



# Cisco IOS Firewall Performance Improvements

## Feature History

| Release  | Modification                 |
|----------|------------------------------|
| 12.2(8)T | This feature was introduced. |

This document describes the Cisco IOS Firewall Performance Improvements feature in Cisco IOS Release 12.2(8)T. It includes the following sections:

- [Feature Overview, page 1](#)
- [Supported Platforms, page 3](#)
- [Supported Standards, MIBs, and RFCs, page 4](#)
- [Configuration Tasks, page 5](#)
- [Configuration Examples, page 5](#)
- [Command Reference, page 6](#)

## Feature Overview

The Cisco IOS Firewall Performance Improvements feature introduces three performance metrics for Context-Based Access Control (CBAC)—[Throughput Improvement](#), [Connections Per Second Improvement](#), and [CPU Utilization Improvement](#).

CBAC is a context-based firewall that performs the following:

- Inspects traffic in one direction for network, transport, and application layer information
- Extracts relevant port information
- Dynamically creates access list entries for return traffic
- Closes ports at the end of a connection

CBAC also forces protocol conformance for some protocols, has a limited vulnerability signature detection mechanism, and extensive denial-of-service (DOS) prevention mechanisms. However, many of these features are CPU intensive, thereby, adversely affecting the performance of the router. The router is also affected during times of heavy traffic due to the processing of the base engine itself. With this feature, the performance of the router running CBAC is no longer subdued.

## Throughput Improvement

Throughput is a metric defined by the number of packets transferred from the input interface to the output interface within 1 second by a router running CBAC. When the CBAC base engine inspects packets that belong to an existing session, it must find out which session the packet belongs to; thus, the base engine implements a hash table to search for the session of the packet.

Collisions in a hash table result in poor hash function distribution because many entries are hashed into the same bucket for certain patterns of addresses. Even if a hash function distribution evenly dispenses the input across all of the buckets, a small hashtable size will not scale well if there are a large number of sessions. As the number of sessions increase, the collisions increase, which increases the length of the linked lists, thereby, deteriorating the throughput performance.

The Cisco IOS Firewall Performance Improvements feature allows users to dynamically change the size of the session hash table without reloading the router by using the **ip inspect hashtable** command. By increasing the size of the hash table, the number of sessions per hash bucket can be reduced, which improves the throughput performance of the base engine.

## Connections Per Second Improvement

Connections per second is a metric defined by the number of short-lived connections that can be created and deleted within 1 second by a router running CBAC. (These connections apply only to TCP connections because UDP is a connectionless protocol.)

Initially, CBAC had several restrictions that limited the connections per second metric. While a packet was being processed for connection setup and connection teardown of TCP connections, the base engine (which allocates and de-allocates memory while processing packets) would “bump up” several packets to the process switching path. Bumping up these packets drastically slowed down their processing. Also, the base engine had to process each packet again when it was bumped up into the process switching path, which also contributed to the degrading performance.

The Cisco IOS Firewall Performance Improvements feature prevents these restrictions by allowing only the first packet of any connection to be bumped up to the process switching path while the remaining packets are processed by the base engine in the fast path. Thus, the base engine is no longer slowed down by bumping up several packets or by processing packets twice.

**Note**

---

In this document, a connection is defined as creating a session, sending a data packet, and immediately deleting a session.

---

## CPU Utilization Improvement

The CPU utilization of the router running CBAC can be measured while a specific throughput or connections per second metric is maintained. This improvement is used in conjunction with the throughput and connections per second metrics.

## Benefits

### Layer 4 Processing Performance Improvement

This enhancement improves the connections per second metric and the CPU utilization. The code path for connection initiation and teardown was rewritten, thereby, enabling quicker creation of the connections per second metric, which reduces CPU utilization per connection.

### Hash Table Function Performance Improvement

With this enhancement, the hash function has been rewritten to ensure better distribution. This newly improved feature allows users to dynamically configure the size of the session hash table from 1K to 8K. When a packet belonging to an existing session comes into the router, a hash table is used to map the packet to an existing firewall session. As the number of sessions increases, the number of sessions hashing into the same bucket increases if the size of the hash table is fixed. By allowing the user to change the size of the hash table when the number of concurrent sessions increases or to reduce the search time for the session, the throughput performance of the base engine is greatly improved.

### Application Module Tuning Performance Improvement

This enhancement makes changes to application modules, ensuring that only the connection-initiating packet from all the packets belonging to the connection initiation and teardown is bumped up to the process switching path. Thus, the connections per second metric is significantly improved.

## Restrictions

To benefit from the performance enhancements, your router must be running CBAC.

