



HTTP Inspection Engine

The HTTP Inspection Engine feature allows users to configure their Cisco IOS Firewall to detect and prohibit HTTP connections—such as tunneling over port 80, unauthorized request methods, and non-HTTP compliant file transfers—that are not authorized within the scope of the security policy configuration. Tunneling unauthorized protocols through port 80 and over HTTP exposes a network to significant security risks.

The Cisco IOS Firewall can now be configured with a security policy that adheres to the following tasks:

- Allowing specific traffic targeted for port 80 to traverse the firewall. The traffic is inspected for protocol conformance and for the types of HTTP commands that are allowed or disallowed.
- Denying specific traffic targeted for port 80 that does not comply to HTTP traffic standards. The firewall is enabled to drop the packet, reset the connection, and send a syslog message, as appropriate.

Feature History for HTTP Inspection Engine

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

- [Restrictions for HTTP Inspection Engine, page 2](#)
- [Information About HTTP Inspection Engine, page 2](#)
- [How to Define and Apply an HTTP Application Policy to a Firewall for Inspection, page 2](#)
- [Configuration Examples for Setting Up an HTTP Inspection Engine, page 9](#)
- [Additional References, page 10](#)
- [Command Reference, page 11](#)



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

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

Restrictions for HTTP Inspection Engine

The Cisco 831 router with 48M RAM does not have enough memory to support this feature.

Information About HTTP Inspection Engine

Before configuring an application firewall to detect and police specific traffic targeted for port 80, you should understand the following concepts:

- [What Is a Security Policy?, page 2](#)
- [Cisco IOS HTTP Application Policy Overview, page 2](#)

What Is a Security Policy?

The application firewall uses a security policy, which consists of a collection of static signatures, to detect security violations. A static signature is a collection of parameters that specify protocol conditions that must be met before an action is taken. (For example, a signature may specify that an HTTP data stream containing the POST method must reset the connection.) These protocol conditions and reactions are defined by the end user via the command-line interface (CLI) to form a security policy.

Cisco IOS HTTP Application Policy Overview

HTTP uses port 80 to transport Internet web services, which are commonly used on the network and rarely challenged with regards to their legitimacy and conformance to standards. Because port 80 traffic is typically allowed through the network without being challenged, many application developers are leveraging HTTP traffic as an alternative transport protocol in which to enable their application to travel through or even bypass the firewall.

Most firewalls provide only packet filtering capabilities that simply permit or deny port 80 traffic without inspecting the data stream; the Cisco IOS application firewall for HTTP performs packet inspection as follows:

- Detects HTTP connections that are not authorized within the scope of the security policy configuration.
- Detects users who are tunneling applications through port 80.

If the packet is not in compliance with the HTTP protocol, it will be dropped, the connection will be reset, and a syslog message will be generated, as appropriate.

How to Define and Apply an HTTP Application Policy to a Firewall for Inspection

This section contains the following procedures:

- [Defining an HTTP Application Policy, page 3](#)
- [Applying an HTTP Application Policy to a Firewall for Inspection, page 6](#)

Defining an HTTP Application Policy

Use this task to create an HTTP application firewall policy.

Restrictions

Although application firewall policies are defined in global configuration mode, only one global policy for a given protocol is allowed per interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appfw policy-name** *policy-name*
4. **application** *protocol*
5. **strict-http action** {reset | allow} [alarm]
6. **content-length** {min *bytes* max *bytes* | min *bytes* | max *bytes*} **action** {reset | allow} [alarm]
7. **content-type-verification** [match-req-resp] **action** {reset | allow} [alarm]
8. **max-header-length** request *bytes* response *bytes* **action** {reset | allow} [alarm]
9. **max-uri-length** *bytes* **action** {reset | allow} [alarm]
10. **request-method** {rfc *rfc-method* | extension *extension-method*} **action** {reset | allow} [alarm]
11. **port-misuse** {p2p | tunneling | im | default} **action** {reset | allow} [alarm]
12. **transfer-encoding type** {chunked | compress | deflate | gzip | identity | default} **action** {reset | allow} [alarm]
13. **timeout** *seconds*
14. **audit-trail** {on | off}
15. **exit**
16. **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.

