show c

show capture

To display the capture configuration when no options are specified, use the **show capture** command.

show capture [*capture_name*] [**access-list** *access_list_name*] [**count** *number*] [**decode**] [**detail**] [**dump**] [**packet-number** *number*] [**trace**]

Syntax Description

access-list *access_list_name*	(Optional) Displays information for packets that are based on IP or higher fields for the specific access list identification.
capture_name	(Optional) Specifies the name of the packet capture.
count *number*	(Optional) Displays the number of packets specified data. Valid values are from 0- 4294967295.
decode	This option is useful when a capture of type isakmp is applied to an interface. All ISAKMP data flowing through that interface will be captured after decryption and shown with more information after decoding the fields.
detail	(Optional) Displays additional protocol information for each packet.
dump	(Optional) Displays a hexadecimal dump of the packets that are transported over the data link.
packet-number *number*	(Optional) Starts the display at the specified packet number. Valid values are from 0- 4294967295.
trace	(Optional) Displays extended trace information for each packet - used if capture is set using the trace keyword as mentioned above, this will show the output of packet tracer for each packet in the inbound direction.

Command History

Release	Modification
6.1	This command was introduced.
7.2.6 7.4.1	The output of show capture detail for the physical port displays the drop configuration (disable or mac-filter).

Usage Guidelines

If you specify the capture name, then the capture buffer contents for that capture are displayed.

The **dump** keyword does not display MAC information in the hexadecimal dump.

The decoded output of the packets depend on the protocol of the packet. In the following table, the bracketed output is displayed when you specify the **detail** keyword.

Table 1: Packet Capture Output Formats

Packet Type	Capture Output Format
802.1Q	*HH:MM:SS.ms* [ether-hdr] *VLAN-info encap-ether-packet*

Packet Type	Capture Output Format
ARP	*HH:MM:SS.ms* [ether-hdr] *arp-type arp-info*
IP/ICMP	*HH:MM:SS.ms* [ether-hdr] *ip-source* > *ip-destination:* icmp: *icmp-type icmp-code* [checksum-failure]
IP/UDP	*HH:MM:SS.ms* [ether-hdr] *src-addr.src-port dest-addr.dst-port:* [checksum-info] udp *payload-len*
IP/TCP	*HH:MM:SS.ms* [ether-hdr] *src-addr.src-port dest-addr.dst-port: tcp-flags* [header-check] [checksum-info] *sequence-number ack-number tcp-window urgent-info tcp-options*
IP/Other	*HH:MM:SS.ms* [ether-hdr] *src-addr dest-addr: ip-protocol ip-length*
Other	*HH:MM:SS.ms ether-hdr: hex-dump*

If the threat defense device receives packets with an incorrectly formatted TCP header and drops them because of the ASP drop reason invalid-tcp-hdr-length, the **show capture** command output on the interface where those packets are received does not show those packets.

Note When the file size option is used:

- The **show capture** [*capture_name*] command shows the number of packets captured and skipped.
- The **show capture** command shows the captured data in KB and MB.

Examples

This example shows how to display the capture configuration:

```
> show capture
capture arp ethernet-type arp interface outside
capture http access-list http packet-length 74 interface inside
```

This example shows how to display the packets that are captured by an ARP capture:

```
> show capture arp
2 packets captured
19:12:23.478429 arp who-has 171.69.38.89 tell 171.69.38.10
19:12:26.784294 arp who-has 171.69.38.89 tell 171.69.38.10
2 packets shown
```

The following example shows how to display the packets that are captured on a single unit in a clustering environment:

```
> show capture
capture 1 cluster type raw-data interface primary interface cluster [Buffer Full - 524187
bytes]
```

```
capture 2 type raw-data interface cluster [Capturing - 232354 bytes]
```

The following example shows how to display the packets that are captured on all units in a clustering environment:

```
> cluster exec show capture
mycapture (LOCAL):--------------------------------------------------------
capture 1 type raw-data interface primary [Buffer Full - 524187 bytes]
capture 2 type raw-data interface cluster [Capturing - 232354 bytes]

yourcapture:--------------------------------------------------------------
capture 1 type raw-data interface primary [Capturing - 191484 bytes]
capture 2 type raw-data interface cluster [Capturing - 532354 bytes]
```

The following example shows the packets that are captured when SGT plus Ethernet tagging has been enabled on an interface:

```
> show capture my-inside-capture
1: 11:34:42.931012 INLINE-TAG 36 10.0.101.22 > 11.0.101.100: icmp: echo request
2: 11:34:42.931470 INLINE-TAG 48 11.0.101.100 > 10.0.101.22: icmp: echo reply
3: 11:34:43.932553 INLINE-TAG 36 10.0.101.22 > 11.0.101.100: icmp: echo request
4: 11.34.43.933164 INLINE-TAG 48 11.0.101.100 > 10.0.101.22: icmp: echo reply
```

When SGT plus Ethernet tagging has been enabled on an interface, the interface can still receive tagged or untagged packets. The example shown is for tagged packets, which have INLINE-TAG 36 in the output. When the same interface receives untagged packets, the output remains unchanged (that is, no “INLINE-TAG 36” entry is included in the output).

The following example shows the hardware log with mac-filter drop enabled packet capture of a Secure Firewall 3100 device:

```
firepower-3110(local-mgmt)# show portmanagerswitch pktcap-rules hardware
Hardware DB rule:1
Hw_index= 6150
Rule_id= 6144
CounterIndex= 0
Packet_count= 1448
Slot= 1
Interface= 1
Protocol= 0
Ethertype= 0x0000V
lan= 3178
SrcPort= 0
DstPort= 0
SrcIp= 0.0.0.0
DstIp= 0.0.0.0
SrcIpv6= ::
DestIpv6= ::
SrcMacAddr= 00:00:00:00:00:00
DestMacAddr= 00:00:00:00:00:00
```

Here, the hardware counter index 0 is assigned for mac-filter drop hardware entry. In addtion the mac-filter dropped packets are included in the packet count.

The following example shows the software log with mac-filter drop enabled packet capture of a Secure Firewall 3100 device:

```
firepower-3110(local-mgmt)# show portmanagerswitch pktcap-rules software
```

```
Software DB rule:1
Slot= 1
Interface= 1
Breakout-port= 0
Protocol= 0
Ethertype= 0x0000
Filter_key= 0x00000200
Session= 4
Vlan= 3178
SrcPort= 0
DstPort= 0
SrcIp= 0.0.0.0
DstIp= 0.0.0.0
SrcIpv6= ::
DestIpv6= ::
SrcMacAddr= 00:00:00:00:00:00
DestMacAddr= 00:00:00:00:00:00
DropFilterEnabled= 1
```

Related Commands

Command	Description
capture	Enables packet capture capabilities for packet sniffing and network fault isolation.
clear capture	Clears the capture buffer.
copy capture	Copies a capture file to a server.

show cert-update

To display the status of automatic updation of CA certificates on the threat defense device, use the **show cert-update** command.

show cert-update

Command History

Release	Modification
7.0.5	This command was introduced.

