



Firewall N2H2 Support

The Firewall N2H2 Support feature provides users with an additional option when choosing the URL filter vendor. Just like the Websense URL filtering server, N2H2 interacts with your Cisco IOS firewall (also known as Cisco Secure Integrated Software [CSIS]) to allow you to prevent users from accessing specified websites on the basis of some policy. The Cisco IOS firewall works with the N2H2 Internet Filtering Protocol (IFP) server to know whether a particular URL should be allowed or denied (blocked).

Feature Specifications for the Firewall N2H2 Support feature

Feature History

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

Supported Platforms

For platforms supported in Cisco IOS Release 12.2(11)YU and 12.2(15)T, consult Cisco Feature Navigator.

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 Firewall N2H2 Support, page 2](#)
- [Information About Cisco N2H2 Support, page 2](#)
- [How to Configure N2H2 URL Support, page 5](#)
- [Configuration Examples for Firewall and Webserver, page 11](#)
- [Additional References, page 16](#)
- [Command Reference, page 17](#)
- [Glossary, page 46](#)

Restrictions for Firewall N2H2 Support

N2H2 IFP (Server) Requirement

To enable this feature, you must have at least one N2H2 server; however, two or more N2H2 servers are preferred. Although there is no limit to the number of N2H2 servers you may have, and you can configure as many servers as you wish, only one server will be active at any given time—the primary server. URL lookup requests will be sent only to the primary server.

URL Filtering Support Restriction

This feature supports only one active URL filtering scheme at a time. (Before enabling N2H2 URL filtering, you should always ensure that there is not another URL filtering scheme configured, such as Websense.)

Username Restriction

N2H2 requires the username to be supplied with the URL lookup request. Thus, the user-based policy will not work with N2H2 because the current Cisco IOS software does not retrieve the username.

Protocol Used to Communicate Between Firewall and N2H2 Server Restriction

TCP is currently the only protocol used to communicate between the Cisco IOS firewall (UNIX FileSystem [UFS]) and the N2H2 server.

Information About Cisco N2H2 Support

To configure Firewall N2H2 support, you must understand the following concepts:

- [Benefits of Firewall N2H2 Support, page 2](#)
- [Feature Design of Firewall N2H2 Support, page 4](#)
- [Supported N2H2 Filtering Methods, page 5](#)

Benefits of Firewall N2H2 Support

The Cisco IOS Firewall N2H2 Support feature provides an Internet management application that allows you to control web traffic for a given host or user on the basis of a specified security policy. In addition, the following functions are available in this feature:

Primary and Secondary Servers

When users configure multiple N2H2 servers, the firewall will use only one server at a time—the primary server; all other servers are called secondary servers. When the primary server becomes unavailable for any reason, it becomes a secondary server and one of the secondary servers becomes the primary server.

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

When all the servers are down and the system is in allow mode, a periodic event that occurs for each minute will trace through the server list, trying to bring up a server by opening a TCP connection. If the TCP connection is successfully opened, the server is considered to be up, and the system will return to operational mode.

IP Cache Table

This function provides an IP cache table that contains the IP addresses of web servers whose underlying URLs can be accessed by all users and hosts.

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

The idle time value is fixed at 10 minutes. The absolute time value is taken from the N2H2 lookup response, which is often greater than 15 hours. The absolute value for cache entries made out of exclusive-domains is 12 hours. The maximum number of cache entries is configurable.

To configure cache table parameters, use the `ip urlfilter cache` command.

Packet Buffering

This function allows you to increase the maximum number of HTTP responses that a Cisco IOS firewall can hold. If the HTTP responses arrive prior to an N2H2 server reply, this buffering scheme allows your firewall to store a maximum of 200 HTTP responses. (After 200 responses have been reached, the firewall will drop further responses.) The responses will remain in the buffer until an allow or deny message is received from N2H2: if the status indicates that the URL is allowed, the firewall will release the HTTP responses in the buffer to the browser of the end user; if the status indicates that the URL is blocked, the firewall will discard the HTTP responses in the buffer and close the connection to both ends. This function prevents numerous HTTP responses from overwhelming your system.

To configure the maximum number of HTTP responses for your firewall, use the `ip urlfilter max-resp-pak` command.

Exclusive Domains

This function provides a configurable list of domain names so that the Cisco IOS firewall does not have to send a lookup request to the N2H2 server for the HTTP traffic that is destined for one of the domains in the exclusive list. Thus, the N2H2 server does not have to deal with look-up requests for HTTP traffic that is destined for a host that has already been marked as “allowed.”

Flexibility when entering domain names is also provided; that is, the user can enter the complete domain name or a partial domain name. If the user adds a complete domain name such as “www.cisco.com” to the exclusive domain list, all HTTP traffic whose URLs are destined for this domain (such as www.cisco.com/news and www.cisco.com/index) will be excluded from the N2H2 URL filtering policies and, on the basis of the configuration, the URLs will be permitted or blocked (denied).

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

To configure an exclusive domain list, use the `ip urlfilter exclusive-domain` command.

Allow Mode

The system will go into allow mode when connections to all the N2H2 servers are down. The system will return to normal mode when a connection to at least one web N2H2 server is up. Allow mode directs your system to forward or drop all packets on the basis of the configurable allow mode setting. By default, allow mode is off, so all HTTP requests are forbidden if all N2H2 servers are down.

