Cisco IOS Firewall MIB

First Published: February 27, 2006
Last Updated: February 27, 2006

The Cisco IOS Firewall MIB feature introduces support for the Cisco Unified Firewall MIB, which helps to manage and monitor firewall performance via Simple Network Management Protocol (SNMP). Statistics can be collected and monitored via standards-based SNMP techniques for firewall features such as stateful packet inspection and URL filtering.

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Cisco IOS Firewall MIB" section on page 34.

Finding Support Information for Platforms and Cisco IOS Software Images

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

Contents

- Prerequisites, page 2
- Restrictions for Cisco IOS Firewall MIB, page 2
- Information About Cisco IOS Firewall MIB, page 2
- How to Use Firewall MIBs, page 6
- Configuration Examples for Cisco IOS Firewall MIB Monitoring, page 9
- Additional References, page 16
- Command Reference, page 17

CISCO SYSTEMS

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

Prerequisites

Before you can provide firewall connection and URL filtering statistics via SNMP, you must set up the firewall by performing the following tasks:

- Configure a firewall policy via the **ip inspect name** command.
- Enable the firewall by applying the firewall on a target via the **interface** command followed by the **ip inspect** command.
- Enable URL filtering, if applicable, via the **ip urlfilter server vendor** command.

You must also enable SNMP on the router. For more information on enabling SNMP, see the section "Enabling SNMP for Firewall Sessions" later in this document.

Restrictions for Cisco IOS Firewall MIB

- Cisco does not support all of the MIB variables that are defined in the Cisco Unified Firewall MIB. For a list of variables that are supported by this feature, see Table 1, Table 2, and Table 3.
- MIB statistics are not provided when the firewall is configured using CPL.

Memory and Performance Impact

Depending on the number of targets that have a configured firewall and the number of configured URL filtering servers, the MIB functionality can create an adverse impact on memory. For each firewall policy that is configured on your system, more memory is required to store SNMP statistics.

The following information defines the minimum memory requirements for connection statistics only:

- Global connection statistics: approximately 64 bytes.
- Protocol-specific statistics: multiply the number of configured protocols by 56 to determine the minimum memory requirement.
- Policy-target-protocol statistics: multiply the number of configured protocols and the number of targets for which the firewall policies are configured by 48 to determine the minimum memory requirement.

The following information defines the minimum memory requirements for URL filtering statistics only:

- Global URL filtering statistics: approximately 96 bytes.
- URL filtering server-specific statistics: multiply the number of configured URL filtering servers by 40 to determine the minimum memory requirement.

Information About Cisco IOS Firewall MIB

To use Cisco IOS Firewall MIBs to monitor firewall performance, you should understand the following concepts:

- Connection Statistics, page 3
- URL Filtering Statistics, page 4

Connection Statistics

Connection statistics are a record of the firewall traffic streams that have attempted to flow through the firewall system. Connection statistics can be displayed on a global basis (that is, an aggregate of all connection statistics for the entire router), protocol-specific basis, or a firewall-policy-specific basis. The Firewall can allow, drop, or deny the connection based on firewall policies and firewall resources.

Table 1 lists all supported connection statistics—global, protocol-specific[1], or firewall-policy-specific[2]—that are available via SNMP.

Table 1 Connection Statistics

Statistic Type	Connection Type	Description
• Global • Protocol-specific • Firewall-policy-specific	Aborted	Number of connections that were abnormally terminated after successful establishment
• Global • Protocol-specific • Firewall-policy-specific	Active	Number of connections that are currently active
• Global • Protocol-specific • Firewall-policy-specific	Attempted	Number of connection attempts sent to the firewall system
Global	Embryonic	Number of embryonic-application-layer connections
Global	Expired	Number of connections that were active but have since been terminated normally
• Global • Protocol-specific	Five-Minute Connection Rate	Number of connection attempts that were established per second, averaged over the last 300 seconds
• Global • Protocol-specific • Firewall-policy-specific	Half-Open	Number of connections that are currently in the process of being established (half-open)
• Global • Protocol-specific	One-Minute Connection Rate	Number of connection attempts that were establish per second, averaged over the last 60 seconds

1. All protocol-based statistics can be accessed with the following index—protocol, which is the protocol of interest such as ICMP, UDP, TCP, HTTP, and FTP. The protocols, which are a predefined static list, must be specified
2. All firewall-policy-specific statistics can be accessed with the following indexes: Policy, which is the name of the firewall security policy of interest. (The policy name is specified via the ip inspect name command.) Policy target type, which is the type of physical or virtual target that has the policy name applied to it. Currently, only include interface targets are supported.

Table 1 Connection Statistics (continued)

Statistic Type	Connection Type	Description
• Global • Protocol-specific • Firewall-policy-specific	Policy Declined	Number of connection attempts that were declined due to application of a firewall security policy
• Global • Protocol-specific • Firewall-policy-specific	Resource Declined	Number of connection attempts that were declined due to firewall resource constraints

URL Filtering Statistics

URL Filtering 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. URL filtering statistics include the status of distinct URL filtering servers that are configured on the firewall and the impact of the performance of the URL filtering servers on the latency and throughput of the firewall.

Table 2 and Table 3 list all supported URL filtering statistics—on a global basis or per server—that are available via SNMP.

Table 2 Global URL Filtering Statistics (across all servers)

Connection Type	Description
Five minute URL Filtering Requests Declined Rate	Rate at which URL access requests were declined by the firewall via the URL filtering server or the firewall exclusive domain configuration, averaged over the last 300 seconds.
Five minute URL Filtering Requests Resource Dropped Rate	Rate at which URL access requests were dropped by the firewall due to firewall resource constraints, averaged over the last 300 seconds.
One minute URL Filtering Requests Declined Rate	Rate at which URL access requests were declined by the firewall via the URL filtering server or the firewall exclusive domain configuration, averaged over the last 60 seconds.
One minute URL Filtering Requests Resource Dropped Rate	Rate at which URL access requests were dropped by the firewall due to firewall resource constraints, averaged over the last 60 seconds.
URL Filtering Allow Mode On	Displays whether the firewall has allowed or discarded URL requests when the URL filtering server is not available. Returns a “true” statistics if the firewall allows all requested URLs to be retrieved from the remote host when the URL server is not available; returns a “false” statistic of the firewall discards all URL.
URL Filtering Allow Mode Requests Allowed	Number of URL access requests that were allowed by the firewall when the URL filtering server was not available.