Examples

The following is sample output from the **show cert-update** command:

```
> show cert-update
Autoupdate is enabled and set for every day at 09:34 UTC
CA bundle was last modified 'Thu Sep 15 16:12:35 2022'
```

Related Commands

Command	Description
configure cert-update auto-update	Enables or disables automatic update of CA certificates every day.
configure cert-update run-now	Instantly attempt to update CA certifications.
configure cert-update test	Performs connection checks using the latest CA certificates from the Cisco server.

show checkheaps

To show the checkheaps statistics, use the **show checkheaps** command. Checkheaps is a periodic process that verifies the sanity of the heap memory buffers (dynamic memory is allocated from the system heap memory region) and the integrity of the code region.

show checkheaps

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show checkheaps** command:

```
> show checkheaps
Checkheaps stats from buffer validation runs
--------------------------------------------
Time elapsed since last run     : 42 secs
Duration of last run            : 0 millisecs
Number of buffers created       : 8082
Number of buffers allocated     : 7808
Number of buffers free          : 274
Total memory in use             : 43570344 bytes
Total memory in free buffers    : 87000 bytes
Total number of runs            : 310
```

show checksum

To display the configuration checksum, use the **show checksum** command.

show checksum

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **show checksum** command allows you to display four groups of hexadecimal numbers that act as a digital summary of the configuration contents. This checksum is calculated only when you store the configuration in flash memory.

If a dot (".") appears before the checksum in the **show running-config** or **show checksum** command output, the output indicates a normal configuration load or write mode indicator (when loading from or writing to the threat defense flash partition). The "." shows that the threat defense device is preoccupied with the operation but is not "hung up." This message is similar to a "system processing, please wait" message.

Examples

This example shows how to display the configuration or the checksum:

```
> show checksum
Cryptochecksum: 1a2833c0 129ac70b 1a88df85 650dbb81
```

show chunkstat

To display the chunk statistics, use the **show chunkstat** command.

show chunkstat

Command History

Release	Modification
6.1	This command was introduced.

Examples

This example shows how to display the chunk statistics:

```
> show chunkstat
Global chunk statistics: created 181, destroyed 34, siblings created 94, siblings destroyed
 34

Per-chunk statistics: siblings created 0, siblings trimmed 0
Dump of chunk at 01edb4cc, name "Managed Chunk Queue Elements", data start @ 01edbd24, end
 @ 01eddc54
next: 01eddc8c, next_sibling: 00000000, prev_sibling: 00000000
flags 00000001
maximum chunk elt's: 499, elt size: 16, index first free 498
# chunks in use: 1, HWM of total used: 1, alignment: 0
Per-chunk statistics: siblings created 0, siblings trimmed 0
Dump of chunk at 01eddc8c, name "Registry Function List", data start @ 01eddea4, end @
01ede348
next: 01ede37c, next_sibling: 00000000, prev_sibling: 00000000
flags 00000001
maximum chunk elt's: 99, elt size: 12, index first free 42
# chunks in use: 57, HWM of total used: 57, alignment: 0
```

Related Commands

Command	Description
show counters	Displays the protocol stack counters.
show cpu	Displays the CPU utilization information.

show clns

To show Connectionless-mode Network Service (CLNS) information for IS-IS, use the **show clns** command.

show clns {**filter-set** [*name*] | **interface** [*interface_name*] | **is-neighbors** [*interface_name* [**detail**] | **neighbors** [**areas**] [*interface_name*] [**detail**] | **protocol** [*domain*] | **traffic**}

Syntax Description

filter-set [*name*]	Shows CLNS filter sets. You can optionally specify the name of a filter set.
interface [*interface_name*]	Shows CLNS interface status and configuration. You can optionally specify the name of an interface to focus the output.
is-neighbors [*interface_name*] [**detail**]	Shows IS neighbor adjacencies. Neighbor entries are sorted according to the area in which they are located. You can optionally specify the name of an interface to focus the output. Specify **detail** to include the areas associated with the intermediate systems. Otherwise, a summary display is provided.
neighbors [**areas**] [*interface_name*] [**detail**]	Displays end system (ES), intermediate system (IS), and multitopology Integrated Intermediate System-to-Intermediate System (M-ISIS) neighbors. You can optionally specify the name of an interface to focus the output. Include the **areas** keyword to show CLNS multiarea adjacencies. Specify **detail** to include the areas associated with the intermediate systems. Otherwise, a summary display is provided.
protocol [*domain*]	Shows CLNS routing protocol process information. There will always be at least two routing processes, a Level 1 and a Level 2, and there can be more. You can optionally specify the name of a CLNS domain to focus the output.
traffic	Lists the CLNS packets that this router has seen.

Command History

Release	Modification
6.3	This command was introduced.

Examples

The following example shows the CLNS filter sets defined in the running configuration, and displays them using the **show clns filter-set** command.

```
> show running-config clns
clns filter-set US-OR-NORDUNET permit 47.0005...
clns filter-set US-OR-NORDUNET permit 47.0023...
clns filter-set LOCAL permit 49.0003
> show clns filter-set

CLNS filter set US-OR-NORDUNET
      permit 47.0005...
      permit 47.0023...
```

```
CLNS filter set LOCAL
      permit 49.0003...
```

The following is sample output from the **show clns interface** command. The information under "Routing Protocol: IS-IS" displays information pertaining to Intermediate System-to-Intermediate System (IS-IS), including the Level 1 and Level 2 metrics, priorities, circuit IDs, and number of active Level 1 and Level 2 adjacencies.

```
> show clns interface
GigabitEthernet0/1 is up, line protocol is up
  Checksums enabled, MTU 1500
  ERPDUs enabled, min. interval 10 msec.
  DEC compatibility mode OFF for this interface
  Next ESH/ISH in 0 seconds
  Routing Protocol: IS-IS
    Circuit Type: level-1-2
    Interface number 0x0, local circuit ID 0x1
    Level-1 Metric: 10, Priority: 64, Circuit ID: c2.01
    DR ID: c2.01
    Level-1 IPv6 Metric: 10
    Number of active level-1 adjacencies: 3
    Level-2 Metric: 10, Priority: 64, Circuit ID: c2.01
    DR ID: c2.01
    Level-2 IPv6 Metric: 10
    Number of active level-2 adjacencies: 3
    Next IS-IS LAN Level-1 Hello in 1 seconds
    Next IS-IS LAN Level-2 Hello in 1 seconds
```

The following is sample output from the **show clns neighbors** command.

```
> show clns neighbors

System Id      Interface   SNPA            State  Holdtime  Type Protocol
CSR7001        inside      000c.2921.ff44  Up          29   L1L2
CSR7002        inside      000c.2906.491c  Up          27   L1L2
```

The following table explains the fields in the neighbors output.