	Command or Action	Purpose
Step 3	<p>appfw policy-name <i>policy-name</i></p> <p>Example: Router(config)# appfw policy-name mypolicy</p>	<p>Defines an application firewall policy and puts the router in application firewall policy configuration mode.</p>
Step 4	<p>application <i>protocol</i></p> <p>Example: Router(cfg-appfw-policy)# application http</p>	<p>Allows you to configure inspection parameters for a given protocol. Currently, only HTTP traffic can be inspected.</p> <ul style="list-style-type: none"> <i>protocol</i> —Specify the http keyword. <p>This command puts you in <i>appfw-policy-protocol</i> configuration mode, where “<i>protocol</i>” is dependent upon the specified protocol. Because only HTTP can be specified, the configuration mode is <i>appfw-policy-http</i>.</p>
Step 5	<p>strict-http action {reset allow} [alarm]</p> <p>Example: Router(cfg-appfw-policy-http)# strict-http action allow alarm</p>	<p>(Optional) Allows HTTP messages to pass through the firewall or resets the TCP connection when HTTP noncompliant traffic is detected.</p>
Step 6	<p>content-length {min bytes max bytes min bytes max bytes} action {reset allow} [alarm]</p> <p>Example: Router(cfg-appfw-policy-http)# content-length max 1 action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic through the firewall on the basis of message size.</p> <ul style="list-style-type: none"> min max bytes—Minimum or maximum content length, in bytes, allowed per message. Number of bytes range: 0 to 65535.
Step 7	<p>content-type-verification [match-req-resp] action {reset allow} [alarm]</p> <p>Example: Router(cfg-appfw-policy-http)# content-type-verification match-req-resp action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic through the firewall on the basis of content message type.</p>
Step 8	<p>max-header-length request bytes response bytes action {reset allow} [alarm]</p> <p>Example: Router(cfg-appfw-policy-http)# max-header-length request 1 response 1 action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic on the basis of the message header length.</p> <ul style="list-style-type: none"> <i>bytes</i>—Number of bytes ranging from 0 to 65535.
Step 9	<p>max-uri-length bytes action {reset allow} [alarm]</p> <p>Example: Router(cfg-appfw-policy-http)# max-uri-length 1 action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic on the basis of the URI length in the request message.</p>

Command or Action	Purpose
<p>Step 10</p> <pre>request method {rfc rfc-method extension extension-method} action {reset allow} [alarm]</pre> <p>Example: Router(cfg-appfw-policy-http)# request-method rfc default action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic according to either the request methods or the extension methods.</p> <ul style="list-style-type: none"> • rfc—Specifies that the supported methods of RFC 2616, <i>Hypertext Transfer Protocol—HTTP/1.1</i>, are to be used for traffic inspection. • rfc-method—Any one of the following RFC 2616 methods can be specified: connect, default, delete, get, head, options, post, put, trace. • extension—Specifies that the extension methods are to be used for traffic inspection. • extension-method—Any one of the following extension methods can be specified: copy, default, edit, getattribute, getproperties, index, lock, mkdir, move, revadd, rexlabel, revlog, save, setattribute, startrev, stoprev, unedit, unlock.
<p>Step 11</p> <pre>port-misuse {p2p tunneling im default} action {reset allow} [alarm]</pre> <p>Example: Router(cfg-appfw-policy-http)# port-misuse default action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic through the firewall on the basis of specified applications in the HTTP message.</p> <ul style="list-style-type: none"> • p2p—Peer-to-peer protocol applications subject to inspection: Kazaa and Gnutella. • tunneling—Tunneling applications subject to inspection: HTTPPort/HTTPHost, GNU Httptunnel, GotoMyPC, Firethru, Http-tunnel.com Client • im—Instant messaging protocol applications subject to inspection: Yahoo Messenger. • default—All applications are subject to inspection.
<p>Step 12</p> <pre>transfer-encoding type {chunked compress deflate gzip identity default} action {reset allow} [alarm]</pre> <p>Example: Router(cfg-appfw-policy-http)# transfer-encoding type default action allow alarm</p>	<p>(Optional) Permits or denies HTTP traffic according to the specified transfer-encoding of the message.</p> <ul style="list-style-type: none"> • chunked—Encoding format (specified in RFC 2616, <i>Hypertext Transfer Protocol—HTTP/1</i>) in which the body of the message is transferred in a series of chunks; each chunk contains its own size indicator. • compress—Encoding format produced by the UNIX “compress” utility. • deflate—“ZLIB” format defined in RFC 1950, <i>ZLIB Compressed Data Format Specification version 3.3</i>, combined with the “deflate” compression mechanism described in RFC 1951, <i>DEFLATE Compressed Data Format Specification version 1.3</i>. • gzip—Encoding format produced by the “gzip” (GNU zip) program. • identity—Default encoding, which indicates that no encoding has been performed. • default—All of the transfer encoding types.