To configure allow mode for your system, use the `ip urlfilter allowmode` command.

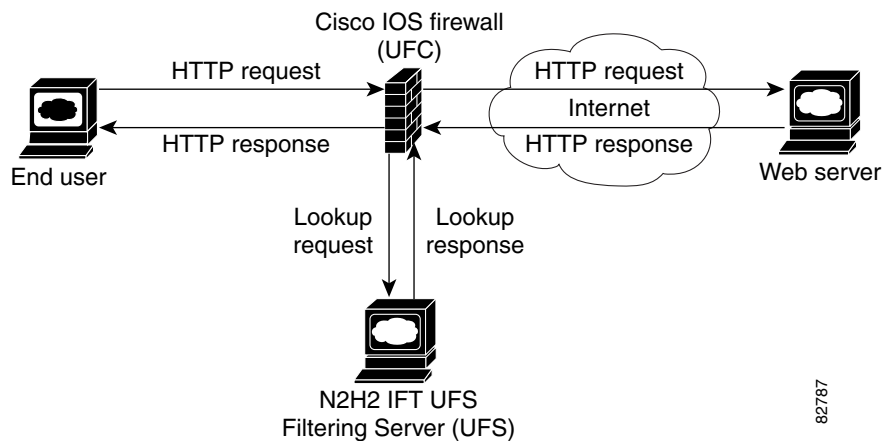
Feature Design of Firewall N2H2 Support

**Note**

This feature assumes that the N2H2 server will be part of a protected network and that requests from the Cisco IOS firewall will not travel over any unprotected network to reach the N2H2 server.

Figure 1 and the corresponding steps explain a sample URL filtering network topology.

Figure 1 Cisco IOS Firewall N2H2 URL Filtering Sample Topology



1. The end user browses a page on the web server, and the browser sends an HTTP request.
2. After the Cisco IOS firewall receives this request, it forwards the request to the web server, while simultaneously extracting the URL and sending a look-up request to the N2H2 server.
3. After the N2H2 server receives the look-up request, it checks its database to see whether it should permit or deny the URL; it returns a permit or deny status via a look-up response to the Cisco IOS firewall.
4. After the Cisco IOS Firewall receives this look-up response, it performs one of the following functions:
 - If the look-up response permits the URL, it sends the HTTP response to the end user.
 - If the look-up response denies the URL, the N2H2 server redirects the user to its own internal web server, which displays a message that describes the category under which the URL is blocked; thereafter, the connection is reset to both ends.

Supported N2H2 Filtering Methods

The Cisco IOS firewall supports most of the filtering methods that are supported by the N2H2 server. [Table 1](#) lists N2H2 filtering methods and identifies which methods are supported by Cisco.

Table 1 N2H2 Filtering Methods Supported on Cisco IOS Firewall

N2H2 Filtering Method	Description	Supported by Cisco IOS Firewall?
Client-IP-based filtering	Filtering is applied to specified client IP addresses	Yes
Global filtering	Filtering is applied to all users, groups, and IP addresses	Yes
User-based filtering	Filtering is applied to a specified user	No

How to Configure N2H2 URL Support

To configure your Cisco IOS firewall to interact with at least one N2H2 server to provide URL filtering, configure the following procedures:

- [Configuring Cisco IOS Firewall N2H2 URL Filtering, page 5](#) (required)
- [Verifying Firewall and N2H2 URL Filtering, page 10](#) (optional)
- [Maintaining the Cache Table, page 10](#) (optional)
- [Monitoring the URL Filter Subsystems, page 11](#) (optional)

Configuring Cisco IOS Firewall N2H2 URL Filtering

N2H2 is based on a pass-through filtering technology, which is the most accurate, reliable, and scalable method of Internet filtering. Pass-through filtering requires all requests for web pages to pass through an Internet control point, such as a firewall, proxy server, or caching device. N2H2 is integrated with these control points and checks each request to determine whether it should be allowed or denied. All responses are logged for reporting purposes.

Prerequisites

- Before enabling N2H2 URL filtering, you should always ensure that there is not another URL filtering scheme configured, such as Websense. If you try to enter a new filtering scheme when one already exists, the new scheme will be ignored, and the system will display an error message that says, “different URL filtering scheme cannot co-exist.”
- URL filtering does not have an interface-specific command. It relies on Cisco IOS firewall C HTTP inspection to classify the traffic that needs filtering. This makes the configuration of Cisco IOS firewall inspection mandatory for the URL filtering feature to work. For more details on Cisco IOS firewall configuration, refer to the chapter “Cisco IOS Firewall Overview” in the IOS Security Configuration Guide, Release 12.2.