Table 2: Fields in the Neighbors Output

Field	Description
System Id	The six-byte value that identifies a system in an area.
Interface	The name of the interface from which the system was learned.
SNPA	The Subnetwork Point of Attachment. This is the data-link address.
State	The state of the ES, IS, or M-ISIS. • Init—The system is an IS and it is waiting for an IS-IS hello message. IS-IS regards the neighbor as not adjacent. • Up—The system believes the ES or IS is reachable.
Holdtime	The number of seconds before this adjacency entry times out.

Field	Description
Type	The adjacency type. • ES—An end-system adjacency either discovered via the ES-IS protocol or statically configured. • IS—A router adjacency either discovered via the ES-IS protocol or statically configured. • M-ISIS—A router adjacency discovered via the multitopology IS-IS protocol. • L1—A router adjacency for Level 1 routing only. • L1L2—A router adjacency for Level 1 and Level 2 routing. • L2—A router adjacency for Level 2 only.
Protocol	Protocol through which the adjacency was learned. Valid protocol sources are ES-IS, IS-IS, ISO IGRP, Static, DECnet, and M-ISIS.

The following is sample output from the **show clns neighbors detail** command.

```
> show clns neighbors detail

System Id      Interface   SNPA            State  Holdtime  Type Protocol
CSR7001        inside      000c.2921.ff44      Up          26   L1L2
  Area Address(es): 49.0001
  IP Address(es):  1.3.3.3*
  Uptime: 01:16:33
  NSF capable
  Interface name: inside
CSR7002        inside      000c.2906.491c      Up          27   L1L2
  Area Address(es): 49.0001
  IP Address(es):  20.3.3.3*
  Uptime: 01:16:33
  NSF capable
  Interface name: inside
```

The following is sample output from the **show clns is-neighbors** command.

```
> show clns is-neighbors

System Id      Interface   State  Type Priority  Circuit Id         Format
CSR7001        inside      Up     L1L2 64/64    ciscoasa.01        Phase V
CSR7002        inside      Up     L1L2 64/64    ciscoasa.01        Phase V
```

The following table explains the columns in the is-neighbors output.

Table 3: Fields in the IS Neighbors Output

Field	Description
System Id	The identification value of the system.
Interface	The interface on which the router was discovered.

Field	Description
State	The adjacency state. Up and Init are the states. For details, see the **show clns neighbors** description.
Type	The adjacency type: L1, L2, or L1L2. For details, see the **show clns neighbors** description.
Priority	The IS-IS priority that the respective neighbor is advertising. The highest priority neighbor is elected the designated IS-IS router for the interface.
Circuit Id	The neighbor's idea of what the designated IS-IS router is for the interface.
Format	The format, which indicates if the neighbor is either a Phase V (OSI) adjacency or Phase IV (DECnet) adjacency.

The following is sample output from the **show clns is-neighbors detail** command.

```
> show clns is-neighbors detail

System Id      Interface   State  Type Priority  Circuit Id         Format
CSR7001        inside      Up     L1L2 64/64    ciscoasa.01        Phase V
  Area Address(es): 49.0001
  IP Address(es):  1.3.3.3*
  Uptime: 00:12:49
  NSF capable
  Interface name: inside
CSR7002        inside      Up     L1L2 64/64    ciscoasa.01        Phase V
  Area Address(es): 49.0001
  IP Address(es):  20.3.3.3*
  Uptime: 00:12:50
  NSF capable
  Interface name: inside
```

The following is sample output from the **show clns protocol** command.

```
> show clns protocol
IS-IS Router
  System Id: 0050.0500.5008.00  IS-Type: level-1-2
  Manual area address(es):
        49.0001
  Routing for area address(es):
        49.0001
  Interfaces supported by IS-IS:
        outside - IP
  Redistribute:
    static (on by default)
  Distance for L2 CLNS routes: 110
  RRR level: none
  Generate narrow metrics: level-1-2
  Accept narrow metrics:   level-1-2
  Generate wide metrics:   none
  Accept wide metrics:     none
```

The following is sample output from the **show clns traffic** command.

```
> show clns traffic

CLNS:  Time since last clear: never
```

```
CLNS & ESIS Output: 0, Input: 8829
CLNS Local: 0, Forward: 0
CLNS Discards:
  Hdr Syntax: 0, Checksum: 0, Lifetime: 0, Output cngstn: 0
  No Route: 0, Discard Route: 0, Dst Unreachable 0, Encaps. Failed: 0
  NLP Unknown: 0, Not an IS: 0
CLNS Options: Packets 0, total 0 , bad 0, GQOS 0, cngstn exprncd 0
CLNS Segments:  Segmented: 0, Failed: 0
CLNS Broadcasts: sent: 0, rcvd: 0
Echos: Rcvd 0 requests, 0 replies
      Sent 0 requests, 0 replies
ESIS(sent/rcvd): ESHs: 0/0, ISHs: 0/0, RDs: 0/0, QCF: 0/0
Tunneling (sent/rcvd): IP: 0/0, IPv6: 0/0
Tunneling dropped (rcvd) IP/IPV6:  0
ISO-IGRP: Querys (sent/rcvd): 0/0 Updates (sent/rcvd): 0/0
ISO-IGRP: Router Hellos: (sent/rcvd): 0/0
ISO-IGRP Syntax Errors: 0

IS-IS: Time since last clear: never
IS-IS: Level-1 Hellos (sent/rcvd): 1928/1287
IS-IS: Level-2 Hellos (sent/rcvd): 1918/1283
IS-IS: PTP Hellos     (sent/rcvd): 0/0
IS-IS: Level-1 LSPs sourced (new/refresh): 7/13
IS-IS: Level-2 LSPs sourced (new/refresh): 7/14
IS-IS: Level-1 LSPs flooded (sent/rcvd): 97/2675
IS-IS: Level-2 LSPs flooded (sent/rcvd): 73/2628
IS-IS: LSP Retransmissions: 0
IS-IS: Level-1 CSNPs (sent/rcvd): 642/0
IS-IS: Level-2 CSNPs (sent/rcvd): 639/0
IS-IS: Level-1 PSNPs (sent/rcvd): 0/554
IS-IS: Level-2 PSNPs (sent/rcvd): 0/390
IS-IS: Level-1 DR Elections: 1
IS-IS: Level-2 DR Elections: 1
IS-IS: Level-1 SPF Calculations: 9
IS-IS: Level-2 SPF Calculations: 8
IS-IS: Level-1 Partial Route Calculations: 0
IS-IS: Level-2 Partial Route Calculations: 0
IS-IS: LSP checksum errors received: 0
IS-IS: Update process queue depth: 0/200
IS-IS: Update process packets dropped: 0
```

The following table explains the fields in the traffic output.

Table 4: Fields in the Traffic Output

Fields	Description
CLNS & ESIS Output	The total number of packets that this router has sent.
Input	The total number of packets that this router has received.
CLNS Local	The number of packets that were generated by this router.
Forward	The number of packets that this router has forwarded.
CLNS Discards	The number of packets that CLNS has discarded, classified by the reason for the discard.
CLNS Options	The options seen in CLNS packets.

