



Inspection of Router-Generated Traffic

The Inspection of Router-Generated Traffic feature allows Context-Based Access Control (CBAC) to inspect traffic that is originated by or destined to the router on which CBAC is configured. Previously, inspection of Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and H.323 connections initiated by or destined to the router were allowed.

Feature History for Inspection of Router-Generated Traffic

Release	Modification
12.3(14)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Prerequisites for Inspection of Router-Generated Traffic, page 2](#)
- [Restrictions for Inspection of Router-Generated Traffic, page 2](#)
- [Information About Inspection of Router-Generated Traffic, page 2](#)
- [How to Configure Inspection of Router-Generated Traffic, page 3](#)
- [Configuration Examples for Inspection of Router-Generated Traffic, page 9](#)
- [Additional References, page 9](#)
- [Command Reference, page 10](#)
- [Glossary, page 23](#)



Corporate Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Copyright © 2005 Cisco Systems, Inc. All rights reserved.

Prerequisites for Inspection of Router-Generated Traffic

- Configure CBAC.
- Configure Cisco Call Manager Express (CCME) or H.323 Gateway to configure the inspection of H.323 connections to and from the router.

Restrictions for Inspection of Router-Generated Traffic

- Inspection of router-generated traffic is supported only on the following protocols: H.323, TCP, and UDP.
- The Cisco IOS Firewall supports only Version 2 of the H.323 protocol. If CCME or the H.323 Gateway has inspection of H.323 router traffic enabled, enter the following commands so that it is configured to support only Version 2 features:

```
voice service voip
h323
session transport tcp calls-per-connection 1
h245 tunnel disable
h245 caps mode restricted
h225 timeout tcp call-idle value 0
```

Information About Inspection of Router-Generated Traffic

To configure Inspection of Router-Generated Traffic, you need to understand the following concepts:

- [CBAC, page 2](#)
- [Inspection of Router-Generated Traffic Overview, page 3](#)

CBAC

CBAC is a Cisco IOS Firewall set feature that provides network protection by using the following functions:

- [Traffic Filtering](#)
- [Traffic Inspection](#)
- [Alerts and Audit Trails](#)
- [Intrusion Detection](#)

Traffic Filtering

CBAC filters TCP and UDP packets based on application-layer protocol session information. You can configure CBAC to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network you want to protect. CBAC can inspect traffic for sessions that originate from either side of the firewall, and CBAC can be used for intranet, extranet, and Internet perimeters of your network.

Traffic Inspection

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.

Alerts and Audit Trails

CBAC generates real-time alerts and audit trails. Enhanced audit trail features use SYSLOG to track all network transactions; it records time stamps, the source host, the destination host, the ports used, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send SYSLOG error messages to central management consoles upon detecting suspicious activity.

Using CBAC inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for HTTP traffic, you can specify that in the CBAC rule covering HTTP inspection.

Intrusion Detection

CBAC provides a limited amount of intrusion detection to protect against specific Simple Mail Transfer Protocol (SMTP) attacks. With intrusion detection, SYSLOG messages are reviewed and monitored for specific "attack signatures." Certain types of network attacks have specific characteristics, or signatures. When CBAC detects an attack, it resets the offending connections and sends SYSLOG information to the SYSLOG server.

Inspection of Router-Generated Traffic Overview

Inspection of Router-Generated Traffic enhances CBAC's functionality to inspect TCP, UDP, and H.323 connections that have a router or firewall as one of the connection endpoints. This enables CBAC to open pinholes for TCP, UDP, and H.323 control channel connections to and from the router, and to open pinholes for data and media channels negotiated over the H.323 control channels.

Inspection of TCP and UDP channels initiated from the router enables dynamic opening of pinholes on the interface access control list (ACL) to allow return traffic. You do not have to modify the ACL when a TCP connection such as Telnet is made from the router.