Restrictions

Enabling HTTP inspection (via the **ip inspect name** command) triggers the Java applet scanner, which is very CPU intensive. The only way to stop the Java applet scanner is to specify the **java-list access-list** option and configure a standard access-list to allow any traffic. Configuring URL filtering without enabling the **java-list access-list** option will severely impact performance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip inspect name** *inspection-name* **http** [**urlfilter**] [**java-list** *access-list*] [**alert** {**on** | **off**}] [**timeout** *seconds*] [**audit-trail** {**on** | **off**}]
4. **ip urlfilter server vendor** {**websense** | **n2h2**} *ip-address* [**port** *port-number*] [**timeout** *seconds*] [**retransmit** *number*]
5. **ip urlfilter alert**
6. **ip urlfilter audit-trail**
7. **ip urlfilter urlf-server-log**
8. **ip urlfilter exclusive-domain** {**permit** | **deny**} *domain-name*
9. **ip urlfilter cache** *number*
10. **ip urlfilter allowmode** [**on** | **off**]
11. **ip urlfilter max-resp-pak** *number*
12. **ip urlfilter max-request** *number*
13. **interface** *type slot/port*
14. **ip inspect inspection-name** {**in** | **out**}

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
<p>Step 3</p> <pre>ip inspect name inspection-name http [urlfilter] [java-list access-list] [alert {on off}] [timeout seconds] [audit-trail {on off}]</pre> <p>Example: Router(config)# ip inspect name fw_urlf http urlfilter java-list 51 timeout 30</p>	<p>Turns on HTTP inspection. The urlfilter keyword associates URL filtering with HTTP inspection.</p> <p>Note You may configure two or more inspections in a router, but URL filtering will work only with the inspections in which the urlfilter keyword is enabled.</p> <p>Note Enabling HTTP inspection with or without any options 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.</p>
<p>Step 4</p> <pre>ip urlfilter server vendor {websense n2h2} ip-address [port port-number] [timeout seconds] [retransmit number]</pre> <p>Example: Router(config)# ip urlfilter server vendor websense 10.201.6.202</p>	<p>Configures an N2H2 server to interact with the firewall to filter HTTP requests based on a specified policy.</p> <ul style="list-style-type: none"> • <i>ip-address</i>—IP address of the vendor server. • port port-number—Port number that the vendor server listens on. The default port number is 4005. • timeout seconds—Length of time that the firewall will wait for a response from the vendor server. The default timeout is 5 minutes. • retransmit number—Number of times the firewall will retransmit the request when a response does not arrive. The default value is 2 times.
<p>Step 5</p> <pre>ip urlfilter alert</pre> <p>Example: Router(config)# ip urlfilter alert</p>	<p>(Optional) Enables the system alert, which displays system messages such as a server entering allow mode or going down.</p> <ul style="list-style-type: none"> • The system alert is enabled by default.
<p>Step 6</p> <pre>ip urlfilter audit-trail</pre> <p>Example: Router(config)# ip urlfilter audit-trail</p>	<p>(Optional) Enables the logging of messages into the syslog server of router.</p> <ul style="list-style-type: none"> • This function is disabled by default.
<p>Step 7</p> <pre>ip urlfilter urlf-server-log</pre> <p>Example: Router(config)# ip urlfilter urlf-server-log</p>	<p>(Optional) Enables the logging of system messages on the URL filtering server (the N2H2 server). This function is disabled by default.</p>

	Command or Action	Purpose
Step 8	<p>ip urlfilter exclusive-domain {permit deny} <i>domain-name</i></p> <p>Example: Router(config)# ip urlfilter exclusive-domain permit www.cisco.com</p>	<p>(Optional) Adds a domain name to or from the exclusive domain list so the firewall does not have to send look-up requests to the N2H2 server.</p> <ul style="list-style-type: none"> • permit—Permits all traffic destined for the specified domain name. • deny—Denies all traffic destined for the specified domain name. • <i>domain-name</i>—Domain name that is added or removed from the exclusive domain list.
Step 9	<p>ip urlfilter cache <i>number</i></p> <p>Example: Router(config)# ip urlfilter cache 4500</p>	<p>(Optional) Configures cache table parameters.</p> <ul style="list-style-type: none"> • <i>number</i>—Specifies the maximum number of destination IP addresses that can be cached into the cache table; the default is 5000.
Step 10	<p>ip urlfilter allowmode [on off]</p> <p>Example: Router(config)# ip urlfilter allowmode on</p>	<p>(Optional) Turns on the default mode of the filtering systems.</p> <ul style="list-style-type: none"> • on—Allows HTTP requests to pass to the end user if all N2H2 servers are down. • off—Blocks all HTTP requests if all N2H2 servers are down; off is the default setting.
Step 11	<p>ip urlfilter max-resp-pak <i>number</i></p> <p>Example: Router(config)# ip urlfilter max-resp-pak 150</p>	<p>(Optional) Configures the maximum number of HTTP responses that the firewall can keep in its packet buffer.</p> <ul style="list-style-type: none"> • The default value is 200. The maximum value is 20000, so you may set the max-resp-pak <i>number</i> to a value up to 20000.
Step 12	<p>ip urlfilter max-request <i>number</i></p> <p>Example: Router(config)# ip urlfilter max-request 500</p>	<p>(Optional) Sets the maximum number of outstanding requests that can exist at any given time.</p> <ul style="list-style-type: none"> • The default value is 1000.
Step 13	<p>interface <i>type slot/port</i></p> <p>Example: Router(config)# interface FastEthernet 0/0</p>	<p>Configures an interface type and enters interface configuration mode</p>
Step 14	<p>ip inspect inspection-name {in out}</p> <p>Example: Router(config-if)# ip inspect inspection-name out</p>	<p>Applies a set of inspection rules to an interface.</p> <ul style="list-style-type: none"> • URL filtering is associated with inspection, and inspection is an interface-specific command. Hence, the ip inspect command needs to be configured on an interface.