Fields	Description
CLNS Segments	The number of packets segmented and the number of failures that occurred because a packet could not be segmented.
CLNS Broadcasts	The number of CLNS broadcasts sent and received.
Echos	The number of echo request packets and echo reply packets received. The line following this field lists the number of echo request packets and echo reply packets sent.
ESIS (sent/rcvd)	The number of End System Hello (ESH), Intermediate System Hello (ISH), and redirects sent and received.
ISO IGRP	The number of ISO Interior Gateway Routing Protocol (IGRP) queries and updates sent and received.
Router Hellos	The number of ISO IGRP router hello packets sent and received.
IS-IS: Level-1 hellos (sent/rcvd)	The number of Level 1 IS-IS hello packets sent and received.
IS-IS: Level-2 hellos (sent/rcvd)	The number of Level 2 IS-IS hello packets sent and received.
IS-IS: PTP hellos (sent/rcvd)	The number of point-to-point IS-IS hello packets sent and received over serial links.
IS-IS: Level-1 LSPs (sent/rcvd)	The number of Level 1 link-state Protocol Data Unit (PDUs) sent and received.
IS-IS: Level-2 LSPs (sent/rcvd)	The number of Level 2 link-state PDUs sent and received.
IS-IS: Level-1 CSNPs (sent/rcvd)	The number of Level 1 Complete Sequence Number Packets (CSNP) sent and received.
IS-IS: Level-2 CSNPs (sent/rcvd)	The number of Level 2 CSNPs sent and received.
IS-IS: Level-1 PSNPs (sent/rcvd)	The number of Level 1 Partial Sequence Number Packets (PSNP) sent and received.
IS-IS: Level-2 PSNPs (sent/rcvd)	The number of Level 2 PSNPs sent and received.
IS-IS: Level-1 DR Elections	The number of times Level 1 designated router election occurred.
IS-IS: Level-2 DR Elections	The number of times Level 2 designated router election occurred.
IS-IS: Level-1 SPF Calculations	The number of times the Level 1 shortest-path-first (SPF) tree was computed.

Fields	Description
IS-IS: Level-2 SPF Calculations	The number of times the Level 2 SPF tree was computed.

Related Commands

Command	Description
clear clns	Clears CLNS-specific information.

show cluster

To view aggregated data for the entire cluster or other information, use the **show cluster** command.

show cluster { access-list [*acl_name*] | **conn** [**count**] | **cpu** [**usage**] | **interface-mode** | **memory** | **resource usage** | **service-policy** | **traffic** | **xlate count** | **zero-trust statistics** }

Syntax Description

access-list [*acl_name*]	Shows hit counters for access policies. To see the counters for a specific ACL, enter the acl_name.
conn [**count**]	Shows the aggregated count of in-use connections for all units. If you enter the **count** keyword, only the connection count is shown.
cpu [**usage**]	Shows CPU usage information.
interface-mode	Shows the cluster interface mode, either spanned or individual.
memory	Shows system memory utilization and other information.
resource usage	Shows system resources and usage.
service-policy	Shows the MPF service policy statistics.
traffic	Shows traffic statistics.
xlate count	Shows current translation information.
zero-trust statistics	Shows the summary of zero trust statistics across nodes in a cluster

Command History

Release	Modification
7.4	Added the **zero-trust statistics** keyword.
6.1	This command was introduced.

Examples

The following is sample output from the **show cluster access-list** command:

```
> show cluster access-list
hitcnt display order: cluster-wide aggregated result, unit-A, unit-B,  unit-C, unit-D
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval 300
access-list 101; 122 elements; name hash: 0xe7d586b5
access-list 101 line 1 extended permit tcp 192.168.143.0 255.255.255.0 any eq www
  (hitcnt=0, 0, 0, 0, 0) 0x207a2b7d
access-list 101 line 2 extended permit tcp any 192.168.143.0 255.255.255.0
  (hitcnt=0, 0, 0, 0, 0) 0xfe4f4947
access-list  101 line 3 extended permit tcp host 192.168.1.183 host 192.168.43.238
  (hitcnt=1, 0, 0, 0, 1) 0x7b521307
access-list 101 line 4 extended permit tcp host 192.168.1.116 host 192.168.43.238
  (hitcnt=0, 0, 0, 0, 0) 0x5795c069
```

```
access-list 101 line 5 extended permit tcp host 192.168.1.177 host 192.168.43.238
  (hitcnt=1, 0, 0, 1, 0) 0x51bde7ee
access list 101 line 6 extended permit tcp host 192.168.1.177 host 192.168.43.13
  (hitcnt=0, 0, 0, 0, 0) 0x1e68697c
access-list 101 line 7 extended permit tcp host 192.168.1.177 host 192.168.43.132
  (hitcnt=2, 0, 0, 1, 1) 0xc1ce5c49
access-list 101 line 8 extended permit tcp host 192.168.1.177 host 192.168.43.192
  (hitcnt=3, 0, 1, 1, 1) 0xb6f59512
access-list 101 line 9 extended permit tcp host 192.168.1.177 host 192.168.43.44
  (hitcnt=0, 0, 0, 0, 0) 0xdc104200
access-list 101 line 10 extended permit tcp host 192.168.1.112 host 192.168.43.44
  (hitcnt=429, 109, 107, 109, 104)0xce4f281d
access-list 101 line 11 extended permit tcp host 192.168.1.170 host 192.168.43.238
  (hitcnt=3, 1, 0, 0, 2) 0x4143a818
access-list 101 line 12 extended permit tcp host 192.168.1.170 host 192.168.43.169
  (hitcnt=2, 0, 1, 0, 1) 0xb18dfea4
access-list 101 line 13 extended permit tcp host 192.168.1.170 host 192.168.43.229
  (hitcnt=1, 1, 0, 0, 0) 0x21557d71
access-list 101 line 14 extended permit tcp host 192.168.1.170 host 192.168.43.106
  (hitcnt=0, 0, 0, 0, 0) 0x7316e016
access-list 101 line 15 extended permit tcp host 192.168.1.170 host 192.168.43.196
  (hitcnt=0, 0, 0, 0, 0) 0x013fd5b8
access-list 101 line 16 extended permit tcp host 192.168.1.170 host 192.168.43.75
  (hitcnt=0, 0, 0, 0, 0) 0x2c7dba0d
```

To display the aggregated count of in-use connections for all units, enter:

```
> show cluster conn count
Usage Summary In Cluster:*********************************************
  200 in use (cluster-wide aggregated)
     cl2(LOCAL):************************************************************
  100 in use, 100 most used
  cl1:************************************************************************
  100 in use, 100 most used
```