Table 2 Global URL Filtering Statistics (across all servers) (continued)

Connection Type	Description
URL Filtering Allow Mode Requests Denied	Number of URL access requests that were denied by the firewall when the URL filtering server was not available.
URL Filtering Enabled	Displays whether or not URL filtering is enabled. Returns a “false” statistic if the firewall will not perform URL filtering, even if the system contains configuration information that pertains to other aspects of URL filtering.
URL Filtering Late Responses	Number of responses from the URL filtering server that were received after the original URL access request was dropped by the Firewall.
URL Filtering Requests Allowed	Number of URL access requests allowed by the firewall via the use of the URL filtering server or the firewall exclusive domain configuration.
URL Filtering Requests Declined	Number of URL access requests that were declined by the firewall via the URL filtering server or the firewall exclusive domain configuration.
URL Filtering Requests Processed	Number of URL access requests that were processed by the firewall.
URL Filtering Request Process Rate	Number of URL access requests that were processed per second by the firewall, averaged over the last 300 seconds.
URL Filtering Requests Resource Dropped	Number of incoming URL access requests that were dropped by the Firewall due to firewall resource constraints.
URL Filtering Responses Resource Dropped	Number of responses to URL access requests from remote hosts that were dropped by the firewall due to resource constraints while the firewall was waiting for a response from the URL filtering server.
URL Filtering Server Timeouts	Number of times the firewall did not receive a response from the URL Filtering server.

Table 3 Per server URL Filtering Statistics

Connection Type	Description
URL Filtering Protocol Version	Version of the transport protocol that is used by the firewall to communicate with the URL filtering server. For TCP, valid version values are 1 and 4. For UDP, 1 is the only valid version.
URL Filtering Server Late Responses	Number of URL access responses received by the firewall from the URL filtering server after the original URL access request was dropped by the firewall.

Table 3 Per server URL Filtering Statistics (continued)

Connection Type	Description
URL Filtering Server Requests	Number of URL access requests forwarded by the firewall to the URL filtering server.
URL Filtering Server Requests Allowed	Number of URL access requests allowed by the URL filtering server. The count does not include late responses.
URL Filtering Server Requests Declined	Number of URL access requests declined by the URL filtering server. The count does not include late responses.
URL Filtering Server Responses	Number of URL access responses received by the firewall from the URL filtering server. The count does not include late responses.
URL Filtering Server Response Time Rate	Average round-trip response time of the URL filtering server, averaged over the last 300 seconds. A value of zero indicates that there was insufficient data to compute this value over the last time interval.
URL Filtering Server Status	Status of the URL filtering server: ONLINE or OFFLINE.
URL Filtering Server Timeouts	Number of times the URL filtering server failed to respond to URL access requests sent by the firewall.
URL Filtering Server Transport Protocol	Transport protocol that is used by the firewall to communicate with the URL filtering server. The protocol will be TCP, UDP, or DEFAULT. DEFAULT is used in implementations that do not explicitly specify a transport protocol.
URL Filtering Server Vendor	Vendor who provided the URL filtering server. Currently only Websense and N2H2 servers are supported.

A URL filtering server is identified by the following items, which also form the indexes into the URL filtering server statistics table:

- URL Filtering Server Address Type—Type of IP address of the URL filtering server. For example, IPv4 or IPv6.
- URL Filtering Server Address—IP address of the URL filtering server.
- URL Filtering Server Port—Port number that the URL filtering server uses to receive filtering requests.

How to Use Firewall MIBs

This section contains the following task:

- Enabling SNMP for Firewall Sessions, page 7
- Verifying Firewall Connection and URL Filtering Statistics, page 8

Enabling SNMP for Firewall Sessions

Use this task to enable SNMP for firewall-related session management.

Prerequisites

Before you can begin monitoring firewall performance via SNMP, you must set up the firewall by performing the following tasks:

- Configure a firewall policy via the **ip inspect name** command.

Note Statistics are collected only for protocols that are specified via the **ip inspect name** command.

- Enable the firewall by applying the firewall on a target via the **interface** command followed by the **ip inspect** command.
- Enable URL filtering, if applicable, via the **ip urlfilter server vendor** command.

Firewall MIB Traps

To receive firewall MIB traps, you need a management station, and you must enable the **snmp-server enable trap firewall serverstatuschange** command (as shown in the configuration task table below).

Output for the SNMP trap fields, which are displated in on the management station, are as follows:

- Server IP Address Type (IPv4 or IPv6)
- Server IP Address Type Length. (4 for IPv4 and 16 for IPv6)
- Server IP Address
- Server Port

Note Only IPv4 is currently supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string*
4. **snmp-server host** *hostname community-string*
5. **snmp-server enable traps firewall** [**serverstatuschange**]

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	**snmp-server community** *string* **Example:** `Router(config)# snmp-server community public`	Sets up the community access string to permit access to the SNMP.
Step 4	**snmp-server host** *hostname community-string* **Example:** `Router(config)# snmp-server host 192.168.1.1 version 2c public`	Specifies the recipient of the firewall-related SNMP notifications.
Step 5	**snmp-server enable traps firewall** [**serverstatuschange**] **Example:** `Router(config)#  snmp-server enable traps firewall serverstatuschange`	Enables firewall-related SNMP notifications.

What to Do Next

After the firewall and SNMP have been properly enabled, statistics will begin to accumulate after the traffic flow starts. To verify whether statistics are being collected and view MIB counters, you can perform at least one of the steps in the task "Verifying Firewall Connection and URL Filtering Statistics."

Verifying Firewall Connection and URL Filtering Statistics

Use this task to verify firewall connection and URL filtering statistics via command-line interface (CLI). (These statistics can also be collected via any SNMP-capable client.)

SUMMARY STEPS

1. **enable**
2. **show ip inspect mib connection-statistics** {**global** | **l4-protocol** {**all** | **icmp** | **tcp** | **udp**} | **l7-protocol** {**all** | **other** | **telnet** | **ftp**} | **policy** *policy-name* **target** *target name* {**l4-protocol** {**all** | **icmp** | **tcp** | **udp**} | **l7-protocol** {**all** | **other** | **telnet** | **ftp**}}
3. **show ip urlfilter** [**mib**] **statistics** {**global** | *server* {*ip-address* [*port*] | **all**}}]
4. **debug ip inspect mib** {**object-creation** | **object-deletion** | **events** | **retrieval** | **update**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip inspect mib connection-statistics** {**global** | **l4-protocol** {**all** | **icmp** | **tcp** | **udp**} | **l7-protocol** {**all** | **other** | **telnet** | **ftp**} | **policy** *policy-name* **target** *target name* {**l4-protocol** {**all** | **icmp** | **tcp** | **udp**} | **l7-protocol** {**all** | **other** | **telnet** | **ftp**}} **Example:** `Router# show ip inspect mib connection-statistics global`	Displays firewall performance summary statistics that are monitored via SNMP. • **global**—Provides global connection statistics. • **l4-protocol**—Provides Layer 4 statistics for a specified protocol. • **l7-protocol**—Provides Layer 7 statistics for a specified protocol. • **policy** *policy-name* **target** *target-name*—Provides statistics on a per-policy target basis. For example, per firewall policy name and the interface on which the firewall is configured.
Step 3	**show ip urlfilter** [**mib**] **statistics** [{**global** | *server* {*ip-address* [*port*] | **all**}}] **Example:** `Router# show ip urlfilter mib statistics global`	Displays URL filtering statistics for firewall-related MIB events.
Step 4	**debug ip inspect mib** {**object-creation** | **object-deletion** | **events** | **retrieval** | **update**} **Example:** `Router# debug ip inspect mib events`	Displays messages about firewall MIB events.

Troubleshooting Tips

All statistics are accumulated since the last reboot of the firewall system. Thus, you must reboot the system to clear MIB connection statistics from your system.