Troubleshooting Tips

This feature introduces the following alert messages:

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

This level three LOG_ERR-type message is displayed when a configured UFS goes down. When this happens, the firewall will mark the configured server as secondary, try to bring up one of the other secondary servers, and mark that server as the primary server. If there is no other server configured, the firewall will enter allow mode and display the “URLF-3-ALLOW_MODE” message.

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

This LOG_ERR type message is displayed when all UFSs are down and the system enters allow mode.



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

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

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

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

This LOG_WARNING-type message is displayed when the URL in a look-up request is too long; any URL longer than 3K will be dropped.

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

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

To display these alert messages, use the **ip urlfilter alert** command.

This feature introduces the following syslog messages:

- “%URLF-6-SITE_ALLOWED: Client 10.0.0.2:12543 accessed server 10.76.82.21:8080”

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

- “%URLF-4-SITE-BLOCKED: Access denied for the site ‘www.sports.com’; client 10.54.192.6:34557 server 172.24.50.12:80”

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

- “%URLF-6-URL_ALLOWED: Access allowed for URL http://www.n2h2.com/; client 10.54.192.6:54123 server 192.168.0.1:80”

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

- “%URLF-6-URL_BLOCKED: Access denied URL http://www.google.com; client 10.54.192.6:54678 server 172.19.14.2:80”

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

To display these syslog messages, use the `ip urlfilter audit-trail` command.

Verifying Firewall and N2H2 URL Filtering

To verify that the Firewall N2H2 Support feature is working, perform any of the following optional steps:

SUMMARY STEPS

1. `enable`
2. `show ip urlfilter cache`
3. `show ip urlfilter config`

DETAILED STEPS

Command or Action	Purpose
enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
show ip urlfilter cache Example: Router# show ip urlfilter cache	Displays the destination IP addresses that are cached into the cache table.
show ip urlfilter config Example: Router# show ip urlfilter config	Displays the size of the cache, the maximum number of outstanding requests, the allow mode state, and the list of configured N2H2 servers.
show ip urlfilter statistics Example: Router# show ip urlfilter statistics	Displays information such as the number of requests that are sent to the N2H2 server, the number of responses received from the N2H2 server, the number pending requests in the system, the number of failed requests, the number of blocked URLs.

Maintaining the Cache Table

To clear the cache table of a specified or all IP addresses, perform the following optional steps:

SUMMARY STEPS

1. `enable`

2. clear ip urlfilter cache

DETAILED STEPS

Command or Action	Purpose
enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
clear ip urlfilter cache { <i>ip-address</i> all } Example: Router# clear ip urlfilter cache all	Clears the cache table.

Monitoring the URL Filter Subsystems

To monitor the URL filter subsystems, perform the following optional steps:

SUMMARY STEPS

- enable
- debug ip urlfilter {function-trace | detailed | events}

DETAILED STEPS

Command or Action	Purpose
enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
debug ip urlfilter { function-trace detailed events } Example: Router# debug ip urlfilter detailed	Enables debugging information of URL filter subsystems. <ul style="list-style-type: none"> function-trace—Prints a sequence of important functions that are called when configuring URL filtering. detailed—Prints detailed information about various activities that occur during URL filtering. events—Prints various events such as queue event, timer event, and socket event.

Configuration Examples for Firewall and Webserver

This section provides the following comprehensive configuration example:

- [URL Filter Client \(Firewall\) Configuration Example, page 12](#)

URL Filter Client (Firewall) Configuration Example

The following example shows how to configure the Cisco IOS firewall (also known as the URL filter client [UFC]) for N2H2 URL filtering:

Topology:

```

End User-----LAN-----Fa0/0 -- Firewall -- S2/0----- Internet ---- Web Server
                        |
                        | Router
                        |
N2H2                    |
Server -----+
  
```

Router Configuration:

Example 1:

```

hostname fw9-7200b
!
!-----
! The following commands define the inspection rule "myfw," allowing
! the specified protocols to be inspected. Note that the "urlfilter"
! keyword entered for HTTP protocol enables URL filtering on HTTP
! traffic that are bound to this inspection.
!-----
!
ip inspect name myfw http urlfilter
ip inspect name myfw ftp
ip inspect name myfw smtp
ip inspect name myfw h323
!
!-----
! The following command sets the URL filtering cache table size to 12000.
!-----
ip urlfilter cache 12000
!
!-----
! The following commands configure three exclusive domains--
! two partial domains and one complete domain.
!-----
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
!
!-----
! The following two commands enable URL filtering Audit Trail and
! Alert messages.
!-----
ip urlfilter audit-trail
ip urlfilter alert
!
!-----
! The command configures the N2H2 URL filtering server installed
! on 192.168.3.1.
!-----
ip urlfilter server vendor n2h2 192.168.3.1
!
!-----
! Create Access Control List 102:
! ACL 102 denies all IP protocol traffic except for ICMP traffic.
! This means that only the return traffic for protocols defined in the
! inspection rule and the ICMP traffic is allowed access through the
  