The following is sample output for the zero trust statistics across nodes in a cluster. The summary section shows a cumulative sum of statistics across nodes in the cluster. The subsequent sections display the statistics in the respective nodes.

```
> show cluster zero-trust statistics
Usage Summary In Cluster:*************************************************
Active zero-trust sessions                    5
Active users                                  0*
Total zero-trust sessions                     5
Total users authorised                        0*
Total zero-trust sessions failed              0*
Total active applications                     2
Total SAML AuthN Requests                     5
Total SAML AuthN Responses                    5
Total SAML Auth Failures                      0*
SAML Assertions Passed                        5
SAML Assertions Failed                        0*
Total bytes in                                1000 Bytes
Total bytes out                               27570 Bytes
Pre-auth latency in millisec (min/max/avg)    7/11/9
Post-auth latency in millisec (min/max/avg)   6/9/7

unit-1-1(LOCAL):*****************************************************
Active zero-trust sessions                    5
Active users                                  0*
Total zero-trust sessions                     5
```

```
Total users authorised                          0*
Total zero-trust sessions failed                0*
Total active applications                       2
Total SAML AuthN Requests                       5
Total SAML AuthN Responses                      5
Total SAML Auth Failures                        0*
SAML Assertions Passed                           5
SAML Assertions Failed                           0*
Total bytes in                                  1000 Bytes
Total bytes out                                 27570 Bytes
Pre-auth latency in millisec (min/max/avg)       7/11/9
Post-auth latency in millisec (min/max/avg)      6/9/7
```

Related Commands

Command	Description
clear zero-trust	Clears zero trust sessions and statistics
show cluster info	Shows cluster information.
show counters protocol zero_trust	Displays the counters that are hit for zero trust flow
show zero-trust	Displays the run-time zero trust statistics and session information

show cluster history

To view event history for the cluster, use the **show cluster history** command in privileged EXEC mode.

show cluster history [**brief**] [**latest** [*number*]] [**reverse**] [**time** [*year month day*] *hh* **:** *mm* **:** *ss*]

Syntax Description

brief	Shows cluster history without generic events.
latest [*number*]	Displays the latest events. By default, the device shows the last 512 events. You can limit the *number* of events, between 1 and 512.
reverse	Shows events in reverse order.
time [*year month day*] *hh***:***mm***:***ss*	Shows events before a specified date and time.

Command Default

No default behavior or values.

Command History

Release	Modification
7.0	We added the **brief**, **latest**, **reverse**, **time** keywords.
6.6	The **show cluster history** command was enhanced with messages about why a cluster unit failed to join or left the cluster.
6.1	This command was added.

Usage Guidelines

The following is sample output from the **show cluster history time** command:

```
> show cluster history time august 26 10:10:05
==================================================================================
From State            To State              Reason
==================================================================================

10:08:49 UTC Aug 26 2020
DISABLED              DISABLED              Disabled at startup


10:09:43 UTC Aug 26 2020
DISABLED              ELECTION              Enabled from CLI


10:10:01 UTC Aug 26 2020
ELECTION              ONCALL                Event: Cluster unit A state is MASTER


10:10:02 UTC Aug 26 2020
ONCALL                SLAVE_COLD            Slave proceeds with configuration sync


10:10:02 UTC Aug 26 2020
SLAVE_COLD            SLAVE_CONFIG          Client progression done
```

```
10:10:04 UTC Aug 26 2020
SLAVE_CONFIG          SLAVE_FILESYS         Configuration replication finished

10:10:05 UTC Aug 26 2020
SLAVE_FILESYS         SLAVE_BULK_SYNC       Client progression done
```

The following is sample output from the **show cluster history brief** command:

```
> show cluster history brief
================================================================================
From State            To State              Reason
================================================================================

10:08:49 UTC Aug 26 2020
DISABLED              DISABLED              Disabled at startup

10:09:43 UTC Aug 26 2020
DISABLED              ELECTION              Enabled from CLI



10:10:02 UTC Aug 26 2020
ONCALL                SLAVE_COLD            Slave proceeds with configuration sync

10:10:02 UTC Aug 26 2020
SLAVE_COLD            SLAVE_CONFIG          Client progression done

10:10:04 UTC Aug 26 2020
SLAVE_CONFIG          SLAVE_FILESYS         Configuration replication finished

10:10:05 UTC Aug 26 2020
SLAVE_FILESYS         SLAVE_BULK_SYNC       Client progression done
```

The following is sample output from the **show cluster history latest** command:

```
> show cluster history latest 3
================================================================================
From State            To State              Reason
================================================================================

10:10:05 UTC Aug 26 2020
SLAVE_FILESYS         SLAVE_BULK_SYNC       Client progression done

10:10:04 UTC Aug 26 2020
SLAVE_CONFIG          SLAVE_FILESYS         Configuration replication finished

10:10:02 UTC Aug 26 2020
SLAVE_COLD            SLAVE_CONFIG          Client progression done
```

Related Commands

Command	Description
show cluster	Shows aggregated data for the entire cluster and other information.
show cluster info	Shows cluster information.

show cluster info

To view cluster information, use the **show cluster info** command.

show cluster info [**auto-join** | **clients** | **conn-distribution** | **flow-mobility counters** | **goid** [*options*] | **health** | **incompatible-config** | **instance-type** | **loadbalance** | **old-members** | **packet-distribution** | **trace** [*options*] | **transport** { **asp** | **cp** }]

Syntax Description

auto-join	Shows whether the cluster unit will automatically rejoin the cluster after a time delay and if the failure conditions (such as waiting for the license, chassis health check failure, and so on) are cleared. If the unit is permanently disabled, or if the unit is already in the cluster, then this command will not show any output.
clients	(Optional) Shows the version of register clients.
conn-distribution	(Optional) Shows the connection distribution in the cluster.
flow-mobility counters	(Optional) Shows EID movement and flow owner movement information.
goid [*options*]	(Optional) Shows the global object ID database. Options include: classmap conn-set hwidb idfw-domain idfw-group interface policymap virtual-context
health	(Optional) Shows health monitoring information.
incompatible-config	(Optional) Shows commands that are incompatible with clustering in the current running configuration. This command is useful before you enable clustering.
instance-type	(Optional) Shows the module type and resource size per cluster member when using multi-instance clustering.
loadbalance	(Optional) Shows load balancing information.
old-members	(Optional) Shows former members of the cluster.
packet-distribution	(Optional) Shows packet distribution in the cluster.

trace [*options*]	(Optional) Shows the clustering control module event trace. Options include: • **latest** [*number*]—Displays the latest number events, where the number is from 1 to 2147483647. The default is to show all. • **level** *level*—Filters events by level where the level is one of the following: **all**, **critical**, **debug**, **informational**, or **warning**. • **module** *module*—Filters events by module where the module is one of the following: **ccp**, **datapath**, **fsm**, **general**, **hc**, **license**, **rpc**, or **transport**. • **time** {[*month day*] [*hh:mm:ss*]}—Shows events before the specified time or date.
transport {**asp** | **cp**}	(Optional) Show transport related statistics for the following: • **asp**—Data plane transport statistics. • **cp**—Control plane transport statistics.

Command History

Release	Modification
6.1	This command was introduced.
6.2.3	Added the **auto-join** keyword.
6.6	The output was enhanced to show multi-instance clustering characteristics. The **instance-type** keyword was also added to show the module type and resource size per cluster member.