Inspection of local H.323 connections enables the deployment of CCME, H.323 gateway, and the Cisco IOS Firewall on the same router. This also simplifies ACL configuration on CCME's interface through which H.323 connections are made. Before this feature, in addition to configuring ACLs to allow H.323 connections on a standard port (for example, port 1720), you had to configure ACLs to allow all dynamically negotiated data and media channels. With this feature you just configure the ACLs to allow H.323 control channels on port 1720. The Cisco IOS Firewall inspects all the traffic on the control channel and opens pinholes to allow dynamically negotiated data and media channels.

To enable Inspection of Router-Generated Traffic, specify the **router-traffic** keyword in the **ip inspect name** command of the appropriate protocol.

How to Configure Inspection of Router-Generated Traffic

This section contains the following procedures:

- [Configuring H.323 Inspection, page 4](#) (required)
- [Configuring CBAC, page 5](#) (required)
- [Verifying the CBAC Configuration, page 7](#) (optional)

Configuring H.323 Inspection

To configure the H.323 protocol, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* {TCP | UDP | H323} [alert {on | off}] [audit-trail {on | off}][router-traffic][timeout *seconds*]
4. **interface** *type slot/port*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ip inspect name <i>inspection-name</i> {TCP UDP H323} [alert {on off}] [audit-trail {on off}][router-traffic][timeout <i>seconds</i>] Example: Router(config)# ip inspect name test H.323 router-traffic	Defines a set of inspection rules.
Step 4	interface <i>type slot/port</i> Example: Router(config)# interface FE 0/0	Configures an interface type.
Step 5	exit Example: Router(config)# exit	Exits global configuration mode.

Configuring CBAC

To configure CBAC, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {deny | permit} *source* [*source-wildcard*] [log]
4. **ip inspect name** *inspection-name* {TCP | UDP | H323} [alert {on | off}] [audit-trail {on | off}][router-traffic][timeout *seconds*]
5. **interface** *type slot/port*
6. **ip inspect** *inspection-name* {in | out}
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	access-list <i>access-list-number</i> {deny permit} <i>source</i> [<i>source-wildcard</i>] [log] Example: Router(config)# access-list 121 permit tcp host 100.168.11.1 any eq 1720	Defines a standard IP access list.
Step 4	ip inspect name <i>inspection-name</i> {TCP UDP H323} [alert {on off}] [audit-trail {on off}][router-traffic][timeout <i>seconds</i>] Example: Router(config)# ip inspect name here H323 router-traffic timeout 180	Defines a set of inspection rules.
Step 5	interface <i>type slot/port</i> Example: Router(config)# Serial0/3/0	Configures an interface type.

	Command or Action	Purpose
Step 6	<pre>ip inspect inspection-name {in out}</pre> <p>Example: Router(config-if)# ip inspect test in</p>	Enables the Cisco IOS Firewall on an interface.
Step 7	<pre>exit</pre> <p>Example: Router(config)# exit</p>	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the CBAC Configuration

To verify the CBAC configuration, perform the following steps.

SUMMARY STEPS

1. **show ip inspect name** *inspection-name*
2. **show ip inspect config**
3. **show ip inspect interfaces**
4. **show ip inspect session [detail]**
5. **show ip inspect all**

DETAILED STEPS

Step 1 **show ip inspect name** *inspection-name*

Use this command to show a particular configured inspection rule. The following example configures the inspection rule `myinspectionrule`. The output shows the protocols that should be inspected by CBAC and the corresponding idle timeouts for each protocol.

```
Router# show ip inspect name myinspectionrule
```

```
Inspection Rule Configuration
```

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

Step 2 **show ip inspect config**

Use this command to show the CBAC configuration, including global timeouts, thresholds, and inspection rules.

```
Router# show ip inspect config
```

```
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
```

Step 3 show ip inspect interfaces

Use this command to show the interface configuration with respect to applied inspection rules and access lists.

```
Router# show ip inspect interfaces

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
```

Step 4 show ip inspect session detail

Use this command to display existing sessions that CBAC is currently tracking and inspecting. The following sample output shows that an outgoing ACL and an inbound ACL (dynamic ACLs) have been created to allow return traffic.

```
Router# show ip inspect session detail

Established Sessions
Session 80E87274 (192.168.1.116:32956)=>(192.168.101.115:23) tcp SIS_OPEN
  Created 00:00:08, Last heard 00:00:04
  Bytes sent (initiator:responder) [140:298] acl created 2
  Outgoing access-list 102 applied to interface FastEthernet0/0
  Inbound access-list 101 applied to interface FastEthernet0/1
```

