Firewall N2H2 Support

Last Updated: June 14, 2011

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).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

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

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.

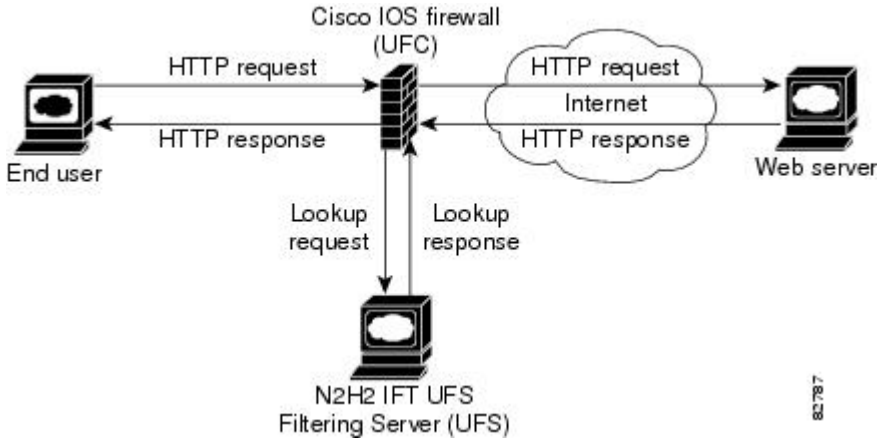
To configure allow mode for your system, use the **ip urlfilter allowmode** command.

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.

The figure below 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:
5. If the look-up response permits the URL, it sends the HTTP response to the end user.
6. 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. The table below 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

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.

- 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 IOS Security Configuration Guide, Release 12.2.

Note 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. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip inspect name** *inspection-name* http [**urlfilter**] [**java-list** *access-list*] [**alert** {**on** | **off**}] [**timeout** *seconds*] [**audit-trail** {**on** | **off**}] **Example:** `Router(config)# ip inspect name fw_urlf http urlfilter java-list 51 timeout 30`	Turns on HTTP inspection. The **urlfilter** keyword associates URL filtering with HTTP inspection. **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. **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.
Step 4	**ip urlfilter server vendor websense** | **n2h2**} *ip-address* [**port** *port-number*] [**timeout** *seconds*] [**retransmit** *number*]	Configures an N2H2 server to interact with the firewall to filter HTTP requests based on a specified policy. • *ip-address* --IP address of the vendor server. • **port** *port-number* --Port number that the vendor server listens on. The default port number is 4005.

	Command or Action	Purpose
	Example: `Router(config)# ip urlfilter server vendor websense 10.201.6.202`	• **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.
Step 5	**ip urlfilter alert** **Example:** `Router(config)# ip urlfilter alert`	(Optional) Enables the system alert, which displays system messages such as a server entering allow mode or going down. • The system alert is enabled by default.
Step 6	**ip urlfilter audit-trail** **Example:** `Router(config)# ip urlfilter audit-trail`	(Optional) Enables the logging of messages into the syslog server of router. • This function is disabled by default.
Step 7	**ip urlfilter urlf-server-log** **Example:** `Router(config)# ip urlfilter urlf-server-log`	(Optional) Enables the logging of system messages on the URL filtering server (the N2H2 server). This function is disabled by default.
Step 8	**ip urlfilter exclusive-domain permit** | **deny**} *domain-name* **Example:** `Router(config)# ip urlfilter exclusive-domain permit www.cisco.com`	(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. • **permit** --Permits all traffic destined for the specified domain name. • **deny** --Denies all traffic destined for the specified domain name. • *domain-name* --Domain name that is added or removed from the exclusive domain list.
Step 9	**ip urlfilter cache** *number* **Example:** `Router(config)# ip urlfilter cache 4500`	(Optional) Configures cache table parameters. • *number* --Specifies the maximum number of destination IP addresses that can be cached into the cache table; the default is 5000.
Step 10	**ip urlfilter allowmode** [**on** | **off**] **Example:** `Router(config)# ip urlfilter allowmode on`	(Optional) Turns on the default mode of the filtering systems. • **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	**ip urlfilter max-resp-pak** *number* **Example:** `Router(config)# ip urlfilter max-resp-pak 150`	(Optional) Configures the maximum number of HTTP responses that the firewall can keep in its packet buffer. • The default value is 200. The maximum value is 20000, so you may set the **max-resp-pak** *number* to a value up to 20000.

	Command or Action	Purpose
Step 12	**ip urlfilter max-request** *number* **Example:** `Router(config)# ip urlfilter max-request 500`	(Optional) Sets the maximum number of outstanding requests that can exist at any given time. • The default value is 1000.
Step 13	**interface** *type slot / port* **Example:** `Router(config)# interface FastEthernet 0/0`	Configures an interface type and enters interface configuration mode
Step 14	**ip inspect inspection-name** {**in** | **out**} **Example:** `Router(config-if)# ip inspect inspection-name out`	Applies a set of inspection rules to an interface. • 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, page 8

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:

Command or Action	Purpose
enable `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
show ip urlfilter cache	Displays the destination IP addresses that are cached into the cache table.

Command or Action	Purpose
Router# show ip urlfilter cache	
show ip urlfilter config 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 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:

Command or Action	Purpose
enable Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
clear ip urlfilter cache {*ip-address* | **all**} Router# clear ip urlfilter cache all	Clears the cache table.

Monitoring the URL Filter Subsytems

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

Command or Action	Purpose
enable	Enables privileged EXEC mode.

Command or Action	Purpose
	• Enter your password if prompted.
`Router> enable`	

debug ip urlfilter [**function-trace** | **detailed** | **events**]
`Router#`
`debug ip urlfilter detailed`
	Enables debugging information of URL filter subsystems.
	• **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

Example URL Filter Client Firewall Configuration

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

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Websense URL filtering information	*Firewall Websense URL Filtering*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	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[1]	Title
RFC 1945	*Hypertext Transfer Protocol -- HTTP/1.0*
RFC 2616	*Hypertext Transfer Protocol -- HTTP/1.1*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these	http://www.cisco.com/cisco/web/support/index.html

1 **Not all supported RFCs are listed.**

Description	Link
resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	

Feature Information for Firewall N2H2 Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Firewall N2H2 Support

Feature Name	Releases	Feature Information
Firewall N2H2 Support	12.2(11)YU 12.2(15)T	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). The following commands were introduced or modified: **clear ip urlfilter cache**, **debug ip urlfilter**, **ip inspect name**, **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**

Feature Name	Releases	Feature Information
		urlfilter server vendor, **ip urlfilter urlf-server-log**, **show ip urlfilter cache**, **show ip urlfilter config**, **show ip urlfilter statistics**.

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.