Usage Guidelines

If you do not specify any options, the **show cluster info** command shows general cluster information including the cluster name and status, the cluster members, the member states, and so on.

Clear statistics using the **clear cluster info** command.

Examples

The following is sample output from the **show cluster info** command:

```
> show cluster info
Cluster stbu: On
  This is "C" in state SLAVE
      ID        : 0
      Site ID : 1
      Version   : 6.2
      Serial No.: P3000000025
      CCL IP    : 10.0.0.3
      CCL MAC   : 000b.fcf8.c192
      Last join : 17:08:59 UTC Sep 26 2011
      Last leave: N/A
Other members in the cluster:
  Unit "D" in state SLAVE
      ID        : 1
      Site ID : 1
      Version   : 6.2
```

```
      Serial No.: P3000000001
      CCL IP    : 10.0.0.4
      CCL MAC   : 000b.fcf8.c162
      Last join : 19:13:11 UTC Sep 23 2011
      Last leave: N/A
  Unit "A" in state MASTER
      ID        : 2
      Site ID : 2
      Version   : 6.2
      Serial No.: JAB0815R0JY
      CCL IP    : 10.0.0.1
      CCL MAC   : 000f.f775.541e
      Last join : 19:13:20 UTC Sep 23 2011
      Last leave: N/A
  Unit "B" in state SLAVE
      ID        : 3
      Site ID : 2
      Version   : 6.2
      Serial No.: P3000000191
      CCL IP    : 10.0.0.2
      CCL MAC   : 000b.fcf8.c61e
      Last join : 19:13:50 UTC Sep 23 2011
      Last leave: 19:13:36 UTC Sep 23 2011
```

The following is sample output from the **show cluster info** command when using multi-instance clustering:

```
> show cluster info
Cluster MI: On
    Interface mode: spanned
    This is "unit-3-1" in state MASTER
        ID             : 0
        Site ID         : 1
        Version           : 6.6
        Serial No.        : FLM2123050F12T
        CCL IP             : 127.2.3.1
        CCL MAC   : a28e.6000.0012
        Module.
: FPR4K-SM-12
        Resource.
: 10 cores / 23876 MB RAM
        Last join          : 19:48:33 UTC Nov 13 2018
        Last leave: N/A
Other members in the cluster:
    Unit "unit-4-1" in state SLAVE
        ID               : 1
        Site ID           : 1
        Version           : 6.6
        Serial No.        : FLM212305ELPXW
        CCL IP            : 127.2.4.1
        CCL MAC     : a2f7.2000.0009
        Module
: FPR4K-SM-12
        Resource
: 6 cores / 14426 MB RAM
        Last join           : 20:29:55 UTC Nov 14 2018
        Last leave        : 19:07:53 UTC Nov 14 2018


Warning: Mixed module and / or mismatched resource profile size in cluster. System may not
 run in an optimized state.
```

The following is sample output from the **show cluster info instance-type** command when using multi-instance clustering:

```
> show cluster info instance-type

     Cluster Member     Module Type                CPU Cores        RAM (MB)
         unit-3-1       FPR4K-SM-12                  10              23876
         unit-4-1       FPR4K-SM-12                  6               14446

Warning: Mixed module type and / or mismatched resource profile in cluster. System may not
 run in an optimized state.
```

The following is sample output from the **show cluster info incompatible-config** command:

```
> show cluster info incompatible-config
INFO: Clustering is not compatible with following commands which given a
user's confirmation upon enabling clustering, can be removed automatically
from running-config.
policy-map global_policy
 class scansafe-http
  inspect scansafe http-map fail-close
policy-map global_policy
 class scansafe-https
  inspect scansafe https-map fail-close
INFO: No manually-correctable incompatible configuration is found.
```

The following is sample output from the **show cluster info trace** command:

```
> show cluster info trace
 Feb 02 14:19:47.456 [DBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
 Feb 02 14:19:47.456 [DBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
 Feb 02 14:19:47.456 [DBUG]Send CCP message to all: CCP_MSG_KEEPALIVE from 80-1 at MASTER
```

The following is sample output from the **show cluster info flow-mobility counters** command:

```
> show cluster info flow-mobility counters
EID movement notification received  : 0
EID movement notification processed : 0
Flow owner moving requested         : 0
```

See the following outputs for the **show cluster info auto-join** command:

```
> show cluster info auto-join
Unit will try to join cluster in 253 seconds.
Quit reason: Received control message DISABLE

> show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Master has application down that slave has up.

> show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Chassis-blade health check failed.

> show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Service chain application became down.

> show cluster info auto-join
```

```
Unit will try to join cluster when quit reason is cleared.
Quit reason: Unit is kicked out from cluster because of Application health check failure.

> show cluster info auto-join
Unit join is pending (waiting for the smart license entitlement: ent1)

> show cluster info auto-join
Unit join is pending (waiting for the smart license export control flag)
```

Related Commands

Command	Description
show cluster	Displays aggregated data for the entire cluster.

show cluster rule hits

To display rule hit information for all evaluated rules of access control policies and prefilter policies, from all nodes of a cluster in an aggregated format, use the **show cluster rule hits** command.

show cluster rule hits [**raw**]

Syntax Description

raw	(Optional) Displays the rule hit information in .csv format.

Command Default

Displays rule hit information for all the rules from all nodes of a cluster.

Command History

Release	Modification
6.4	This command was introduced.

Usage Guidelines

The rule hit information covers only the access control rules and prefilter rules.

Examples

The following example displays rule hit information from each node of a cluster in a segregated format:

```
> show cluster rule hits
RuleID               Hit Count           First Hit Time(UTC)     Last Hit Time(UTC)
--------------------------------------------------------------------------------------
268435264            1                   06:54:44 Mar 8 2019     06:54:44 Mar 8 2019
268435265            1                   06:54:58 Mar 8 2019     06:54:58 Mar 8 2019
268435270            1                   06:54:53 Mar 8 2019     06:54:53 Mar 8 2019
268435271            1                   06:55:01 Mar 8 2019     06:55:01 Mar 8 2019
268435260            1                   06:55:17 Mar 8 2019     06:55:17 Mar 8 2019
268435261            1                   06:55:19 Mar 8 2019     06:55:19 Mar 8 2019
```

Related Commands

Command	Description
cluster exec show rule hits	Display rule hit information for all evaluated rules of access control policies and prefilter policies from each node of a cluster in a segregated format.
cluster exec clear rule hits	Clears rule hit information for all evaluated rules of access control policies and prefilter policies and reset them to zero, from all nodes in a cluster.
show rule hits	Displays the rule hit information for all evaluated rules of access control policies and prefilter policies.
clear rule hits	Clears the rule hit information for all evaluated rules of access control policies and prefilter policies and resets them to zero.

show community-list

To display routes that are permitted by a specific community list, use the **show community-list** command.

show community-list [*community_list_name*]