```

```
! interface where this rule is applied.
!
! Note that ACL is given here for an example; it is not relevant
! to the URL filtering. The URL filtering will work without ACL also.
!-----
!
access-list 102 permit icmp any any
access-list 102 deny tcp any any
access-list 102 deny udp any any
access-list 102 deny ip any any
!
!
interface FastEthernet0/0
ip address 192.168.3.254 255.255.255.0
ip nat inside
no ip route-cache
no ip mroute-cache
!
interface Ethernet1/0
no ip route-cache
no ip mroute-cache
shutdown
duplex half
!
interface Ethernet1/1
no ip address
no ip mroute-cache
shutdown
duplex half
!
!-----
! The ACL and CBAC inspection rules are applied to the Serial2/0 interface.
! In this example, the ACL is applied IN, meaning that it applies to traffic
! inbound from the internet. The CBAC inspection rule myfw is applied OUT,
! meaning that CBAC inspects the traffic that goes out through the interface
! and controls return traffic to the router for an existing connection.
!-----
interface Serial2/0
ip address 10.6.9.7 255.255.0.0
ip access-group 102 in
ip nat outside
ip inspect myfw out
no ip directed-broadcast
no ip mroute-cache
!
ip nat inside source static 192.168.3.1 10.6.243.1
ip nat inside source static 192.168.3.2 10.6.243.2
ip nat inside source static 192.168.3.3 10.6.243.3
ip classless
ip route 192.168.0.30 255.255.255.255 10.6.0.1
!
!
line con 0
exec-timeout 0 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
password letmein
login
!
end
```

Example 2:

```
! In the above example, the CBAC can also be configured on the inbound
! FastEthernet0/0 interface as IN, in which case the CBAC inspects all
! the traffic that comes in on FastEthernet0/0 and controls return traffic
! that leaves out of this interface for an existing connection.
```

```
interface FastEthernet0/0
ip address 192.168.3.254 255.255.255.0
ip access-group 102 out
ip nat inside
ip inspect myfw in
no ip route-cache
no ip mroute-cache
!
!
hostname fw9-7200b
!
logging buffered 64000 debugging
enable secret 5 $1$qMOf$umPb75mb3sV27JpNbW//7.
!
clock timezone PST -8
clock summer-time PDT recurring
ip subnet-zero
ip cef
no ip domain lookup
!
ip inspect name test http urlfilter
ip urlfilter cache 5
ip urlfilter exclusive-domain permit .weapons.com
ip urlfilter exclusive-domain deny .nbc.com
ip urlfilter exclusive-domain permit www.cisco.com
ip urlfilter audit-trail
ip urlfilter alert
ip urlfilter server vendor n2h2 192.168.3.1
ip audit notify log
ip audit po max-events 100
ip port-map http port 8080
!
no voice hpi capture buffer
no voice hpi capture destination
!
mta receive maximum-recipients 0
!
interface FastEthernet0/0
ip address 192.168.3.254 255.255.255.0
ip access-group 101 out
ip nat inside
ip inspect test in
no ip route-cache
no ip mroute-cache
!
interface Ethernet1/0
ip address 10.6.9.7 255.255.0.0
ip nat outside
no ip route-cache
no ip mroute-cache
duplex half
!
interface Ethernet1/1
no ip address
no ip mroute-cache
shutdown
duplex half
```

```
!  
interface Ethernet1/2  
  no ip address  
  no ip mroute-cache  
  shutdown  
  duplex half  
!  
interface Ethernet1/3  
  no ip address  
  no ip mroute-cache  
  shutdown  
  duplex half  
!  
interface Serial2/0  
  no ip address  
  no ip mroute-cache  
  shutdown  
  dsu bandwidth 44210  
  framing c-bit  
  cablelength 10  
  serial restart_delay 0  
  fair-queue  
!  
ip nat pool devtest 10.6.243.21 10.6.243.220 netmask 255.255.0.0  
ip nat inside source list 1 pool devtest  
ip nat inside source static 192.168.3.1 10.6.243.1  
ip nat inside source static 192.168.3.2 10.6.243.2  
ip nat inside source static 192.168.3.3 10.6.243.3  
ip classless  
ip route 192.168.0.30 255.255.255.255 10.6.0.1  
no ip http server  
no ip http secure-server  
!  
ip pim bidir-enable  
!  
!  
access-list 101 deny tcp any any  
access-list 101 deny udp any any  
access-list 101 permit ip any any  
access-list 102 deny tcp any any  
access-list 102 deny udp any any  
access-list 102 permit ip any any  
dialer-list 1 protocol ip permit  
dialer-list 1 protocol ipx permit  
!  
!  
call rsvp-sync  
!  
!  
mgcp profile default  
!  
dial-peer cor custom  
!  
!  
!gatekeeper  
  shutdown  
!  
!  
line con 0  
  exec-timeout 0 0  
  stopbits 1  
line aux 0  
  stopbits 1  
line vty 0 4
```

```

password letmein
login
!
exception core-file sisu-devtest/coredump/fw9-7200b.core
exception dump 192.168.0.1
no scheduler max-task-time
!
end