Configuration Examples for Cisco IOS Firewall MIB Monitoring

This section contains the following examples:

- Sample Cisco IOS Firewall Configuration: Example, page 10
- Sample URL Filtering Configuration: Example, page 12
- show ip inspect mib Output: Examples, page 14
- show ip urlfilter mib statistics command output: Examples, page 15

Sample Cisco IOS Firewall Configuration: Example

The following output from the show running-config command shows how to configure a Cisco IOS Firewall:

```
Router# show running-config

Building configuration...

Current configuration : 2205 bytes
!
version 12.4
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
no logging console
!
no aaa new-model
!
resource policy
!
clock timezone MST -8
clock summer-time MDT recurring
no ip cef
!
!
!
!
ip inspect name test tcp
ip inspect name test udp
ip inspect name test icmp timeout 30
ip inspect name test ftp
ip inspect name test http
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
policy-map ratelimit
class class-default
police cir 10000000
conform-action transmit
exceed-action drop
```

```
!
!
!
!
!
!
!
interface FastEthernet0/0
ip address 192.168.27.2 255.255.255.0
ip access-group 101 out
ip inspect test in
duplex full
service-policy input ratelimit
!
interface FastEthernet1/0
no ip address
no ip route-cache
shutdown
duplex half
!
interface FastEthernet4/0
ip address 192.168.127.2 255.255.255.0
ip access-group 102 in
duplex full
service-policy input ratelimit
!
router eigrp 100
network 192.168.27.0
network 192.168.127.0
no auto-summary
no eigrp log-neighbor-changes
no eigrp log-neighbor-warnings
!
ip default-gateway 192.168.27.116
ip route 192.168.100.0 255.255.255.0 192.168.27.1
ip route 192.168.200.0 255.255.255.0 192.168.127.1
no ip http server
no ip http secure-server
!
!
!
logging alarm informational
access-list 101 permit tcp any any fragments
access-list 101 permit udp any any fragments
access-list 101 deny tcp any any
access-list 101 deny udp any any
access-list 101 permit ip any any
access-list 102 permit tcp any any fragments
access-list 102 permit udp any any fragments
access-list 102 permit udp any gt 1024 any eq snmp
access-list 102 deny tcp any any
access-list 102 deny udp any any
access-list 102 permit ip any any
snmp-server community public RO
snmp-server location FW Testbed UUT
snmp-server contact STG/IOS FW Devtest
!
!
!
!
!
!
control-plane
!
```

```
!
!
!
!
!
gatekeeper
shutdown
!
!
line con 0
exec-timeout 0 0
stopbits 1
line aux 0
stopbits 1
line vty 0 4
login
!
exception core-file sisu-devtest/coredump/Router.core
exception dump 192.168.27.116
!
end
```

Sample URL Filtering Configuration: Example

The following sample output from the show running-config command shows how to configure a Websense server for URL filtering:

```
Router# show running-config
Building configuration...

Current configuration : 2043 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service internal
!
hostname Router
!
boot-start-marker
boot-end-marker
!
no logging console
!
no aaa new-model
!
resource policy
!
clock timezone MST -8
clock summer-time MDT recurring
no ip cef
!
!
ip inspect name test tcp
ip inspect name test udp
ip inspect name test http urlfilter
!
!
ip urlfilter allow-mode on
ip urlfilter exclusive-domain deny www.cnn.com
ip urlfilter exclusive-domain permit www.cpp.com
```

```
ip urlfilter server vendor websense 192.168.29.116
!
!
!
!
!
!
!
!
!
interface FastEthernet0/0
ip address 192.168.29.2 255.255.255.0
ip access-group 101 out
ip inspect test in
speed auto
full-duplex
!
interface FastEthernet0/1
ip address 192.168.129.2 255.255.255.0
ip access-group 102 in
duplex auto
speed auto
!
router eigrp 100
network 192.168.29.0
network 192.168.129.0
no auto-summary
no eigrp log-neighbor-changes
no eigrp log-neighbor-warnings
!
ip default-gateway 192.168.28.116
ip route 192.168.100.0 255.255.255.0 192.168.29.1
ip route 192.168.200.0 255.255.255.0 192.168.129.1
!
!
ip http server
no ip http secure-server
!
access-list 101 permit tcp any any fragments
access-list 101 permit udp any any fragments
access-list 101 deny tcp any any
access-list 101 deny udp any any
access-list 101 permit ip any any
access-list 102 permit tcp any any fragments
access-list 102 permit udp any any fragments
access-list 102 permit udp any gt 1024 any eq snmp
access-list 102 deny tcp any any
access-list 102 deny udp any any
access-list 102 permit ip any any
snmp-server community public RO
snmp-server location FW Testbed UUT
snmp-server contact STG/IOS FW Devtest
!
!
!
!
control-plane
!
!
!
line con 0
exec-timeout 0 0
transport output all
line aux 0
```

```
transport output all
line vty 0 4
login
!
exception core-file sisu-devtest/coredump/Router.core
exception dump 192.168.28.116
!
webvpn context Default_context
ssl authenticate verify all
!
no inservice
!
!
end
```

show ip inspect mib Output: Examples

The following examples are sample outputs from the **show ip inspect mib** command with global or protocol-specific keywords:

- Global MIB Statistics, page 14
- Protocol-Based MIB Statistics, page 14
- Policy-Target-Based MIB Statistics, page 15

Global MIB Statistics

```
Router# show ip inspect mib connection-statistics global
------------------------------------------------
Connections Attempted 7
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 2 Connections Active 3
Connections Expired 2
Connections Aborted 0
Connections Embryonic 0
Connections 1-min Setup Rate 5
Connections 5-min Setup Rate 7
```

Protocol-Based MIB Statistics

```
Router# show ip inspect mib connection-statistics l4-protocol tcp
------------------------------------------------
Protocol tcp
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 1
Connections Active 2
Connections Aborted 0
Connections 1-min Setup Rate 3
Connections 5-min Setup Rate 3
```

```
Router# show ip inspect mib connection-statistics l7-protocol http
------------------------------------------------
Protocol http
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 2
Connections Resource Declined 0
Connections Half Open 0
Connections Active 1
Connections Aborted 0
Connections 1-min Setup Rate 1
Connections 5-min Setup Rate 2
```

Policy-Target-Based MIB Statistics

```
Router# show ip inspect mib connection-statistics policy ftp interface GigabitEthernet0/0
l4-protocol tcp
! Policy Target Protocol Based Connection Summary Stats
------------------------------------------------------
Policy ftp-inspection
Target GigabitEthernet0/0
Protocol tcp
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 1
Connections Active 2
Connections Aborted 0

Router# show ip inspect mib connection-statistics policy ftp interface GigabitEthernet0/0
l7-protocol ftp
! Policy Target Protocol Based Connection Summary Stats
------------------------------------------------------
Policy ftp-inspection
Target GigabitEthernet0/0
Protocol ftp
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 1
Connections Active 2
Connections Aborted 0
```

show ip urlfilter mib statistics command output: Examples

The following example is sample output when MIBs are enabled to track URL filtering statistics across the entire device (global):

```
Router# show ip urlfilter mib statistics global
URL Filtering Group Summary Statistics
------------------------------------------------------
URL Filtering Enabled
Requests Processed 260
Requests Processed 1-minute Rate 240
Requests Processed 5-minute Rate 215
Requests Allowed 230
Requests Denied 30
Requests Denied 1-minute Rate 15
Requests Denied 5-minute Rate 0
```

```
Requests Cache Allowed 5
Requests Cache Denied 5
Allow Mode Requests Allowed 15
Allow Mode Requests Denied 15
Requests Resource Dropped 0
Requests Resource Dropped 1-minute Rate 0
Requests Resource Dropped 5-minute Rate 0
Server Timeouts 0
Server Retries 0
Late Server Responses 0
Access Responses Resource Dropped 0
```