	Command or Action	Purpose
Step 13	timeout <i>seconds</i> Example: Router(cfg-appfw-policy-http)# timeout 60	(Optional) Overrides the global TCP idle timeout value for HTTP traffic. Note If this command is not issued, the default value specified via the ip inspect tcp idle-time command will be used.
Step 14	audit-trail {on off} Example: Router(cfg-appfw-policy-http)# audit-trail on	(Optional) Turns audit trail messages on or off. Note If this command is not issued, the default value specified via the ip inspect audit-trail command will be used.
Step 15	exit Example: Router(cfg-appfw-policy-http)# exit	Exits cfg-appfw-policy-http configuration mode.
Step 16	exit Example: Router(cfg-appfw-policy)# exit	Exits cfg-appfw-policy configuration mode.

What to Do Next

After you have successfully defined an application policy for HTTP traffic inspection, you must apply the policy to an inspection rule. Thereafter, the inspection rule must be applied to an interface. For information on completing this task, see the section [“Applying an HTTP Application Policy to a Firewall for Inspection.”](#)

Applying an HTTP Application Policy to a Firewall for Inspection

Use this task to apply an HTTP application policy to an inspection rule, followed by applying the inspection rule to an interface.



Note

An application policy can coexist with other inspection protocols (for example, an HTTP policy and an FTP policy can coexist).

Prerequisites

You must have already defined an application policy (as shown in the section [“Defining an HTTP Application Policy”](#)).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **appfw** *policy-name*
4. **ip inspect name** *inspection-name* **http** [**alert** {on | off}] [**audit-trail** {on | off}] [**timeout** *seconds*]

5. **interface** *type number*
6. **ip inspect** *inspection-name* {**in** | **out**}
7. **exit**
8. **exit**
9. **show appfw configuration** [*name*]
or
show ip inspect {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **statistics** | **all**}

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	ip inspect name <i>inspection-name</i> appfw <i>policy-name</i> Example: Router(config)# ip inspect name firewall appfw mypolicy	Defines a set of inspection rules for the application policy. <ul style="list-style-type: none"><i>policy-name</i>—Must match the policy name specified via the appfw policy-name command.
Step 4	ip inspect name <i>inspection-name</i> http [alert { on off }] [audit-trail { on off }] [timeout <i>seconds</i>] Example: Router(config)# ip inspect name firewall http	Defines a set of inspection rules that is to be applied to all HTTP traffic. <ul style="list-style-type: none">The <i>inspection-name</i> argument must match the <i>inspection-name</i> argument specified in Step 3.
Step 5	interface <i>type number</i> Example: Router#(config)# interface FastEthernet0/0	Configures an interface type and enters interface configuration mode.
Step 6	ip inspect <i>inspection-name</i> { in out }	Applies the inspection rules (defined in Step 3 and Step 4) to all traffic entering the specified interface. <ul style="list-style-type: none">The <i>inspection-name</i> argument must match the inspection name defined via the ip inspect name command.
Step 7	exit Example: Router#(config-if)# exit	Exits interface configuration mode.

	Command or Action	Purpose
Step 8	<p>exit</p> <p>Example: Router(config)# exit</p>	Exits global configuration mode.
Step 9	<p>show appfw configuration [name]</p> <p>Example: Router# show appfw configuration</p> <p>or</p> <p>show ip inspect {name inspection-name config interfaces session [detail] statistics all}</p> <p>Example: Router# show ip inspect config</p>	<p>(Optional) Displays application firewall policy configuration information.</p> <p>(Optional) Displays firewall-related configuration information.</p>

Troubleshooting Tips

To help troubleshoot the application firewall configuration, issue the following application-firewall specific debug command: **debug appfw {application protocol | function-trace | object-creation | object-deletion | events | timers | detailed}**.