```

Additional References

For additional information related to the Firewall N2H2 Support feature, refer to the following references:

- [Related Documents, page 16](#)
- [Standards, page 16](#)
- [MIBs, page 16](#)
- [RFCs, page 17](#)
- [Technical Assistance, page 17](#)

Related Documents

Related Topic	Document Title
Websense URL filtering information	<i>Firewall Websense URL Filtering</i> , Cisco IOS Release 12.2(15)T feature module
Additional Cisco IOS firewall configuration tasks and information	The part “Traffic Filtering and Firewalls” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.2
Additional Cisco IOS firewall commands	The part “Traffic Filtering and Firewalls” in the <i>Cisco IOS Security Command Reference</i> , Release 12.2
Cisco IOS firewall configuration	The chapter “Cisco IOS Firewall Overview” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.2

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

<http://tools.cisco.com/ITDIT/MIBS/servlet/index>

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

<http://www.cisco.com/register>

RFCs

RFCs ¹	Title
RFC 1945	<i>Hypertext Transfer Protocol — HTTP/1.0</i>
RFC 2616	<i>Hypertext Transfer Protocol — HTTP/1.1</i>

1. Not all supported RFCs are listed.

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. All other commands used with this feature are documented in the Cisco IOS Release 12.2 T command reference publications.

New Commands

- [clear ip urlfilter cache](#)
- [debug ip urlfilter](#)
- [ip urlfilter alert](#)
- [ip urlfilter allowmode](#)
- [ip urlfilter audit-trail](#)
- [ip urlfilter cache](#)

- **ip urlfilter exclusive-domain**
- **ip urlfilter max-request**
- **ip urlfilter max-resp-pak**
- **ip urlfilter server vendor**
- **ip urlfilter urlf-server-log**
- **show ip urlfilter cache**
- **show ip urlfilter config**
- **show ip urlfilter statistics**

Modified Command

- **ip inspect name**

clear ip urlfilter cache

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

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

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

Defaults This command is not enabled.

Command Modes EXEC

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

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

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

```
clear ip urlfilter cache 172.18.139.21
```

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

```
clear ip urlfilter cache all
```

Related Commands	Command	Description
	ip urlfilter cache	Configures cache parameters.
	show ip urlfilter cache	Displays the destination IP addresses that are cached into the cache table.

debug ip urlfilter

To enable debug information of URL filter subsystems, use the **debug ip urlfilter** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

debug ip urlfilter {function-trace | detailed | events}

no debug ip urlfilter {function-trace | detailed | events}

Syntax Description

function-trace	The system prints a sequence of important functions that are called when configuring URL filtering.
detailed	The system prints detailed information about various activities that occur during URL filtering.
events	The system prints various events such as queue event, timer event, and socket event.

Defaults

This command is not enabled.