Step 5 show ip inspect all

Use this command to show all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.

```
Router# show ip inspect all

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
```

Configuration Examples for Inspection of Router-Generated Traffic

This section provides the following configuration examples:

- [Configuring CBAC with Inspection of H.323 Traffic: Example, page 9](#)

Configuring CBAC with Inspection of H.323 Traffic: Example

These commands create the ACL. In this example, TCP traffic from subnet 100.168.11.1, 192.168.11.50, and 192.168.100.1 is permitted.

```
access-list 120 permit tcp host 100.168.11.1 any eq 1720
access-list 121 permit tcp host 192.168.11.50 host 100.168.11.1 eq 1720
access-list 121 permit tcp host 192.168.100.1 host 100.168.11.1 eq 1720
```

These commands create the CBAC inspection rule LOCAL-H323, allowing inspection of the protocol traffic specified by the rule. This inspection rule sets the timeout value to 180 seconds for each protocol (except for RPC). The timeout value defines the maximum time that a connection for a given protocol can remain active without any traffic passing through the router. When these timeouts are reached, the dynamic ACLs that are inserted to permit the returning traffic are removed, and subsequent packets (possibly even valid ones) are not permitted.

```
ip inspect name LOCAL-H323 tftp timeout 180
ip inspect name LOCAL-H323 h323 router-traffic timeout 180
```

These commands apply the inspection rule and ACL. In this example, the inspection rule LOCAL-H323 is applied to traffic at interface Serial0/3/0.

```
interface Serial0/3/0
 ip address 11.168.11.2 255.255.255.0
 ip access-group 121 in
 ip access-group 120 out
 ip inspect LOCAL-H323 in
 ip inspect LOCAL-H323 out
 encapsulation frame-relay
 frame-relay map ip 11.168.11.1 168 broadcast
 no frame-relay inverse-arp
 frame-relay intf-type dce
```

Additional References

The following sections provide references related to Inspection of Router-Generated Traffic.

Related Documents

Related Topic	Document Title
CBAC	<i>Cisco IOS Security Configuration Guide</i> , Release 12.3
	<i>Cisco IOS Security Command Reference</i> , Release 12.3T
H.323	<i>Cisco IOS H.323 Configuration Guide</i> , Release 12.3

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.3 command reference publications.

- [ip inspect name](#)

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 [parameter max-sessions number] protocol [alert {on | off}]
[audit-trail {on | off}] [timeout seconds]
```

```
no ip inspect name inspection-name [parameter max-sessions number] protocol [alert {on | off}]
[audit-trail {on | off}] [timeout seconds]
```

HTTP Inspection Syntax

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

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

SMTP and ESMTP Inspection Syntax

```
ip inspect name inspection-name {smtp | esmtip} [alert {on | off}] [audit-trail {on | off}]
[max-data number] [timeout seconds]
```

remote-procedure call (RPC) Inspection Syntax

```
ip inspect name inspection-name [parameter max-sessions number] rpc program-number
number [wait-time minutes] [alert {on | off}] [audit-trail {on | off}] [timeout seconds]
```

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

POP3/IMAP Inspection Syntax

```
ip inspect name inspection-name imap [alert {on | off}] [audit-trail {on | off}] [reset]
[secure-login] [timeout number]
```

```
ip inspect name inspection-name pop3 [alert {on | off}] [audit-trail {on | off}] [reset]
[secure-login] [timeout number]
```

Fragment Inspection Syntax

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

```
no ip inspect name inspection-name [parameter max-sessions number] fragment [max number
timeout seconds]
```

Application Firewall Provisioning Syntax

```
ip inspect name inspection-name [parameter max-sessions number] appfw policy-name
```