The following example is sample output when MIBs are enabled to track URL filtering statistics across the server with IP address 192.168.27.116:

```
Router# show ip urlfilter mib statistics server address 192.168.27.116
URL Filtering Server Statistics
------------------------------------------------------
URL Server Host Name 192.168.27.116
Server Address 192.168.27.116
Server Port 15868
Server Vendor Websense
Server Status Online
Requests Processed 4
Requests Allowed 1
Requests Denied 3
Server Timeouts 0
Server Retries 9
Responses Received 1
Late Server Responses 12
1 Minute Average Response Time 0
5 Minute Average Response Time 0
```

Additional References

The following sections provide references related to Cisco IOS Firewall MIB.

Related Documents

Related Topic	Document Title
Description of SNMP, SNMP MIBs, and how to configure SNMP on Cisco devices	"Configuring SNMP Support" in the *Cisco IOS Network Management Configuration Guide*, Release 12.4
Description of Cisco IOS firewalls and functions such as how to configure a firewall and URL filtering	"Configuring Context-based Access Control" in the *Cisco IOS Security Configuration Guide*, Release 12.4

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
• CISCO-UNIFIED-FIREWALL-MIB.my • CISCO-FIREWALL-TC.my	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

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of 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/techsupport

Command Reference

This section documents new and modified commands.

New Command

- **snmp-server enable traps firewall**

Modified Commands

- **debug ip inspect**
- **show ip inspect**
- **show ip urlfilter statistics**

debug ip inspect

Note Effective with Cisco IOS Release 12.4(20)T, the **debug ip inspect** command is replaced by the **debug policy-firewall** command. See the **debug policy-firewall** command for more information.

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

debug ip inspect {**function-trace** | **object-creation** | **object-deletion** | **events** | **timers** | *protocol* | **detailed** | **update**}

Firewall MIB Statistics Syntax

debug ip inspect mib {**object-creation** | **object-deletion** | **events** | **retrieval** | **update**}

no debug ip inspect

Syntax Description

mib	(Optional) Displays messages about MIB functionality.
function-trace	Displays messages about software functions called by the Cisco IOS Firewall.
object-creation	Displays messages about software objects being created by the Cisco IOS Firewall. Object creation corresponds to the beginning of Cisco IOS Firewall-inspected sessions.
object-deletion	Displays messages about software objects being deleted by the Cisco IOS Firewall. Object deletion corresponds to the closing of Cisco IOS Firewall-inspected sessions.
events	Displays messages about Cisco IOS Firewall software events, including information about Cisco IOS Firewall packet processing or MIB special events.
timers	Displays messages about Cisco IOS Firewall timer events such as when the Cisco IOS Firewall idle timeout is reached.
protocol	Displays messages about Cisco IOS Firewall-inspected protocol events, including details about the packets of the protocol. Table 4 provides a list of *protocol* keywords.
detailed	Displays detailed information to be displayed for all the other enabled Cisco IOS Firewall debugging. Use this form of the command in conjunction with other Cisco IOS Firewall debug commands.
retrieval	Displays messages of statistics requested via Simple Network Management Protocol (SNMP) or command-line interface (CLI).
update	Displays messages about Cisco IOS Firewall software updates or updates to MIB counters.

Table 4 Protocol Keywords for the debug ip inspect Command

Application Protocol	Protocol Keyword
Transport-layer protocols	
ICMP	icmp
TCP	tcp
User Datagram Protocol (UDP)	udp
Application-layer protocols	
CU-SeeMe	cuseeme
FTP commands and responses	ftp-cmd
FTP tokens (enables tracing of the FTP tokens parsed)	ftp-tokens
H.323 (version 1 and version 2)	h323
HTTP	http
IMAP	imap
Microsoft NetShow	netshow
POP3	pop3
RealAudio	realaudio
Remote procedure call (RPC)	rpc
Real Time Streaming Protocol (RTSP)	rtsp
Session Initiation Protocol (SIP)	sip
Simple Mail Transfer Protocol (SMTP)	smtp
Skinny Client Control Protocol (SCCP)	skinny
Structured Query Language*Net (SQL*Net)	sqlnet
StreamWorks	streamworks
TFTP	tftp
UNIX r-commands (rlogin, rexec, rsh)	rcmd
VDOLive	vdolive

Command Modes

Privileged EXEC

Command History

Release	Modification
11.2 P	This command was introduced.
12.0(5)T	NetShow support was added.
12.0(7)T	H.323 V2 and RTSP protocol support were added.
12.2(11)YU	Support for the ICMP and SIP protocols was added.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.3(1)	Support for the skinny protocol was added.

Release	Modification
11.2 P	This command was introduced.
12.3(14)T	Support for the IMAP and POP3 protocols was added.
12.4(6)T	The MIB syntax was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	This command was replaced by the **debug policy-firewall** command.