Syntax Description

community_list_name	(Optional) Community list name.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show community-list** command:

```
> show community-list

Named Community expanded list comm2
    permit 10
Named Community standard list excomm1
    permit internet 100 no-export no-advertise
```

show conn

To display the connection state for the designated connection type, use the **show conn** command. This command supports IPv4 and IPv6 addresses.

show conn [**vrf** { *name* | **global** }] [**count** | [**all**] [**detail**] [**data-rate-filter** { **lt** | **eq** | **gt** } **value** }]] [**long**] [**state** *state_type*] [**flow-rule**] [**inline-set**] [**protocol** { **tcp** | **udp** | **sctp** }] [**address** *src_ip* [– *src_ip*] [**netmask** *mask*]] [**port** *src_port* [– *src_port*]] [**address** *dest_ip* [– *dest_ip*] [**netmask** *mask*]] [**port** *dest_port* [– *dest_port*]] [**state** *state_type*] [**zone** [*zone_name*]] [**data-rate**]

Syntax Description

address {*src_ip* | *dest_ip*}	(Optional) Displays connections with the specified source or destination IPv4 or IPv6 address. To specify a range, separate the IP addresses with a dash (-). For example, 10.1.1.1-10.1.1.5.
all	(Optional) Displays connections that are to the device or from the device, in addition to through-traffic connections.
count	(Optional) Displays the number of active connections.
detail	(Optional) Displays connections in detail, including translation type and interface information.
data-rate-filter {lt | eq | gt}*value*	(Optional) Displays connections that are filtered based on a data-rate value (bytes per second). For example: *data-rate-filter gt 123*
flow-rule	(Optional) Displays connections of a flow rule.
inline-set	(Optional) Displays connections of an inline-set.
long	(Optional) Displays connections in long format.
netmask *mask*	(Optional) Specifies a subnet mask for use with the given IP address.
port {*src_port* | *dest_port*}	(Optional) Displays connections with the specified source or destination port. To specify a range, separate the port numbers with a dash (-). For example, 1000-2000.
protocol {**tcp | udp** | **sctp**}	(Optional) Specifies the connection protocol.
state *state_type*	(Optional) Specifies the connection state type. See the table in the usage section for a list of the keywords available for connection state types.
zone [*zone_name*]	(Optional) Displays connections for a zone. The **long** and **detail** keywords show the primary interface on which the connection was built and the current interface used to forward the traffic.

[**vrf** {*name* | **global**}]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. Specify **vrf global** to limit the command to the global virtual router. If you omit this keyword, the command applies to all virtual routers.
data-rate	(Optional) Displays whether data-rate tracking status is enabled or disabled.

Command Default

All through connections are shown by default. You need to use the **all** keyword to also view management connections to the device.

Command History

Release	Modification
6.1	This command was introduced.
6.4	The **egress_optimization** connection state type was added.
6.5	Dead Connection Detection (DCD) initiator/responder probe counts were added to the **show conn detail** output for DCD-enabled connections.
6.6	The following changes were introduced: • The **vrf** keyword was added. Connection data-rate tracking status was added. The **data-rate-filter** keyword was added to the `show conn detail` command to filter the connections by user-specified data rate value. • The **packet id** parameter in the **show conn detail** command output was changed to **Connection lookup keyid**.
6.7	The B flag to the command output was added to indicate that the tcp flow is used for obtaining the TLS server certificate.
7.2	The N flag to the command output was enhanced to include 3, 4, 5, 7 and 8 to indicate elephant flow connections and the action taken on them.
7.3	The Q flag, for the QUIC protocol, was added.
7.4	The N flag to the command output was enhanced to include 7 and 8.

Usage Guidelines

The **show conn** command displays the number of active TCP and UDP connections, and provides information about connections of various types. Use the **show conn all** command to see the entire table of connections. You can use this command to find the live connections that are being rate limited by a specific QoS rule ID.

Note When the threat defense device creates a pinhole to allow secondary connections, this is shown as an incomplete conn by the **show conn** command. To clear this incomplete conn use the **clear conn** command.

Note In Firepower 3100 and Cisco Secure Firewall 1200 model devices, though the priority queue is disabled and no connection is established, the output of the **show conn details** displays a note that mentions an invalid connection due to an invalid internal-data/Rx-ring number.

The connection types that you can specify using the **show conn state** command are defined in the following table. When specifying multiple connection types, use commas without spaces to separate the keywords. The following example displays information about RPC, H.323, and SIP connections in the Up state:

```
> show conn state up,rpc,h323,sip
```

Table 5: Connection State Types

Keyword	Connection Type Displayed
up	Connections in the up state.
conn_inbound	Do not use this keyword. It does not show inbound connections correctly.
ctiqbe	CTIQBE connections
data_in	Inbound data connections.
data_out	Outbound data connections.
egress_optimization	Displays information about connections eligible for egress optimization, a feature that enhances performance. Use this command on the advice of Cisco TAC. This command uses flags **F** (only the forward flow is eligible for egress optimization), **R** (only the reverse flow is eligible), or **FR** (both forward and reverse flows are eligible).
finin	FIN inbound connections.
finout	FIN outbound connections.
h225	H.225 connections
h323	H.323 connections
http_get	HTTP get connections.
mgcp	MGCP connections.
nojava	Connections that deny access to Java applets.
rpc	RPC connections.
service_module	Connections being scanned by an SSM.
sip	SIP connections.
skinny	SCCP connections.

Keyword	Connection Type Displayed
smtp_data	SMTP mail data connections.
sqlnet_fixup_data	SQL*Net data inspection engine connections.
tcp_embryonic	TCP embryonic connections.
vpn_orphan	Orphaned VPN tunneled flows.

When you use the **detail** option, the system displays information about the translation type and interface information using the connection flags defined in the following table.

Table 6: Connection Flags

Flag	Description
a	awaiting initiator ACK to SYN
A	awaiting responder ACK to SYN
b	TCP state bypass or nailed
B	TCP probe for server certificate
C	Computer Telephony Interface Quick Buffer Encoding (CTIQBE) media connection
c	cluster centralized
d	dump
D	DNS
E	outside back connection. This is a secondary data connection that must be initiated from the inside host. For example, using FTP, after the inside client issues the PASV command and the outside server accepts, the threat defense preallocates an outside back connection with this flag set. If the inside client attempts to connect back to the server, then the threat defense denies this connection attempt. Only the outside server can use the preallocated secondary connection.
e	semi-distributed
f	initiator FIN
F	responder FIN
g	Media Gateway Control Protocol (MGCP) connection
G	group The G flag indicates the connection is part of a group. It is set by the GRE and FTP Strict inspections to designate the control connection and all its associated secondary connections. If the control connection terminates, then all associated secondary connections are also terminated.
h	H.225
H	H.323