The following sample configuration shows how to configure an HTTP policy with application firewall debugging enabled:

```
Router(config)# appfw policy-name myPolicyAPPFW FUNC:appfw_policy_find
APPFW FUNC:appfw_policy_find -- Policy myPolicy is not found
APPFW FUNC:appfw_policy_alloc
APPFW FUNC:appfw_policy_alloc -- policy_alloc 0x65727278
APPFW FUNC:appfw_policy_alloc -- Policy 0x65727278 is set to valid
APPFW FUNC:appfw_policy_alloc -- Policy myPolicy has been created
APPFW FUNC:appfw_policy_command -- memlock policy 0x65727278

! Debugging sample for application (HTTP) creation

Router(cfg-appfw-policy)# application httpAPPFW FUNC:appfw_http_command
APPFW FUNC:appfw_http_appl_find
APPFW FUNC:appfw_http_appl_find -- Application not found
APPFW FUNC:appfw_http_appl_alloc
APPFW FUNC:appfw_http_appl_alloc -- appl_http 0x64D7A25C
APPFW FUNC:appfw_http_appl_alloc -- Application HTTP parser structure 64D7A25C created

! Debugging sample for HTTP-specific application inspection
Router(cfg-appfw-policy-http)#
Router(cfg-appfw-policy-http)# strict-http action reset alarm
APPFW FUNC:appfw_http_subcommand
APPFW FUNC:appfw_http_subcommand -- strict-http cmd turned on

Router# debug appfw detailed

APPFW Detailed Debug debugging is on
fw7-7206a#debug appfw object-creation
APPFW Object Creations debugging is on
fw7-7206a#debug appfw object-deletion
APPFW Object Deletions debugging is on
```

Configuration Examples for Setting Up an HTTP Inspection Engine

This section contains the following configuration example:

- [Setting Up and Verifying an HTTP Inspection Engine: Example, page 9](#)

Setting Up and Verifying an HTTP Inspection Engine: Example

The following example show 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 put 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 put 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
    
```

Additional References

The following sections provide references related to the HTTP Inspection Engine feature.

Related Documents

Related Topic	Document Title
Firewall commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Security Command Reference , Release 12.3T

Standards

Standards	Title
No new or modified standards are supported by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported 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
RFC 2616	<i>Hypertext Transfer Protocol -- HTTP/1.1</i>

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 new and modified commands.

New Commands

Global Configuration Commands

- [appfw policy-name](#)
- [application](#)

Application Firewall Policy HTTP (cfg-appfw-policy-http) Configuration Commands

- [audit-trail](#)
- [content-length](#)
- [content-type-verification](#)
- [max-header-length](#)
- [max-uri-length](#)
- [port-misuse](#)
- [request-method](#)
- [strict-http](#)
- [timeout](#)
- [transfer-encoding type](#)

Privileged Exec Commands

- [debug appfw](#)
- [show appfw](#)

Modified Command

- [ip inspect name](#)

appfw policy-name

To define an application firewall policy and put the router in application firewall policy configuration mode, use the **appfw policy-name** command in global configuration mode. To remove a policy from the router configuration, use the **no** form of this command.

appfw policy-name *policy-name*

no appfw policy-name *policy-name*

Syntax Description

<i>policy-name</i>	Name of application policy.
--------------------	-----------------------------

Defaults

If this command is not issued, an application firewall policy cannot be created.

Command Modes

Global configuration

Command History

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

Usage Guidelines

This command puts the router in application firewall policy (*appfw-policy-protocol*) configuration mode, which allows you to begin defining the application firewall policy that will later be applied to the Cisco IOS Firewall via the **ip inspect name** command.

What Is an Application Firewall Policy?

The application firewall uses static signatures to detect security violations. A static signature is a collection of parameters that specifies which protocol conditions must be met before an action is taken. (For example, a signature may specify that an HTTP data stream containing the POST method must reset the connection.) These protocol conditions and reactions are defined by the end user via a command-line interface (CLI) to form an application firewall policy (also known as a security policy).

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

Related Commands

Command	Description
application	Puts the router in <i>appfw-policy-protocol</i> configuration mode and begin configuring inspection parameters for a given protocol.
ip inspect name	Defines a set of inspection rules.

application

To put the router in `appfw-policy-protocol` configuration mode and begin configuring inspection parameters for a given protocol, use the **application** command in application firewall policy configuration mode. To remove protocol-specific rules, use the **no** form of this command.

application *protocol*

no application *protocol*

Syntax Description

<i>protocol</i>	Protocol-specific traffic will be inspected. Currently, the only supported protocol is HTTP (specified via the http keyword), which defines the web policy.
-----------------	--

Defaults

You cannot set up protocol-specific inspection parameters.

Command Modes

Application firewall policy configuration

Command History

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

Usage Guidelines

This command puts the router in `appfw-policy-protocol` configuration mode, where “*protocol*” is dependent upon the specified protocol. Because HTTP is currently the only available protocol, the configuration mode is “`appfw-policy-http`.”

HTTP-Specific Inspection Commands

After you issue the **application** command and enter the `appfw-policy-http` configuration mode, begin configuring inspection parameters for HTTP traffic by issuing any of the following commands:

- **audit-trail**
- **content-length**
- **content-type-verification**
- **max-header-length**
- **max-uri-length**
- **port-misuse**
- **request-method**
- **strict-http**
- **timeout**
- **transfer-encoding**

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

Related Commands

Command	Description
appfw policy-name	Defines an application firewall policy and puts the router in application firewall policy configuration mode.

audit-trail

To turn audit trail messages on or off, use the **audit-trail** command in appfw-policy-http configuration mode. To return to the default value, use the **no** form of this command.

audit-trail {on | off}

no audit-trail {on | off}

Syntax Description

on	Audit trail messages are generated.
off	Audit trail messages are not generated.

Defaults

If this command is not issued, the default value specified via the **ip inspect audit-trail** command will be used.

Command Modes

appfw-policy-http configuration

Command History

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

Usage Guidelines

The **audit-trail** command will override the **ip inspect audit-trail** global command.

Examples

The following example, which shows how to define the HTTP application firewall policy “mypolicy,” enables audit trail messages for the given policy. This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! Define the HTTP policy.
appfw policy-name mypolicy
  application http
  audit trail on
  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
!
```