Examples

The following is sample output from the **debug ip inspect function-trace** command:

```
Router# debug ip inspect function-trace

*Mar  2 01:16:16: CBAC FUNC: insp_inspection
*Mar  2 01:16:16: CBAC FUNC: insp_pre_process_sync
*Mar  2 01:16:16: CBAC FUNC: insp_find_tcp_host_entry addr 40.0.0.1 bucket 41
*Mar  2 01:16:16: CBAC FUNC: insp_find_pregen_session
*Mar  2 01:16:16: CBAC FUNC: insp_get_idbsb
*Mar  2 01:16:16: CBAC FUNC: insp_get_idbsb
*Mar  2 01:16:16: CBAC FUNC: insp_get_irc_of_idb
*Mar  2 01:16:16: CBAC FUNC: insp_get_idbsb
*Mar  2 01:16:16: CBAC FUNC: insp_create_sis
*Mar  2 01:16:16: CBAC FUNC: insp_inc_halfopen_sis
*Mar  2 01:16:16: CBAC FUNC: insp_link_session_to_hash_table
*Mar  2 01:16:16: CBAC FUNC: insp_inspect_pak
*Mar  2 01:16:16: CBAC FUNC: insp_l4_inspection
*Mar  2 01:16:16: CBAC FUNC: insp_process_tcp_seg
*Mar  2 01:16:16: CBAC FUNC: insp_listen_state
*Mar  2 01:16:16: CBAC FUNC: insp_ensure_return_traffic
*Mar  2 01:16:16: CBAC FUNC: insp_add_acl_item
*Mar  2 01:16:16: CBAC FUNC: insp_ensure_return_traffic
*Mar  2 01:16:16: CBAC FUNC: insp_add_acl_item
*Mar  2 01:16:16: CBAC FUNC: insp_process_syn_packet
*Mar  2 01:16:16: CBAC FUNC: insp_find_tcp_host_entry addr 40.0.0.1 bucket 41
*Mar  2 01:16:16: CBAC FUNC: insp_create_tcp_host_entry
*Mar  2 01:16:16: CBAC* FUNC: insp_fast_inspection
*Mar  2 01:16:16: CBAC* FUNC: insp_inspect_pak
*Mar  2 01:16:16: CBAC* FUNC: insp_l4_inspection
*Mar  2 01:16:16: CBAC* FUNC: insp_process_tcp_seg
*Mar  2 01:16:16: CBAC* FUNC: insp_synrcvd_state
*Mar  2 01:16:16: CBAC* FUNC: insp_fast_inspection
*Mar  2 01:16:16: CBAC* FUNC: insp_inspect_pak
*Mar  2 01:16:16: CBAC* FUNC: insp_l4_inspection
*Mar  2 01:16:16: CBAC* FUNC: insp_process_tcp_seg
*Mar  2 01:16:16: CBAC* FUNC: insp_synrcvd_state
*Mar  2 01:16:16: CBAC FUNC: insp_dec_halfopen_sis
*Mar  2 01:16:16: CBAC FUNC: insp_remove_sis_from_host_entry
*Mar  2 01:16:16: CBAC FUNC: insp_find_tcp_host_entry addr 40.0.0.1 bucket 41
```

This output shows the functions called by the Cisco IOS Firewall as a session is inspected. Entries with an asterisk (*) after the word “CBAC” are entries when the fast path is used; otherwise, the process path is used.

The following is sample output from the **debug ip inspect object-creation** and **debug ip inspect object-deletion** commands:

```
Router# debug ip inspect object-creation
Router# debug ip inspect object-deletion

*Mar  2 01:18:30: CBAC OBJ_CREATE: create pre-gen sis 25A3574
*Mar  2 01:18:30: CBAC OBJ_CREATE: create acl wrapper 25A36FC -- acl item 25A3634
```

```
*Mar  2 01:18:30: CBAC OBJ_CREATE: create sis 25C1CC4
*Mar  2 01:18:30: CBAC OBJ_DELETE: delete pre-gen sis 25A3574
*Mar  2 01:18:30: CBAC OBJ_CREATE: create host entry 25A3574 addr 10.0.0.1 bucket 31
*Mar  2 01:18:30: CBAC OBJ_DELETE: delete sis 25C1CC4
*Mar  2 01:18:30: CBAC OBJ_DELETE: delete create acl wrapper 25A36FC -- acl item 25A3634
*Mar  2 01:18:31: CBAC OBJ_DELETE: delete host entry 25A3574 addr 10.0.0.1
```

The following is sample output from the **debug ip inspect object-creation**, **debug ip inspect object-deletion**, and **debug ip inspect events** commands:

```
Router# debug ip inspect object-creation
Router# debug ip inspect object-deletion
Router# debug ip inspect events

*Mar  2 01:18:51: CBAC OBJ_CREATE: create pre-gen sis 25A3574
*Mar  2 01:18:51: CBAC OBJ_CREATE: create acl wrapper 25A36FC -- acl item 25A3634
*Mar  2 01:18:51: CBAC Src 10.1.0.1 Port [1:65535]
*Mar  2 01:18:51: CBAC Dst 10.0.0.1 Port [46406:46406]
*Mar  2 01:18:51: CBAC Pre-gen sis 25A3574 created: 10.1.0.1[1:65535]
30.0.0.1[46406:46406]
*Mar  2 01:18:51: CBAC OBJ_CREATE: create sis 25C1CC4
*Mar  2 01:18:51: CBAC sis 25C1CC4 initiator_addr (10.1.0.1:20) responder_addr
(30.0.0.1:46406) initiator_alt_addr (40.0.0.1:20) responder_alt_addr (10.0.0.1:46406)
*Mar  2 01:18:51: CBAC OBJ_DELETE: delete pre-gen sis 25A3574
*Mar  2 01:18:51: CBAC OBJ_CREATE: create host entry 25A3574 addr 10.0.0.1 bucket 31
*Mar  2 01:18:51: CBAC OBJ_DELETE: delete sis 25C1CC4
*Mar  2 01:18:51: CBAC OBJ_DELETE: delete create acl wrapper 25A36FC -- acl item 25A3634
*Mar  2 01:18:51: CBAC OBJ_DELETE: delete host entry 25A3574 addr 10.0.0.1
```

The following is sample output from the **debug ip inspect timers** command:

```
Router# debug ip inspect timers

*Mar  2 01:19:15: CBAC Timer Init Leaf: Pre-gen sis 25A3574
*Mar  2 01:19:15: CBAC Timer Start: Pre-gen sis 25A3574 Timer: 25A35D8 Time: 30000
milisecs
*Mar  2 01:19:15: CBAC Timer Init Leaf: sis 25C1CC4
*Mar  2 01:19:15: CBAC Timer Stop: Pre-gen sis 25A3574 Timer: 25A35D8
*Mar  2 01:19:15: CBAC Timer Start: sis 25C1CC4 Timer: 25C1D5C Time: 30000 milisecs
*Mar  2 01:19:15: CBAC Timer Start: sis 25C1CC4 Timer: 25C1D5C Time: 3600000 milisecs
*Mar  2 01:19:15: CBAC Timer Start: sis 25C1CC4 Timer: 25C1D5C Time: 5000 milisecs
*Mar  2 01:19:15: CBAC Timer Stop: sis 25C1CC4 Timer: 25C1D5C
```

The following is sample output from the **debug ip inspect tcp** command:

```
Router# debug ip inspect tcp

*Mar  2 01:20:43: CBAC* sis 25A3604 pak 2541C58 TCP P ack 4223720032 seq 4200176225(22)
(10.0.0.1:46409) => (10.1.0.1:21)
*Mar  2 01:20:43: CBAC* sis 25A3604 ftp L7 inspect result: PROCESS-SWITCH packet
*Mar  2 01:20:43: CBAC sis 25A3604 pak 2541C58 TCP P ack 4223720032 seq 4200176225(22)
(10.0.0.1:46409) => (10.1.0.1:21)
*Mar  2 01:20:43: CBAC sis 25A3604 ftp L7 inspect result: PASS packet
*Mar  2 01:20:43: CBAC* sis 25A3604 pak 2544374 TCP P ack 4200176247 seq 4223720032(30)
(10.0.0. 1:46409) <= (10.1.0.1:21)
*Mar  2 01:20:43: CBAC* sis 25A3604 ftp L7 inspect result: PASS packet
*Mar  2 01:20:43: CBAC* sis 25A3604 pak 25412F8 TCP P ack 4223720062 seq 4200176247(15)
(10.0.0. 1:46409) => (10.1.0.1:21)
*Mar  2 01:20:43: CBAC* sis 25A3604 ftp L7 inspect result: PASS packet
*Mar  2 01:20:43: CBAC sis 25C1CC4 pak 2544734 TCP S seq 4226992037(0) (10.1.0.1:20) =>
(10.0.0.1:46411)
*Mar  2 01:20:43: CBAC* sis 25C1CC4 pak 2541E38 TCP S ack 4226992038 seq 4203405054(0)
(10.1.0.1:20) <= (10.0.0.1:46411)
```

This sample shows TCP packets being processed and lists the corresponding acknowledge (ACK) packet numbers and sequence (SEQ) numbers. The number of data bytes in the TCP packet is shown in parentheses—for example, (22). For each packet shown, the addresses and port numbers are shown separated by a colon. For example, (10.1.0.1:21) indicates an IP address of 10.1.0.1 and a TCP port number of 21.