Command Modes

Privileged EXEC

Command History

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

Examples

The following is sample output for the **debug ip urlfilter** command:

```
Router# debug ip urlfilter
urlfilter:
  Urlfilter Detailed Debugs debugging is on

Router# show ip urlfilter config

N2H2 URL Filtering is ENABLED

Primary N2H2 server configurations
=====
N2H2 server IP address:192.168.1.103
N2H2 server port:4005
N2H2 retransmission time out:6 (in seconds)
N2H2 number of retransmission:2

Secondary N2H2 servers configurations
=====
Other configurations
=====
Allow Mode:OFF
System Alert:ENABLED
Audit Trail:ENABLED
```

```
Log message on N2H2 server:DISABLED
Maximum number of cache entries:5
Maximum number of packet buffers:20
Maximum outstanding requests:1000
fwl_4#
1d15h:URLF:got a socket read event...
1d15h:URLF:socket recv failed.
1d15h:URLF:Closing the socket for server (192.168.1.103:4005)
1d15h:%URLF-3-SERVER_DOWN:Connection to the URL filter server 192.168.1.103 is down
1d15h:URLF:Opening a socket for server (192.168.1.103:4005)
1d15h:URLF:socket fd 0
1d15h:%URLF-5-SERVER_UP:Connection to an URL filter server(192.168.1.103) is made, the
router is returning from ALLOW MODE
1d15h:URLF:got cache idle timer event...
1d16h:URLF:got cache absolute timer event...
1d16h:URLF:got cache idle timer event...
1d16h:URLF:creating uis 0x63A95DB4, pending request 1
1d16h:URLF:domain name not found in the exclusive list
1d16h:URLF:got an cbac queue event...
1d16h:URLF:socket send successful...172.17.192.130:8080) -> 192.168.1.103:1052 seq
3344720064 wnd 24820
1d16h:URLF:holding pak 0x634A8A08 (172.17.192.130:8080) -> 192.168.1.103:1052 seq
3344721524 wnd 24820
1d16h:URLF:holding pak 0x634A98CC (172.17.192.130:8080) -> 192.168.1.103:1052 seq
3344722984 wnd 24820
1d16h:URLF:got a socket read event...
1d16h:URLF:socket recv (header) successful.
1d16h:URLF:socket recv (data) successful.
1d16h:URLF:n2h2 lookup code = 1
1d16h:URLF:Site/URL Blocked:sis 0x63675DC4, uis 0x63A95DB4
1d16h:%URLF-4-URL_BLOCKED:Access denied URL 'http://www.google.com/', client
192.168.1.103:1052 server 172.17.192.130:8080
1d16h:URLF:(192.168.1.103:1052) RST -> 172.17.192.130:8080 seq 3361738063 wnd 0
1d16h:URLF:(172.17.192.130:8080) FIN -> 192.168.1.103:1052 seq 3344720064 wnd 0
1d16h:URLF:deleting uis 0x63A95DB4, pending requests 0
1d16h:URLF:got cache idle timer event...
1d16h:URLF:creating uis 0x63A95DB4, pending request 1
1d16h:URLF:domain name not found in the exclusive list
1d16h:URLF:got an cbac queue event...
1d16h:URLF:socket send successfull...
1d16h:URLF:holding pak 0x634A812C (172.17.192.130:8080) -> 192.168.1.103:1101 seq
3589711120 wnd 24820
1d16h:URLF:holding pak 0x634A2E7C (172.17.192.130:8080) -> 192.168.1.103:1101 seq
3589712580 wnd 24820
1d16h:URLF:holding pak 0x634A3464 (172.17.192.130:8080) -> 192.168.1.103:1101 seq
3589714040 wnd 24820
1d16h:URLF:got a socket read event...
1d16h:URLF:socket recv (header) successful.
1d16h:URLF:socket recv (data) successful.
1d16h:URLF:n2h2 lookup code = 0
1d16h:%URLF-6-URL_ALLOWED:Access allowed for URL 'http://www.alcohol.com/', client
192.168.1.103:1101 server 172.17.192.130:8080
1d16h:URLF:Site/URL allowed:sis 0x6367D0C4, uis 0x63A95DB4
1d16h:URLF:releasing pak 0x634A812C:(172.17.192.130:8080) -> 192.168.1.103:1101 seq
3589711120 wnd 24820
1d16h:URLF:releasing pak 0x634A2E7C:(172.17.192.130:8080) -> 192.168.1.103:1101 seq
3589712580 wnd 24820
1d16h:URLF:releasing pak 0x634A3464:(172.17.192.130:8080) -> 192.168.1.103:1101 seq
3589714040 wnd 24820
1d16h:URLF:deleting uis 0x63A95DB4, pending requests 0
1d16h:URLF:got cache idle timer event...
1d16h:URLF:creating uis 0x63A9777C, pending request 1
1d16h:URLF:domain name not found in the exclusive list
1d16h:URLF:got an cbac queue event...
```

```
1d16h:URLF:socket send successful...
1d16h:URLF:got a socket read event...
1d16h:URLF:socket recv (header) successful.
1d16h:URLF:socket recv (data) successful.
1d16h:URLF:n2h2 lookup code = 1
1d16h:URLF:Site/URL Blocked:sis 0x63677ED4, uis 0x63A9777C
1d16h:%URLF-4-URL_BLOCKED:Access denied URL 'http://www.google.com/', client
192.168.1.103:1123 server 172.17.192.130:8080
1d16h:URLF:(192.168.1.103:1123) RST -> 172.17.192.130:8080 seq 3536466275 wnd 0
1d16h:URLF:(172.17.192.130:8080) FIN -> 192.168.1.103:1123 seq 3618929551 wnd 0
1d16h:URLF:deleting uis 0x63A9777C, pending requests 0
1d16h:URLF:got cache idle timer event...
```

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.

HTTP Inspection Syntax

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

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

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.
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 works only with numbered standard access lists.
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, audit trail messages 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) idle timeouts for the specified protocol, specify the number of seconds for a different idle timeout. This timeout overrides the global TCP and UPD timeouts but will not override the global Domain Name System (DNS) timeout.

Defaults

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

Command Modes

Global configuration

Command History

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

Usage Guidelines

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

To define a single set of inspection rules, configure inspection for all the desired application-layer protocols, and for TCP or UDP 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.

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.

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 you use a “placeholder” access list in the **ip inspect name inspection-name http** command, all Java applets will be blocked.

**Caution**

Cisco IOS Firewall 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. Cisco IOS Firewall also does not detect or block applets loaded via FTP, gopher, or HTTP on a nonstandard port.

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.

Use of the **urlfilter** Keyword

If you specify the **urlfilter** keyword, Cisco IOS Firewall will interact with a URL filtering software to control web traffic for a given host or user based upon a specified security policy.



Note

Enabling HTTP inspection with or without any option triggers the Java applet scanner, which is very 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.

Examples

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

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

Related Commands

Command	Description
access-list	Configures the access list mechanism for filtering frames by protocol type or vendor code.
ip inspect	Applies a set of inspection rules to an interface.
ip inspect alert-off	Disables Cisco IOS Firewall alert messages.
ip inspect audit trail	Turns on Cisco IOS Firewall audit trail messages, which will be displayed on the console after each CBAC session close.

ip urlfilter alert

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

ip urlfilter alert

no ip urlfilter alert

Syntax Description This command has no arguments or keywords.

Defaults URL filtering messages are enabled.

Command Modes Global configuration

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

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

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

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

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

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

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

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

This LOG_ERR type message is displayed when UFSs are down and the system enters into allow mode.