## Related Documents

- “Traffic Filtering and Firewalls” part in the *Cisco IOS Security Configuration Guide*, Release 12.2
- “Traffic Filtering and Firewalls” part in the *Cisco IOS Security Command Reference*, Release 12.2

## Supported Platforms

- Cisco 800 series
- Cisco 805
- Cisco 820
- Cisco 827
- Cisco 1400 series
- Cisco 1600 series
- Cisco 1700 series
- Cisco 2500 series
- Cisco 2600 series
- Cisco 3620
- Cisco 3640

- Cisco 3660
- Cisco 7100 series
- Cisco 7200 series
- Cisco 7500 series
- Cisco Catalyst 6500 series MSFC software
- Cisco uBR7200 series
- Cisco uBR905
- Cisco uBR925

### Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to [cco-locksmith@cisco.com](mailto:cco-locksmith@cisco.com). An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at <http://www.cisco.com/register>.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

## Supported Standards, MIBs, and RFCs

### Standards

None

### MIBs

None

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

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

### RFCs

None

## Configuration Tasks

See the following sections for configuration tasks for the Cisco IOS Firewall Performance Improvements feature. Each task in the list is identified as either required or optional.

- [Changing the Size of the Hash Table](#) (required)
- [Verifying CBAC Configurations](#) (optional)

### Changing the Size of the Hash Table

You can increase the hash table to improve packet distribution. To change the size of the session hash table, use the following command in global configuration mode:

| Command  | Purpose   |
|--|---|
| Router# <code>ip inspect hashtable number</code> | <p>Changes the size of the hash table.</p> <p><i>number</i> specifies the size of the hash table in terms of buckets. Possible values for the hash table are 1024, 2048, 4096, and 8192; the default value is 1024.</p> <p><b>Note</b> You should increase the hash table size when the total number of sessions running through the CBAC router is approximately twice the current hash size; decrease the hash table size when the total number of sessions is reduced to approximately half the current hash size. Essentially, try to maintain a 1:1 ratio between the number of sessions and the size of the hash table.</p> |

### Verifying CBAC Configurations

To verify all CBAC configurations and all existing sessions that are currently being tracked and inspected by CBAC, use the `show ip inspect all` command in EXEC mode.

## Configuration Examples

This section provides the following configuration example:

- [Changing the Size of the Hash Table Example](#)

### Changing the Size of the Hash Table Example

The following example shows how to change the size of the hash table to 2048 buckets:

```
ip inspect hashtable 2048
```

# Command Reference

This section documents the new **ip inspect hashtable** command that configures the Cisco IOS Firewall Performance Improvements feature. All other commands used with this feature are documented in the Cisco IOS Release 12.2 command reference publications.

# ip inspect hashtable

To change the size of the session hash table, use the **ip inspect hashtable** command in global configuration mode. To restore the size of the session hash table to the default, use the **no** form of this command.

**ip inspect hashtable** *number*

**no ip inspect hashtable** *number*

|                           |               |   |
|---------------------------|---------------|---|
| <b>Syntax Description</b> | <i>number</i> | Size of the hash table in terms of buckets. Possible values for the hash table are 1024, 2048, 4096, and 8192; the default value is 1024. |
|---------------------------|---------------|---|

|                 |              |
|-----------------|--------------|
| <b>Defaults</b> | 1024 buckets |
|-----------------|--------------|

|                      |                      |
|----------------------|----------------------|
| <b>Command Modes</b> | Global configuration |
|----------------------|----------------------|

|                        |                |                              |
|------------------------|----------------|------------------------------|
| <b>Command History</b> | <b>Release</b> | <b>Modification</b>          |
|                        | 12.2(8)T       | This command was introduced. |

**Usage Guidelines** Use the **ip inspect hashtable** command to increase the size of the hash table when the number of concurrent sessions increases or to reduce the search time for the session. Collisions in a hash table result in poor hash function distribution because many entries are hashed into the same bucket for certain patterns of addresses. Even if a hash function distribution evenly dispenses the input across all of the buckets, a small hashtable size will not scale well if there are a large number of sessions. As the number of sessions increase, the collisions increase, which increases the length of the linked lists, thereby, deteriorating the throughput performance.



**Note**

You should increase the hash table size when the total number of sessions running through the CBAC router is approximately twice the current hash size; decrease the hash table size when the total number of sessions is reduced to approximately half the current hash size. Essentially, try to maintain a 1:1 ratio between the number of sessions and the size of the hash table.

**Examples** The following example shows how to change the size of the session hash table to 2048 buckets:

```
ip inspect hashtable 2048
```