Related Commands

Command	Description
ip inspect audit-trail	Turns on audit trail messages.

content-length

To permit or deny HTTP traffic through the firewall on the basis of message size, use the **content-length** command in `appfw-policy-http` configuration mode. To remove message-size limitations from your configuration, use the **no** form of this command.

content-length { *min bytes max bytes* | *min bytes* | *max bytes* } **action** { **reset** | **allow** } [**alarm**]

no content-length { *min bytes max bytes* | *min bytes* | *max bytes* } **action** { **reset** | **allow** } [**alarm**]

Syntax Description

min bytes	Minimum content length, in bytes, allowed per message. Number of bytes range: 0 to 65535.
max bytes	Maximum content length, in bytes, allowed per message. Number of bytes range: 0 to 65535.
action	Messages whose size do not meet the minimum or exceed the maximum number of bytes are subject to the specified action (reset or allow).
reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
allow	Forwards the packet through the firewall.
alarm	(Optional) Generates system logging (syslog) messages for the given action.

Defaults

If this command is not enabled, message size is not considered when permitting or denying HTTP messages.

Command Modes

`appfw-policy-http` configuration

Command History

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

Usage Guidelines

All messages exceeding the specified content-length range, will be subjected to the configured action (**reset** or **allow**).

Examples

The following example, which shows how to define the HTTP application firewall policy “mypolicy,” will not permit HTTP messages longer than 1 byte. This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! Define the HTTP policy.
appfw policy-name mypolicy
  application http
    strict-http action allow alarm
    content-length max 1 action allow alarm
    content-type-verification match-req-resp 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
!
!
```

content-type-verification

To permit or deny HTTP traffic through the firewall on the basis of content message type, use the **content-type-verification** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

content-type-verification [match-req-resp] action {reset | allow} [alarm]

no content-type-verification [match-req-resp] action {reset | allow} [alarm]

Syntax Description

match-req-resp	(Optional) Verifies the content type of the HTTP response against the accept field of the HTTP request.
action	Messages that match the specified content type are subject to the specified action (reset or allow).
reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
allow	Forwards the packet through the firewall.
alarm	(Optional) Generates system logging (syslog) messages for the given action.

Defaults

If this command is not issued, all traffic will be allowed.

Command Modes

appfw-policy-http configuration

Command History

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

Usage Guidelines

After the **content-type-verification** command is issued, all HTTP messages are subjected to the following inspections:

- Verify that the message header’s content type is listed as a supported content type. (See [Table 1](#).)
- Verify that the header’s content type matches the content of the message data or entity body portion of the message.

[Table 1](#) contains a list of supported content types.

Table 1 HTTP Header Supported Content Types

Supported Content Types
audio/*
audio/basic
audio/midi
audio/mpeg

Table 1 HTTP Header Supported Content Types (Continued)

Supported Content Types
audio/x-adpcm
audio/x-aiff
audio/x-ogg
audio/x-wav
application/msword
application/octet-stream
application/pdf
application/postscript
application/vnd.ms-excel
application/vnd.ms-powerpoint
application/x-gzip
application/x-java-arching
application/x-java-xm
application/zip
image/*
image/cgf
image/gif
image/jpeg
image/png
image/tiff
image/x-3ds
image/x-bitmap
image/x-niff
image/x-portable-bitmap
image/x-portable-greymap
image/x-xpm
text/*
text/css
text/html
text/plain
text/richtext
text/sgml
text/xmcd
text/xml
video/*
video/-flc

Table 1 HTTP Header Supported Content Types (Continued)

Supported Content Types
video/mpeg
video/quicktime
video/sgi
video/x-avi
video/x-fli
video/x-mng
video/x-msvideo

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! Define the HTTP policy.
appfw policy-name mypolicy
  application http
    strict-http action allow alarm
    content-length max 1 action allow alarm
    content-type-verification match-req-resp 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
!
!
```