Entries with an asterisk (*) after the word “CBAC” are entries when the fast path is used; otherwise, the process path is used.

The following is sample output from the **debug ip inspect tcp** and **debug ip inspect detailed** commands:

```
Router# debug ip inspect tcp
Router# debug ip inspect detailed

*Mar  2 01:20:58: CBAC* Pak 2541E38 Find session for (30.0.0.1:46409) (40.0.0.1:21) tcp
*Mar  2 01:20:58:  P ack 4223720160 seq 4200176262(22)
*Mar  2 01:20:58: CBAC* Pak 2541E38 Addr:port pairs to match: (30.0.0.1:46409)
(40.0.0.1:21)
*Mar  2 01:20:58: CBAC* sis 25A3604 SIS_OPEN
*Mar  2 01:20:58: CBAC* Pak 2541E38 IP: s=30.0.0.1 (Ethernet0), d=40.0.0.1 (Ethernet1),
len 76,proto=6
*Mar  2 01:20:58: CBAC sis 25A3604 Saving State: SIS_OPEN/ESTAB iisn 4200176160 i_rcvnxt
4223720160 i_sndnxt 4200176262 i_rcvwnd 8760 risn 4223719771 r_rcvnxt 4200176262 r_sndnxt
4223720160 r_rcvwnd 8760
*Mar  2 01:20:58: CBAC* sis 25A3604 pak 2541E38 TCP P ack 4223720160 seq 4200176262(22)
(30.0.0.1:46409) => (40.0.0.1:21)
*Mar  2 01:20:58: CBAC* sis 25A3604 pak 2541E38 SIS_OPEN/ESTAB TCP seq 4200176262(22)
Flags: ACK 4223720160 PSH
*Mar  2 01:20:58: CBAC* sis 25A3604 pak 2541E38 --> SIS_OPEN/ESTAB iisn 4200176160
i_rcvnxt 4223720160 i_sndnxt 4200176284 i_rcvwnd 8760 risn 4223719771 r_rcvnxt 4200176262
r_sndnxt 4223720160 r_rcvwnd 8760
*Mar  2 01:20:58: CBAC* sis 25A3604 L4 inspect result: PASS packet 2541E38
(30.0.0.1:46409) (40.0.0.1:21) bytes 22 ftp
*Mar  2 01:20:58: CBAC sis 25A3604 Restoring State: SIS_OPEN/ESTAB iisn 4200176160
i_rcvnxt 4223
720160 i_sndnxt 4200176262 i_rcvwnd 8760 risn 4223719771 r_rcvnxt 4200176262 r_sndnxt
4223720160 r_rcvwnd 8760
*Mar  2 01:20:58: CBAC* sis 25A3604 ftp L7 inspect result: PROCESS-SWITCH packet
*Mar  2 01:20:58: CBAC* sis 25A3604 ftp L7 inspect result: PROCESS-SWITCH packet
*Mar  2 01:20:58: CBAC* Bump up: inspection requires the packet in the process
path(30.0.0.1) (40.0.0.1)
*Mar  2 01:20:58: CBAC Pak 2541E38 Find session for (30.0.0.1:46409) (40.0.0.1:21) tcp
*Mar  2 01:20:58:  P ack 4223720160 seq 4200176262(22)
*Mar  2 01:20:58: CBAC Pak 2541E38 Addr:port pairs to match: (30.0.0.1:46409)
(40.0.0.1:21)
*Mar  2 01:20:58: CBAC sis 25A3604 SIS_OPEN
*Mar  2 01:20:58: CBAC Pak 2541E38 IP: s=30.0.0.1 (Ethernet0), d=40.0.0.1 (Ethernet1), len
76, proto=6
```

The following is sample output from the **debug ip inspect icmp** and **debug ip inspect detailed** commands:

```
Router# debug ip inspect icmp
Router# debug ip inspect detailed

1w6d:CBAC sis 81073F0C SIS_CLOSED
1w6d:CBAC Pak 80D2E9EC IP:s=192.168.133.3 (Ethernet1), d=0.0.0.0 (Ethernet0), len 98,
proto=1
1w6d:CBAC ICMP:sis 81073F0C pak 80D2E9EC SIS_CLOSED ICMP packet (192.168.133.3:0) =>
(0.0.0.0:0) datalen 56
1w6d:CBAC ICMP:start session from 192.168.133.3
1w6d:CBAC sis 81073F0C --> SIS_OPENING (192.168.133.3:0) (0.0.0.0:0)
1w6d:CBAC sis 81073F0C L4 inspect result:PASS packet 80D2E9EC (192.168.133.3:0)
(0.0.0.0:0) bytes 56 icmp
1w6d:CBAC sis 81073F0C SIS_OPENING
1w6d:CBAC Pak 80E72BFC IP:s=0.0.0.0 (Ethernet0), d=192.168.133.3 (Ethernet1), len 98,
proto=1
1w6d:CBAC ICMP:sis 81073F0C pak 80E72BFC SIS_OPENING ICMP packet (192.168.133.3:0) <=
(0.0.0.0:0) datalen 56
1w6d:CBAC sis 81073F0C --> SIS_OPEN (192.168.133.3:0) (0.0.0.0:0)
```

```
1w6d:CBAC sis 81073F0C L4 inspect result:PASS packet 80E72BFC (0.0.0.0:0)
(192.168.133.3:0) bytes 56 icmp
1w6d:CBAC* sis 81073F0C SIS_OPEN
1w6d:CBAC* Pak 80D2F2C8 IP:s=192.168.133.3 (Ethernet1), d=0.0.0.0 (Ethernet0), len 98,
proto=1
1w6d:CBAC* ICMP:sis 81073F0C pak 80D2F2C8 SIS_OPEN ICMP packet (192.168.133.3:0) =>
(0.0.0.0:0) datalen 56
1w6d:CBAC* sis 81073F0C --> SIS_OPEN (192.168.133.3:0) (0.0.0.0:0)
1w6d:CBAC* sis 81073F0C L4 inspect result:PASS packet 80D2F2C8 (192.168.133.3:0)
(0.0.0.0:0) bytes 56 icmp
1w6d:CBAC* sis 81073F0C SIS_OPEN
1w6d:CBAC* Pak 80E737CC IP:s=0.0.0.0 (Ethernet0), d=192.168.133.3 (Ethernet1), len 98,
proto=1
1w6d:CBAC* ICMP:sis 81073F0C pak 80E737CC SIS_OPEN ICMP packet (192.168.133.3:0) <=
(0.0.0.0:0) datalen 56
1w6d:CBAC* sis 81073F0C --> SIS_OPEN (192.168.133.3:0) (0.0.0.0:0)
1w6d:CBAC* sis 81073F0C L4 inspect result:PASS packet 80E737CC (0.0.0.0:0)
(192.168.133.3:0) bytes 56 icmp
1w6d:CBAC* sis 81073F0C SIS_OPEN
1w6d:CBAC* Pak 80F554F0 IP:s=192.168.133.3 (Ethernet1), d=0.0.0.0 (Ethernet0), len 98,
proto=1
1w6d:CBAC* ICMP:sis 81073F0C pak 80F554F0 SIS_OPEN ICMP packet (192.168.133.3:0) =>
(0.0.0.0:0) datalen 56
1w6d:CBAC* sis 81073F0C --> SIS_OPEN (192.168.133.3:0) (0.0.0.0:0)
1w6d:CBAC* sis 81073F0C L4 inspect result:PASS packet 80F554F0 (192.168.133.3:0)
(0.0.0.0:0) bytes 56 icmp
1w6d:CBAC* sis 81073F0C SIS_OPEN
1w6d:CBAC* Pak 80E73AC0 IP:s=0.0.0.0 (Ethernet0), d=192.168.133.3 (Ethernet1), len 98,
proto=1
1w6d:CBAC* ICMP:sis 81073F0C pak 80E73AC0 SIS_OPEN ICMP packet (192.168.133.3:0) <=
(0.0.0.0:0) datalen 56
1w6d:CBAC* sis 81073F0C --> SIS_OPEN (192.168.133.3:0) (0.0.0.0:0)
1w6d:CBAC* sis 81073F0C L4 inspect result:PASS packet 80E73AC0 (0.0.0.0:0)
(192.168.133.3:0) bytes 56 icmp
```

