
Log message on the router: OFF
Log message on URL filter server:ON
Maximum number of cache entries :5000
Cache timeout :12 (hours)
Maximum number of packet buffers:200
Maximum outstanding requests:1000
```

**show ip urlfilter config**

Related Commands	Command	Description
	<a href="#">ip urlfilter allowmode</a>	Turns on the default mode (allow mode) of the filtering algorithm.
	<a href="#">ip urlfilter cache</a>	Configures cache parameters.
	<a href="#">ip urlfilter max-request</a>	Sets the maximum number of outstanding requests that can exist at any given time.
	<a href="#">ip urlfilter server vendor</a>	Configures a vendor server for URL filtering.

# show ip urlfilter statistics

To display URL filtering statistics, use the **show ip urlfilter statistics** command in EXEC mode.

**show ip urlfilter statistics**

**Syntax Description** This command has no arguments or keywords.

**Defaults** This command is not enabled.

**Command Modes** EXEC

Command History	Release	Modification
	12.2(11)YU	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.

**Usage Guidelines** This command shows information, such as the number of requests that are sent to the vendor server (Websense or N2H2), the number of responses received from the vendor server, the number of pending requests in the system, the number of failed requests, and the number of blocked URLs.

**Examples** The following example is sample output from the **show ip urlfilter statistics** command:

```
Router# show ip urlfilter statistics

URL filtering statistics
=====
Current requests count:25
Current packet buffer count(in use):40
Current cache entry count:3100

Maxever request count:526
Maxever packet buffer count:120
Maxever cache entry count:5000

Total requests sent to URL Filter Server: 44765
Total responses received from URL Filter Server: 44550
Total requests allowed: 44320
Total requests blocked: 224
```

[Table 22](#) describes the significant fields shown in the display.

**Table 22** *show ip urlfilter statistics Field Descriptions*

Field	Description
Current requests count <sup>1</sup>	Number of requests that have been sent to the vendor server.
Current packet buffer count (in use) <sup>2</sup>	Number of HTTP responses that are currently in the packet buffer of the firewall.
Current cache entry count <sup>3</sup>	Number of destination IP addresses that have been cached into the cache table.
Maxever request count <sup>1</sup>	Maximum number of requests that have been sent to the vendor server since power on.
Maxever packet buffer count <sup>2</sup>	Maximum number of HTTP responses that have been stored in the packet buffer of the firewall since power on.
Maxever cache entry count <sup>3</sup>	Maximum number of destination IP addresses that have been cached into the cache table since power on.

1. This value can be specified via the **ip urlfilter max-request** command.

2. This value can be specified via the **ip urlfilter max-resp-pak** command.

3. This value can be specified via the **ip urlfilter cache** command.

**Related Commands**

Command	Description
<b>ip urlfilter cache</b>	Configures cache parameters.
<b>ip urlfilter max-request</b>	Sets the maximum number of outstanding requests that can exist at any given time.
<b>ip urlfilter max-resp-pak</b>	Configures the maximum number of HTTP responses that the firewall can keep in its packet buffer.