```
no ip inspect name inspection-name [parameter max-sessions number] appfw policy-name
```

User-Defined Application Syntax

ip inspect name *inspection-name user-10* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]

no ip inspect name *inspection-name user-10* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]

Session Limiting Syntax

no ip inspect name *inspection-name* [**parameter max-sessions** *number*]

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. Note The <i>inspection-name</i> cannot exceed 16 characters; otherwise, the name will be truncated to the 16-character limit.
parameter max-sessions <i>number</i>	(Optional) Limits the number of established firewall sessions that a firewall rule creates. The default is that there is no limit to the number of firewall sessions.
<i>protocol</i>	A protocol keyword listed in Table 1 or Table 2 .
alert { on off }	(Optional) For each inspected protocol, the generation of alert messages can be set be on or off . If no option is selected, alerts are generated on the basis of the setting of the ip inspect alert-off command.
audit-trail { on off }	(Optional) For each inspected protocol, audit trail can be set on or off . If no option is selected, an audit trail message are generated on the basis of the setting of the ip inspect audit-trail command.
timeout <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.
http	Specifies the HTTP protocol for Java applet blocking.
urlfilter	(Optional) Associates URL filtering with HTTP inspection.
java-list <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.
smtp esmtplib	Specifies the protocol being used to inspect the traffic.
max-data <i>number</i>	(Optional) Specifies the maximum number of bytes (data) that can be transferred in a single Simple Mail Transport Protocol (SMTP) session. After the maximum value is exceeded, the firewall logs an alert message and closes the session. Default value: 20 MB
rpc program-number <i>number</i>	Specifies the program number to permit. This keyword is available only for the remote-procedure call protocol.

wait-time <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 remote-procedure call (RPC) protocol.
reset	(Optional) Resets the TCP connection if the client enters a non-protocol command before authentication is complete.
secure-login	(Optional) Causes a user at a non-secure location to use encryption for authentication.
imap	Specifies that the Internet Message Access Protocol (IMAP) is being used.
pop3	Specifies that the Post Office Protocol, Version 3 (POP3) is being used.
fragment	Specifies fragment inspection for the named rule.
max <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.
timeout <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 1 second. If this number is set to a value greater than 1 second, it is 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 fewer than 32, the timeout is divided by 2. When the number of free states is fewer than 16, the timeout is set to 1 second.
appfw	Specifies application firewall provisioning.
<i>policy-name</i>	Application firewall policy name. Note This name must match the name specified via the appfw policy-name command.
<i>appname</i>	Specifies a user- or a system-defined application; for example, user-payroll-sap and user-sametime . Application names can contain hyphens and underscores; however, a user-defined application must have the prefix user- in its title.
port	Specifies the port range for an application.
tcp udp	Specifies the protocol being used to inspect the traffic.
from <i>begin_port_num to end_port_num</i> <i>port_num1 ...</i>	Specifies the starting and ending port numbers or a range of ports from 1 to 5. You must use the from and to keywords together.
list <i>acl_list_num</i>	(Optional) Specifies an access control list number. Only standard ACLs are supported.
description <i>description_string</i>	(Optional) Specifies a description of up to 40 characters.

<i>user-10</i>	Represents a user-defined application in the port-to-application mapping (PAM) table of the ip port-map command.
router-traffic	(Optional) Enables inspection of traffic destined to or originated from a router. Applicable only for H.323, TCP, and UDP protocols. For the command format, see the Note after Table 1 .

Defaults

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

no ip inspect-name protocol removes the inspection rule for the specified protocol.

no ip inspect name removes the entire set of inspection rules.

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 urlfilter keyword was added to the HTTP inspection syntax.
12.2(15)T	Support was added for ICMP, SIP protocols, and the urlfilter keyword was integrated into Cisco IOS Release 12.2(15)T.
12.3(1)	Skinny protocol support was added.
12.3(7)T	Extended Simple Mail Transfer Protocol (ESMTP) protocol support was added.
12.3(14)T	The appfw keyword and the <i>policy-name</i> argument were added to support application firewall provisioning. The parameter max-sessions , secure-login , reset , and router-traffic keywords were added. Support for a larger list of protocols including user-defined applications was added.

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 ICMP, TCP, and UDP, or 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.

Table 1 Protocol Keywords—Transport-Layer and Network-Layer Protocols

Protocol	Keyword
ICMP	icmp
TCP	tcp
UDP	udp

Note The TCP, UDP, and H.323 protocols support the **router-traffic** keyword, which enables inspection of traffic destined to or originated from a router. The command format is as follows:

```
ip inspect name inspection-name { TCP | UDP | H323 } [alert { on | off }] [audit-trail { on | off }][router-traffic][timeout seconds]
```

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.

Granular protocol inspection allows you to specify TCP or UDP ports by using the PAM table. This eliminates having to inspect all applications running under TCP or UDP and the need for multiple access control lists (ACLs) to filter the traffic.

Using the PAM table, you simply pick an existing application or define a new one for inspection thereby simplifying ACL configuration.

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 wildcard 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. [Table 2](#) lists the supported application-layer protocols.

Table 2 Protocol Keywords—Application-Layer Protocols

Protocol	Keyword
Application Firewall	appfw
CU-SeeMe	cuseeme
ESMTP	smtp
FTP	ftp
IMAP	imap
Java	http
H.323	h323
Microsoft NetShow	netshow
POP3	pop3
RealAudio	realaudio
RPC	rpc
SIP	sip
Simple Mail Transfer Protocol (SMTP)	smtp
Skinny Client Control Protocol (SCCP)	skinny
StreamWorks	streamworks
Structured Query Language*Net (SQL*Net)	sqlnet
TFTP	tftp
UNIX R commands (rlogin, rexec, rsh)	rcmd
VDOLive	vdolive
WORD	user-defined application name; use prefix -user
	Note All applications that appear under the show ip port-map command are supported.

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**

Context-Based Access Control (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. Packets with illegal commands are modified to a “xxxx” pattern and forwarded to the server. This process causes the server to send a negative reply, forcing the client to issue a valid command. An illegal SMTP command is any command except the following:

- DATA
- HELO
- HELP

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

ESMTP Inspection

Like SMTP, ESMTP inspection also causes the commands to be inspected for illegal commands. Packets with illegal commands are modified to a “xxxx” pattern and forwarded to the server. This process causes the server to send a negative reply, forcing the client to issue a valid command. An illegal ESMTP command is any command except the following:

- AUTH
- DATA
- EHLO
- ETRN
- HELO
- HELP
- MAIL
- NOOP
- QUIT
- RCPT
- RSET
- SAML
- SEND
- SOML
- VRFY

In addition to inspecting commands, the ESMTP firewall also inspects the following extensions via deeper command inspection:

- Message Size Declaration (SIZE)
- Remote Queue Processing Declaration (ETRN)
- Binary MIME (BINARYMIME)
- Command Pipelining
- Authentication
- Delivery Status Notification (DSN)

- Enhanced Status Code (ENHANCEDSTATUSCODE)
- 8bit-MIMEtransport (8BITMIME)



Note

SMTP and ESMTP cannot exist simultaneously. An attempt to configure both protocols will result in an error message.

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 noninitial 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. Noninitial fragments are discarded unless the corresponding initial fragment was permitted to pass through the firewall. Noninitial 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.

Application Firewall Provisioning

Application firewall provisioning allows you to configure your Cisco IOS Firewall to detect and prohibit a specific protocol type of traffic.

Most firewalls provide only packet filtering capabilities that simply permit or deny traffic without inspecting the data stream; the Cisco IOS application firewall can detect whether or not a packet is in compliance with given HTTP protocol. If the packet is determined to be unauthorized, it will be dropped, the connection will be reset, and a syslog message will be generated, as appropriate.

User-Defined Applications

You can define your own applications and enter them into the port-to-application mapping (PAM) table using the **ip port-map** command. Then you set up your inspection rules by inserting your user-defined application as a value for the *protocol* argument in the **ip inspect name** command.

Session Limiting

Users can limit the number of established firewall sessions that a firewall rule creates by setting the "max-sessions" threshold. A session counter is maintained for each firewall interface. When a session count exceeds the specified threshold, an alert FW-4-SESSION_THRESHOLD_EXCEEDED message is logged to the syslog server and no new sessions can be created.

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
```