show ip inspect

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

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

Firewall MIB Statistics Syntax

show ip inspect mib connection-statistics {**global** | **l4-protocol** {**all** | **icmp** | **tcp** | **udp**} | **l7-protocol** {**all** | **other** | **telnet** | **ftp**} | **policy** *policy-name* **target** *target name* {**l4-protocol** {**all** | **icmp** | **tcp** | **udp**} | **l7-protocol** {**all** | **other** | **telnet** | **ftp**}}

Syntax Description

name *inspection-name*	Displays the configured inspection rule with the name *inspection-name*.
config	Displays the complete CBAC or HA inspection configuration.
interfaces	Displays the interface configuration with respect to applied inspection rules and access lists.
session [**detail**]	Displays existing sessions that are currently being tracked and inspected by CBAC or HA. The optional **detail** keyword allows additional details about these sessions to be shown.
statistics	Displays CBAC sessions statistics, such as the number of TCP and HTTP packets that are processed through the inspection, the number of sessions that have been created since the subsystem startup, the current session count, the maximum session count, and the session creation rate.
all	Displays all CBAC configuration and all existing sessions that are currently being tracked and inspected by CBAC.
vrf *vrf-name*	(Optional) Displays information only for the specified Virtual Routing and Forwarding (VRF) interface.
mib connection-statistics	Displays firewall performance summary statistics that are monitored via firewall MIBs.
global	Displays global connection summary statistics, which are kept for the entire device.
l4-protocol	Displays Layer 4 protocol-based connection summary statistics for one of the follwing specified protocols: **all**, **icmp**, **tcp**, **udp**.
l7-protocol	Displays Layer 7 protocol-based connection summary statistics for one of the follwing specified protocols: **all**, **other**, **telnet**, **ftp**.
policy *policy-name*	Name of the firewall policy that is being monitored.
target *target name*	Name of the interface on which the specified firewall policy is applied.

Command Modes

Privileged EXEC

Command History

Release	Modification
11.2 P	This command was introduced.
12.3(4)T	The output for the **show ip inspect session detail** command was enhanced to support dynamic access control list (ACL) bypass.
12.3(11)T	The **statistics** keyword was added.
12.3(14)T	The output shows the IMAP and POP3 configuration. The **vrf** *vrf-name* keyword/argument pair was added.
12.4(6)T	The firewall MIB statistics syntax was added to support firewall performance via SNMP. High Availability (HA) configuration and session information was added to support Stateful Failover.

Usage Guidelines

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

ACL Bypass Functionality

ACL bypass allows a packet to avoid redundant ACL checks by allowing the firewall to permit the packet on the basis of existing inspection sessions instead of dynamic ACLs. Because input and output dynamic ACLs have been eliminated from the firewall configuration, the **show ip inspect session detail** command output no longer shows dynamic ACLs. Instead, the output displays the matching inspection session for each packet that is permitted through the firewall.

Firewall MIB Functionality

The Cisco Unified Firewall MIB monitors the following firewall performance statistics:

- Connection statistics, which are a record of the firewall traffic streams that have attempted to flow through the firewall system. Connection statistics can be displayed on a global basis, a protocol-specific basis, or a firewall policy basis.
- URL filtering statistics, which include the status of distinct URL filtering servers that are configured on the firewall and the impact of the performance of the URL filtering servers on the latency and throughput of the firewall.

Examples

The following example shows sample output for the **show ip inspect name myinspectionrule** command, where the inspection rule "myinspectionrule" is configured. In this example, the output shows the protocols that should be inspected by CBAC and the corresponding idle timeouts for each protocol.

```
Router# show ip inspect name myinspectionrule

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

The following is sample output for the **show ip inspect config** command. In this example, the output shows CBAC configuration, including global timeouts, thresholds, and inspection rules.

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

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

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

The following is sample output for the **show ip inspect session** command. In this example, the output shows the source and destination addresses and port numbers (separated by colons), and it indicates that the session is an FTP session.

```
Router# show ip inspect session

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

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

```
Router# show ip inspect all

Session audit trail is disabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name all
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
Interface Configuration
 Interface Ethernet0
  Inbound inspection rule is all
    tcp timeout 3600
    udp timeout 30
    ftp timeout 3600
  Outgoing inspection rule is not set
  Inbound access list is not set
  Outgoing access list is not set
```

```
 Established Sessions
 Session 25A6E1C (30.0.0.1:46065)=>(40.0.0.1:21) ftp SIS_OPEN
 Session 25A34A0 (40.0.0.1:20)=>(30.0.0.1:46072) ftp-data SIS_OPEN
```

The following is sample output from the **show ip inspect session detail** command, which shows that an outgoing ACL and an inbound ACL (dynamic ACLs) have been created to allow return traffic:

```
Router# show ip inspect session detail

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

The following is sample output from the **show ip inspect session detail** command, which shows related ACL information (such as session identifiers [SIDs]), but does not show dynamic ACLs, which are no longer created:

```
Router# show ip inspect session detail

Established Sessions
 Session 814063CC (192.168.1.116:32955)=>(192.168.101.115:23) tcp SIS_OPEN
  Created 00:00:10, Last heard 00:00:06
  Bytes sent (initiator:responder) [140:298]
  HA state: HA_STANDBY
  In  SID 192.168.101.115[23:23]=>192.168.1.117[32955:32955] on ACL 101 (15 matches)
  Out SID 192.168.101.115[23:23]=>192.168.1.116[32955:32955] on ACL 102
```

The following is sample output from the **show ip inspect statistics** command:

```
Router# show ip inspect statistics

Packet inspection statistics [process switch:fast switch]
  tcp packets: [616668:0]
  http packets: [178912:0]
Interfaces configured for inspection 1
Session creations since subsystem startup or last reset 42940
Current session counts (estab/half-open/terminating) [0:0:0]
Maxever session counts (estab/half-open/terminating) [98:68:50]
Last session created 5d21h
Last statistic reset never
Last session creation rate 0
Last half-open session total 0
```

The following examples are sample outputs from the **show ip inspect mib** command with global or protocol-specific keywords.

Global MIB Statistics

```
Router# show ip inspect mib connection-statistics global
------------------------------------------------
Connections Attempted 7
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 2 Connections Active 3
Connections Expired 2
Connections Aborted 0
Connections Embryonic 0
```

```
Connections 1-min Setup Rate 5
Connections 5-min Setup Rate 7
```

Protocol-Based MIB Statistics

```
Router# show ip inspect mib connection-statistics l4-protocol tcp
------------------------------------------------
Protocol tcp
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 1
Connections Active 2
Connections Aborted 0
Connections 1-min Setup Rate 3
Connections 5-min Setup Rate 3

Router# show ip inspect mib connection-statistics l7-protocol http
------------------------------------------------
Protocol http
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 2
Connections Resource Declined 0
Connections Half Open 0
Connections Active 1
Connections Aborted 0
Connections 1-min Setup Rate 1
Connections 5-min Setup Rate 2
```

Policy-target-Based MIB Statistics

```
Router# show ip inspect mib connection-statistics policy ftp interface GigabitEthernet0/0
l4-protocol tcp
! Policy Target Protocol Based Connection Summary Stats
------------------------------------------------------
Policy ftp-inspection
Target GigabitEthernet0/0
Protocol tcp
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 1
Connections Active 2
Connections Aborted 0

Router# show ip inspect mib connection-statistics policy ftp interface GigabitEthernet0/0
l7-protocol ftp
! Policy Target Protocol Based Connection Summary Stats
------------------------------------------------------
Policy ftp-inspection
Target GigabitEthernet0/0
Protocol ftp
Connections Attempted 3
Connections Setup Aborted 0
Connections Policy Declined 0
Connections Resource Declined 0
Connections Half Open 1
Connections Active 2
Connections Aborted 0
```

show ip urlfilter statistics

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

show ip urlfilter [**mib**] **statistics** [**vrf** *vrf-name*] [{**global** | *server* {*ip-address* [*port*] | **all**}}]

Syntax Description

mib	(Optional) Displays statistics only for firewall MIB events.
vrf *vrf-name*	(Optional) Displays the information only for the specified Virtual Routing and Forwarding (VRF) interface. **Note** The firewall MIB is not yet VRF aware; thus, this option is not supported if the **mib** keyword is used.
global	(Optional) Displays global URL filtering statistics.
server ip-address	(Optional) Displays statistics for the server specified via IP address.
server port	(Optional) Displays statistics for the server specified via IP address and port. **Note** You must issue the *ip-address* argument before issuing the *port* argument.
all	(Optional) Displays statistics for all configured servers.

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.
12.3(14)T	The **vrf** *vrf-name* keyword/argument pair was added.
12.4(6)T	The following keywords and arguments were added: **mib**, **global**, *server*, *ip-address*, *port*, **all**.

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 5 describes the significant fields shown in the display.

Table 5 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 packet buffer of the firewall.
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.

The following example is sample output when MIBs are enabled to track URL filtering statistics across the entire device (global):

```
Router# show ip urlfilter mib statistics global
URL Filtering Group Summary Statistics
------------------------------------------------------
URL Filtering Enabled
Requests Processed 260
Requests Processed 1-minute Rate 240
Requests Processed 5-minute Rate 215
Requests Allowed 230
Requests Denied 30
Requests Denied 1-minute Rate 15
Requests Denied 5-minute Rate 0
Requests Cache Allowed 5
Requests Cache Denied 5
Allow Mode Requests Allowed 15
Allow Mode Requests Denied 15
Requests Resource Dropped 0
Requests Resource Dropped 1-minute Rate 0
Requests Resource Dropped 5-minute Rate 0
Server Timeouts 0
Server Retries 0
Late Server Responses 0
Access Responses Resource Dropped 0
```

The following example is sample output when MIBs are enabled to track URL filtering statistics across the server with IP address 192.168.27.116:

```
Router# show ip urlfilter mib statistics server address 192.168.27.116
URL Filtering Server Statistics
-------------------------------------------------------
URL Server Host Name 192.168.27.116
Server Address 192.168.27.116
Server Port 15868
Server Vendor Websense
Server Status Online
Requests Processed 4
Requests Allowed 1
Requests Denied 3
Server Timeouts 0
Server Retries 9
Responses Received 1
Late Server Responses 12
1 Minute Average Response Time 0
5 Minute Average Response Time 0
```

Related Commands

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

snmp-server enable traps firewall

To enable the router to send firewall Simple Network Management Protocol (SNMP) notifications, use the **snmp-server enable traps firewall** command in global configuration mode. To disable firewall SNMP notifications, use the **no** form of this command.

snmp-server enable traps firewall serverstatus

no snmp-server enable traps firewall serverstatus

Syntax Description

serverstatus	Displays the status of configured servers.

Command Default

SNMP notifications are disabled by default.

Command Modes

Global configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.

Usage Guidelines

SNMP notifications are sent as traps by the agent. Currently, only one URL filtering trap is generated.

For a complete description of the notification types and additional MIB functions, refer to the CISCO-UNIFIED-FIREWALL-MIB.my and CISCO-FIREWALL-TC.my files, available on Cisco.com through:

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

The **snmp-server enable traps firewall** command is used in conjunction with the **snmp-server host** command. Use the **snmp-server host** command to specify which host or hosts receive SNMP notifications. To send SNMP notifications, you must configure at least one **snmp-server host** command.

Examples

In the following example, the router is configured to send firewall MIB inform notifications to the host nms.cisco.com using the community string named “public”:

```
snmp-server enable traps firewall serverstatus
snmp-server host nms.cisco.com informs public firewall
```

Related Commands

Command	Description
snmp-server host	Specifies the recipient of an SNMP notification operation.

Feature Information for Cisco IOS Firewall MIB

Table 6 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. 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.

Note Table 6 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 6 Feature Information for Cisco IOS Firewall MIB

Feature Name	Releases	Feature Information
Cisco IOS Firewall MIB	12.4(6)T	Introduces support for the Cisco Unified Firewall MIB, which helps to manage and monitor firewall performance via SNMP. Statistics can be collected and monitored via standards-based SNMP techniques for firewall features such as stateful packet inspection and URL filtering.