debug appfw

To display debug messages about Cisco IOS Firewall events, use the **debug appfw** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

```
debug appfw { application protocol | function-trace | object-creation | object-deletion | events | timers | detailed }
```

```
no debug appfw { application protocol | function-trace | object-creation | object-deletion | events | timers | detailed }
```

Syntax Description	
application <i>protocol</i>	Displays messages about protocol events of firewall-inspected applications, including details about the protocol's packets. Currently, the only supported protocol is HTTP. (Issue the http keyword.)
function-trace	Displays messages about software functions called by Cisco IOS Firewall.
object-creation	Displays messages about software objects that are being created by Cisco IOS Firewall. Cisco IOS firewall-inspected sessions begin when the object is created.
object-deletion	Displays messages about software objects that are being deleted by Cisco IOS Firewall. Cisco IOS firewall-inspected sessions close when the object is deleted.
events	Displays messages about Cisco IOS software events, including Cisco IOS Firewall packet processing.
timers	Displays messages about Cisco IOS Firewall timer events, such as an idle timeout by the Cisco IOS firewall.
detailed	Detailed information for all other enabled Cisco IOS firewall debugging is displayed.
	Note This keyword should be used in conjunction with other Cisco IOS firewall debugging commands.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

Examples The following sample configuration shows how to configure an HTTP policy with application firewall debugging enabled:

```
Router(config)# appfw policy-name myPolicy APPFW FUNC:appfw_policy_find
APPFW FUNC:appfw_policy_find -- Policy myPolicy is not found
APPFW FUNC:appfw_policy_alloc
APPFW FUNC:appfw_policy_alloc -- policy_alloc 0x65727278
APPFW FUNC:appfw_policy_alloc -- Policy 0x65727278 is set to valid
APPFW FUNC:appfw_policy_alloc -- Policy myPolicy has been created
APPFW FUNC:appfw_policy_command -- memlock policy 0x65727278
```

! Debugging sample for application (HTTP) creation

```
Router(cfg-appfw-policy)# application httpAPPFW FUNC:appfw_http_command
APPFW FUNC:appfw_http_appl_find
APPFW FUNC:appfw_http_appl_find -- Application not found
APPFW FUNC:appfw_http_appl_alloc
APPFW FUNC:appfw_http_appl_alloc -- appl_http 0x64D7A25C
APPFW FUNC:appfw_http_appl_alloc -- Application HTTP parser structure 64D7A25C created
```

! Debugging sample for HTTP-specific application inspection

```
Router(cfg-appfw-policy-http)#
Router(cfg-appfw-policy-http)# strict-http action reset alarm
APPFW FUNC:appfw_http_subcommand
APPFW FUNC:appfw_http_subcommand -- strict-http cmd turned on
```

Router# **debug appfw detailed**

```
APPFW Detailed Debug debugging is on
fw7-7206a#debug appfw object-creation
APPFW Object Creations debugging is on
fw7-7206a#debug appfw object-deletion
APPFW Object Deletions debugging is on
```

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 2 or Table 3 .
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 2 .

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 2 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 3](#) lists the supported application-layer protocols.