**Note**

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

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

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

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

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

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

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

ip urlfilter allowmode

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

ip urlfilter allowmode [on | off]

no ip urlfilter allowmode [on | off]

Syntax Description

on	(Optional) Allow mode is on.
off	(Optional) Allow mode is off.

Defaults

Allow mode is off.

Command Modes

Global configuration

Command History

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

Usage Guidelines

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

Examples

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

```
ip urlfilter allow-mode on
```

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

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

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

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

ip urlfilter audit-trail

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

ip urlfilter audit-trail

no ip urlfilter audit-trail

Syntax Description This command has no arguments or keywords.

Defaults This command is disabled.

Command Modes Global configuration

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

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

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

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

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

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

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

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

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

```
%URLF-6-URL_ALLOWED:Access allowed for URL http://www.n2h2.com/; client 12.54.192.6:54123  
server 192.168.0.1:80
```

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

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

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

ip urlfilter cache

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

ip urlfilter cache *number*

no ip urlfilter cache *number*

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

Defaults	Maximum number of destination IP addresses is 5000. The cache table is cleared every 12 hours.
----------	---

Command Modes	Global configuration
---------------	----------------------

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

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

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

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



Note

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

Examples

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

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

Related Commands

Command	Description
clear ip urlfilter cache	Clears the cache table.
show ip urlfilter cache	Displays the destination IP addresses that are cached into the cache table.

ip urlfilter exclusive-domain

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

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

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

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

Defaults This command is not enabled.

Command Modes Global configuration

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

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

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

Complete Domain Name

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

Partial Domain Name

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

Examples

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

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

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

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

ip urlfilter max-request

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

ip urlfilter max-request *number*

no ip urlfilter max-request *number*

Syntax Description	<i>number</i>	Maximum number of outstanding requests. The default value is 1000.
--------------------	---------------	--

Defaults	Maximum number of requests is 1000.
----------	-------------------------------------

Command Modes	Global configuration
---------------	----------------------

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

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



Note

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

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

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

Related Commands	Command	Description
	ip inspect name	Defines a set of inspection rules.
	ip urlfilter server vendor	Configures a vendor server for URL filtering.

ip urlfilter max-resp-pak

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

ip urlfilter max-resp-pak *number*

no ip urlfilter max-resp-pak *number*

Syntax Description

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

Defaults

200 HTTP responses

Command Modes

Global configuration

Command History

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

Usage Guidelines

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

Examples

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

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

ip urlfilter server vendor

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

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

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

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

Defaults A vendor server is not configured.

Command Modes Global configuration

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

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

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

Primary and Secondary Servers

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

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

Examples

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

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

Related Commands

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

ip urlfilter urlf-server-log

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

ip urlfilter urlf-server-log

no ip urlfilter urlf-server-log

Syntax Description

This command has no arguments or keywords.

Defaults

This command is disabled.

Command Modes

Global configuration

Command History

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

Usage Guidelines

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

Examples

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

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

show ip urlfilter cache

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

show ip urlfilter cache

Syntax Description This command has no arguments or keywords.

Defaults This command is not enabled.

Command Modes EXEC

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

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

```
Router# show ip urlfilter cache

Maximum number of entries allowed: 5000
Number of entries cached: 5
IP addresses cached ....
 10.64.128.54
 172.18.139.21
 10.76.82.25
 192.168.0.1
 10.0.1.2
```

[Table 2](#) describes the significant fields shown in the display.

Table 2 *show ip urlfilter cache Field Descriptions*

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

Related Commands

Command	Description
clear ip urlfilter cache	Clears the cache table.
ip urlfilter cache	Configures cache parameters.

show ip urlfilter config

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

show ip urlfilter config

Syntax Description This command has no arguments or keywords.

Defaults This command is not enabled.

Command Modes EXEC

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

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

```
Router# show ip urlfilter config

URL filter is ENABLED

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

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

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

Related Commands

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

show ip urlfilter statistics

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

show ip urlfilter statistics

Syntax Description This command has no arguments or keywords.

Defaults This command is not enabled.

Command Modes EXEC

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

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

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

```
Router# show ip urlfilter statistics

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

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

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

[Table 3](#) describes the significant fields shown in the display.

Table 3 *show ip urlfilter statistics Field Descriptions*

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

1. This value can be specified via the `ip urlfilter max-request` command.
2. This value can be specified via the `ip urlfilter max-resp-pak` command.
3. This value can be specified via the `ip urlfilter cache` command.

Related Commands

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

Glossary

ACL—Access Control List.

CSIS—Cisco Secure Integrated Software. CSIS is a content-based firewall that currently inspects application data, checks for protocol conformance, extracts the relevant port information to create dynamic access list entries that successfully allows return traffic, and closes the ports at the end of the session.

ICMP—Internet Control Message Protocol. ICMP is a network layer Internet protocol that reports errors and provides other information relevant to IP packet processing. ICMP is documented in RFC 792.

UFC—URL filter client. UFC is a separate process that accepts URLs from CSIS, forwards the URL to the Websense server, and process the replies from the vendor server (Websense or N2H2).

UFS—URL filter server. UFS is a generic name given to the vendor server (Websense or N2H2), which processes URLs and decides whether to allow or deny web traffic based on a given policy.

**Note**

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