The following example shows how to define the HTTP application firewall policy mypolicy. This policy includes all supported HTTP policy rules. This example also includes sample output from the **show appfw configuration** and **show ip inspect config** commands, which allow you to verify the configured setting for the application policy.

```
! Define the HTTP policy.
appfw policy-name mypolicy
 application http
  strict-http action allow alarm
  content-length maximum 1 action allow alarm
  content-type-verification match-req-rsp action allow alarm
  max-header-length request 1 response 1 action allow alarm
  max-uri-length 1 action allow alarm
  port-misuse default action allow alarm
```

```

request-method rfc default action allow alarm
request-method extension default action allow alarm
transfer-encoding type default action allow alarm
!
!
! Apply the policy to an inspection rule.
ip inspect name firewall appfw mypolicy
ip inspect name firewall http
!
!
! Apply the inspection rule to all HTTP traffic entering the FastEthernet0/0 interface.
interface FastEthernet0/0
 ip inspect firewall in
!
!
! Issue the show appfw configuration command and the show ip inspect config command after
the inspection rule "mypolicy" is applied to all incoming HTTP traffic on the
FastEthernet0/0 interface.
!
Router# show appfw configuration

Application Firewall Rule configuration
Application Policy name mypolicy
  Application http
    strict-http action allow alarm
    content-length minimum 0 maximum 1 action allow alarm
    content-type-verification match-req-rsp action allow alarm
    max-header-length request length 1 response length 1 action allow alarm
    max-uri-length 1 action allow alarm
    port-misuse default action allow alarm
    request-method rfc default action allow alarm
    request-method extension default action allow alarm
    transfer-encoding default action allow alarm

Router# show ip inspect config

Session audit trail is disabled
Session alert is enabled
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 firewall
http alert is on audit-trail is off timeout 3600

```

Related Commands

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

Glossary

CBAC—Context-Based Access Control. Scrutinizes source and destination addresses to enhance security for TCP and UDP applications that use well-known ports, such as FTP and e-mail traffic.

firewall—One or more router or access servers designated as a buffer between any connected public networks and a private network. A firewall router uses access lists and other methods to ensure the security of the private network.

FTP—File Transfer Protocol. An application protocol, part of the TCP/IP protocol stack, for transferring files between network nodes.

H.323—A multimedia conferencing protocol that includes voice, video, and data conferencing for use over packet-switched networks. H.323 allows dissimilar communication devices to communicate with each other by using a standardized communication protocol.

IMAP—Internet Message Access Protocol. A method of accessing e-mail or bulletin board messages kept on a mail server that can be shared. IMAP permits client e-mail applications to access remote message stores as if they were local without actually transferring the message.

IP—Internet Protocol. Connectionless protocol at the network layer (Layer 3) of the OSI reference model. Provides features for addressing, type-of-service specification, fragmentation and reassemble, and security. IP works with TCP and is usually identified as TCP/IP.

POP—Post Office Protocol. A protocol that client e-mail applications use to retrieve mail from a mail server.

SMTP—Simple Mail Transfer Protocol. A simple ASCII protocol that describes the exchange of e-mail between two message-transfer agents using TCP/IP.

TCP—Transmission Control Protocol. A connection-oriented transport-layer protocol that provides reliable full-duplex data transmissions.

TCP/IP—Transmission Control Protocol/Internet Protocol. Common name for the suite of protocols developed by the U.S. Department of Defense in the 1970s to support the construction of worldwide internetworks. TCP and IP are the two best-known protocols in the suite.

UDP—User Datagram Protocol. A connectionless transport-layer protocol for exchanging datagrams without acknowledgments or guaranteed delivery.

VoIP—Voice over IP. Capability of carrying normal telephony-style voice over an IP network with circuit-based telephone-like functionality, reliability, and voice quality. VoIP generally refers to the Cisco standards-based (H.323 and so forth) approach to IP voice traffic.

**Note**

Refer to [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

Copyright © 2005 Cisco Systems, Inc. All rights reserved.