Table 3 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.

max-header-length

To permit or deny HTTP traffic on the basis of the message header length, use the **max-header-length** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

max-header-length request bytes response bytes action {reset | allow} [alarm]

no max-header-length request bytes response bytes action {reset | allow} [alarm]

Syntax Description		
request bytes		Maximum header length, in bytes, allowed in the request message. Number of bytes range: 0 to 65535.
response bytes		Maximum header length, in bytes, allowed in the response message. Number of bytes range: 0 to 65535.
action		Messages that exceed the maximum size are subject to the specified action (reset or allow).
reset		Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
allow		Forwards the packet through the firewall.
alarm		(Optional) Generates system logging (syslog) messages for the given action.

Defaults

If this command is not issued, all traffic is permitted.

Command Modes

appfw-policy-http configuration

Command History

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

Usage Guidelines

All message header lengths exceeding the configured maximum size will be subjected to the specified action (**reset** or **allow**).

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

max-uri-length

To permit or deny HTTP traffic on the basis of the URI length in the request message, use the **max-uri-length** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

```
max-uri-length bytes action {reset | allow} [alarm]
```

```
no max-uri-length bytes action {reset | allow} [alarm]
```

Syntax Description		
	<i>bytes</i>	Number of bytes ranging from 0 to 65535.
	action	Messages that exceed the maximum URI length are subject to the specified action (reset or allow).
	reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
	allow	Forwards the packet through the firewall.
	alarm	(Optional) Generates system logging (syslog) messages for the given action.

Defaults

If this command is not issued, all traffic is permitted.

Command Modes

appfw-policy-http configuration

Command History

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

Usage Guidelines

All URI lengths exceeding the configured value will be subjected to the specified action (**reset** or **allow**).

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

port-misuse

To permit or deny HTTP traffic through the firewall on the basis of specified applications in the HTTP message, use the **port-misuse** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

```
port-misuse {p2p | tunneling | im | default} action {reset | allow} [alarm]
```

```
no port-misuse {p2p | tunneling | im | default} action {reset | allow} [alarm]
```

Syntax Description		
p2p	Peer-to-peer protocol applications subject to inspection: Kazaa and Gnutella.	
tunneling	Tunneling applications subject to inspection: HTTPPort/HTTPHost, GNU Httptunnel, GotoMyPC, Firethru, Http-tunnel.com Client	
im	Instant messaging protocol applications subject to inspection: Yahoo Messenger.	
default	All applications are subject to inspection.	
action	Applications detected within the HTTP messages that are outside of the specified application are subject to the specified action (reset or allow).	
reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.	
allow	Forwards the packet through the firewall.	
alarm	(Optional) Generates system logging (syslog) messages for the given action.	

Defaults If this command is not enabled, HTTP messages are permitted through the firewall if any of the applications are detected within the message.

Command Modes appfw-policy-http configuration

Command History	Release	Modification
	12.3(14)T	This command was introduced.

Examples The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

request-method

To permit or deny HTTP traffic according to either the request methods or the extension methods, use the **request-method** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

```
request-method { rfc rfc-method | extension extension-method } action { reset | allow } [alarm]
```

```
no request-method { rfc rfc-method | extension extension-method } action { reset | allow } [alarm]
```

Syntax Description		
rfc		Specifies that the supported methods of RFC 2616, <i>Hypertext Transfer Protocol—HTTP/1.1</i> , are to be used for traffic inspection.
<i>rfc-method</i>		Any one of the following RFC 2616 methods can be specified: connect , default , delete , get , head , options , post , put , trace .
extension		Specifies that the extension methods are to be used for traffic inspection.
<i>extension-method</i>		Any one of the following extension methods can be specified: copy , default , edit , getattribute , getproperties , index , lock , mkdir , move , revadd , relabel , revlog , save , setattribute , startrev , stoprev , unedit , unlock , .
action		Methods and extension methods outside of the specified method are subject to the specified action (reset or allow).
reset		Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
allow		Forwards the packet through the firewall.
alarm		(Optional) Generates system logging (syslog) messages for the given action.

Defaults If a given method is not specified, all methods and extension methods are supported with the reset alarm action.

Command Modes appfw-policy-http configuration

Command History	Release	Modification
	12.3(14)T	This command was introduced.

Usage Guidelines Only methods configured by the **request-method** command are allowed through the firewall; all other HTTP traffic is subjected to the specified action (**reset** or **allow**).

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

show appfw

To display application firewall policy configuration information, use the **show appfw configuration** command in privileged EXEC mode.

show appfw configuration [*name*]

Syntax Description	<i>name</i> (Optional) Displays information only for the specified policy.
---------------------------	--

Defaults If no keywords are specified, information for all policies is shown.

Command Modes Privileged EXEC

Command History	Release	Modification
	12.3(14)T	This command was introduced.

Usage Guidelines Use this command to display information regarding the application firewall policy configuration.

Examples This sample output for the **show appfw configuration** command and the **show ip inspect configuration** command displays the configuration for the inspection rule “mypolicy,” which has been applied to all incoming HTTP traffic on the FastEthernet0/0 interface. In this example, you can see that all available HTTP inspection parameters have been defined.

```
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
show ip inspect	Displays firewall configuration and session information.

strict-http

To allow HTTP messages to pass through the firewall or to reset the TCP connection when HTTP noncompliant traffic is detected, use the **strict-http** command in appfw-policy-http configuration mode. To disable configured settings, use the **no** form of this command.

```
strict-http action {reset | allow} [alarm]
```

```
no strict-http action {reset | allow} [alarm]
```

Syntax Description

action	HTTP messages are subject to the specified action (reset or allow).
reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.
allow	Forwards the packet through the firewall.
alarm	(Optional) Generates system logging (syslog) messages for the given action.

Defaults

If this command is not enabled, all traffic will be allowed through the firewall.

Command Modes

appfw-policy-http configuration

Command History

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

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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  
!  
!
```

timeout

To override the global TCP idle timeout value for HTTP traffic, use the **timeout** command in `appfw-policy-http` configuration mode. To return to the default value, use the **no** form of this command.

timeout *seconds*

no timeout *seconds*

Syntax Description	<i>seconds</i>	Idle timeout value. Available range: 5 to 43200 (12 hours).
--------------------	----------------	---

Defaults	If this command is not issued, the default value specified via the ip inspect tcp idle-time command will be used.
----------	--

Command Modes	appfw-policy-http configuration
---------------	---------------------------------

Command History	Release	Modification
	12.3(14)T	This command was introduced.

Examples	The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.
----------	--

```
! 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
    timeout 60
!
!
! 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
!
!
```

Related Commands

Command	Description
ip inspect tcp idle-time	Specifies the TCP idle timeout (the length of time a TCP session will be managed while there is no activity).

transfer-encoding type

To permit or deny HTTP traffic according to the specified transfer-encoding of the message, use the **transfer-encoding type** command in appfw-policy-http configuration mode. To disable this inspection parameter, use the **no** form of this command.

```
transfer-encoding type { chunked | compress | deflate | gzip | identity | default } action { reset | allow } [alarm]
```

```
no transfer-encoding type { chunked | compress | deflate | gzip | identity | default } action { reset | allow } [alarm]
```

Syntax Description		
chunked	Encoding format (specified in RFC 2616, <i>Hypertext Transfer Protocol—HTTP/1</i>) in which the body of the message is transferred in a series of chunks; each chunk contains its own size indicator.	
compress	Encoding format produced by the UNIX “compress” utility.	
deflate	“ZLIB” format defined in RFC 1950, <i>ZLIB Compressed Data Format Specification version 3.3</i> , combined with the “deflate” compression mechanism described in RFC 1951, <i>DEFLATE Compressed Data Format Specification version 1.3</i> .	
gzip	Encoding format produced by the “gzip” (GNU zip) program.	
identity	Default encoding, which indicates that no encoding has been performed.	
default	All of the transfer encoding types.	
action	Encoding types outside of the specified type are subject to the specified action (reset or allow).	
reset	Sends a TCP reset notification to the client or server if the HTTP message fails the mode inspection.	
allow	Forwards the packet through the firewall.	
alarm	(Optional) Generates system logging (syslog) messages for the given action.	

Defaults If a given type is not specified, all transfer-encoding types are supported with the reset alarm action.

Command Modes appfw-policy-http configuration

Command History	Release	Modification
	12.3(14)T	This command was introduced.

Usage Guidelines Only encoding types specified by the **transfer-encoding-type** command are allowed through the firewall.

Examples

The following example shows how to define the HTTP application firewall policy “mypolicy.” This policy includes all supported HTTP policy rules. After the policy is defined, it is applied to the inspection rule “firewall,” which will inspect all HTTP traffic entering the FastEthernet0/0 interface.

```
! 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
!
!
```

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.