Flag	Description
i	incomplete TCP or UDP connection
I	initiator data
j	GTP data
J	GTP
k	Skinny Client Control Protocol (SCCP) media connection
K	GTP t3-response
L	Outer flow to be decapsulated
m	SIP media connection
M	SMTP data
n	GUP (gatekeeper update protocol)
N	Inspected by Snort. If the system is configured to preserve connections if Snort goes down (this is enabled by default), the N flag includes a number. See the **configure snort** command for more information. • 1—This connection will be preserved if Snort goes down. • 2—Snort did go down, and this connection was preserved. The connection will no longer be inspected by Snort. • 3—Indicates the connections pertain to elephant flow. • 4—The Snort inspection was bypassed for the elephant flows. • 5—The dynamic rate limit policy (10% reduction) was applied on the elephant flows. • 6—The Snort inspection was exempted for the elephant flows. **Note** Elephant flows exemption flag was introduced in 7.4.0. Hence, this flag will not be present in 7.2.0 devices. • 7—This connection is fast-forwarded by Snort and inspected by the data plane engine. • 8—Packets flowing through this connection are dropped due to Snort being either busy or down.
o	Off-loaded flow.
O	responder data
p	passenger flow

Flag	Description
P	inside back connection. This is a secondary data connection that must be initiated from the inside host. For example, using FTP, after the inside client issues the PORT command and the outside server accepts, the threat defense device preallocates an inside back connection with this flag set. If the outside server attempts to connect back to the client, then the device denies this connection attempt. Only the inside client can use the preallocated secondary connection.
q	SQL*Net data
Q	QUIC protocol.
r	Initiator acknowledged FIN . This flag appears when the initiator's FIN is acknowledged by the responder.
R	Responder acknowledged FIN for TCP connection. This flag appears when the responder's FIN is acknowledged by the initiator.
R	UDP RPC. Because each row of **show conn** command output represents one connection (TCP or UDP), there will be only one R flag per row.
t	SIP transient connection. For UDP connections, the value t indicates that it will timeout after one minute.
T	SIP connection. For UDP connections, the value T indicates that the connection will timeout according to the value specified using the **timeout sip** command.
U	up
v	M3UA connection
V	VPN orphan
W	WAAS
w	For inter-chassis clustering on the Firepower 9300, identifies a flow on a backup owner on a separate chassis.
X	Inspected by a service module.
x	per session
y	For clustering, identifies a backup stub flow.
Y	For clustering, identifies a director stub flow.
z	For clustering, identifies a forwarder stub flow.
Z	Scansafe redirection

Note For connections using a DNS server, the source port of the connection may be replaced by the IP address of DNS server in the **show conn** command output.

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by *app_id*, and the idle timer for each app_id runs independently.

Because the app_id expires independently, a legitimate DNS response can only pass through the threat defense device within a limited period of time and there is no resource build-up. However, when you enter the **show conn** command, you will see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

Note When there is no TCP traffic for the period of connection inactivity timeout (by default, 1:00:00), the connection is closed and the corresponding conn flag entries are no longer displayed.

If a LAN-to-LAN/Network-Extension Mode tunnel drops and does not come back, there might be a number of orphaned tunnel flows. These flows are not torn down as a result of the tunnel going down, but all the data attempting to flow through them is dropped. The **show conn** command output shows these orphaned flows with the V flag.

When you use the **count** option in Versions 6.2.0.2, and 6.2.3 or later, the system displays information about the number of connections using the statuses defined in the following table.

Table 7: Connection Status

Status	Description
enabled	Connections for which preserve-connection is currently enabled.
in effect	Connections for which preserve-connection is currently in effect.
most enabled	The most number of connections ever preserved.
most in effect	The most number of connections simultaneously preserved.

Use the **data-rate** keyword to view the current state of the connection data rate tracking feature—enabled or disabled. Use the **data-rate filter** keyword to filter the connections based on the data-rate value in bytes per second. Use the relational operators (lesser than, equal to, or greater than) to filter the connections data. The output displays the active connections along with two data rate values—instantaneous one-second and maximum data rate, for both forward and reverse flows.

Examples

The following is sample output from the **show conn** command. This example shows a TCP session connection from inside host 10.1.1.15 to the outside Telnet server at 10.10.49.10. Because there is no B flag, the connection is initiated from the inside. The “U”, “I”, and “O” flags denote that the connection is active and has received inbound and outbound data.

```
> show conn
```

```
54 in use, 123 most used
TCP out 10.10.49.10:23 in 10.1.1.15:1026 idle 0:00:22, bytes 1774, flags UIO
UDP out 10.10.49.10:31649 in 10.1.1.15:1028 idle 0:00:14, bytes 0, flags D-
TCP dmz 10.10.10.50:50026 inside 192.168.1.22:5060, idle 0:00:24, bytes 1940435, flags UTIOB
TCP dmz 10.10.10.50:49764 inside 192.168.1.21:5060, idle 0:00:42, bytes 2328346, flags UTIOB
TCP dmz 10.10.10.51:50196 inside 192.168.1.22:2000, idle 0:00:04, bytes 31464, flags UIB
TCP dmz 10.10.10.51:52738 inside 192.168.1.21:2000, idle 0:00:09, bytes 129156, flags UIOB
TCP dmz 10.10.10.50:49764 inside 192.168.1.21:0, idle 0:00:42, bytes 0, flags Ti
TCP outside 192.168.1.10(20.20.20.24):49736 inside 192.168.1.21:0, idle 0:01:32, bytes 0,
flags Ti
TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:00:24, bytes 0, flags Ti
TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:01:34, bytes 0,
flags Ti
TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:02:24, bytes 0, flags Ti
TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:03:34, bytes 0,
flags Ti
TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:04:24, bytes 0, flags Ti
TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:05:34, bytes 0,
flags Ti
TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:06:24, bytes 0, flags Ti
TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:07:34, bytes 0,
flags Ti
```

The following is sample output from the **show conn count** command:

```
> show conn count
30 in use, 3194964 most used
Cluster:
        fwd connections: 1 in use, 52 most used
        dir connections: 7 in use, 43826206 most used
        centralized connections: 0 in use, 15 most used
Inspect Snort:
       preserve-connection: 100 enabled, 80 in effect, 400 most enabled, 300 most in effect
```

The following is sample output from the **show conn detail** command. This example shows a UDP connection from outside host 10.10.49.10 to inside host 10.1.1.15. The D flag denotes that this is a DNS connection. The number 1028 is the DNS ID over the connection.

```
> show conn detail
2 in use, 39 most used
Inspect Snort:
        preserve-connection: 2 enabled, 0 in effect, 39 most enabled, 0 most in effect
Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,
       b - TCP state-bypass or nailed,
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - initiator FIN, f - responder FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media
       N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in
 effect)
       n - GUP, O - responder data, o - offloaded,
       P - inside back connection, p - passenger flow
       q - SQL*Net data, R - initiator acknowledged FIN,
       R - UDP SUNRPC, r - responder acknowledged FIN,
       T - SIP, t - SIP transient, U - up,
       V - VPN orphan, v - M3UA W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
```

```
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow

TCP out: 151.101.128.134/443 in: 192.168.1.9/51570,
    flags UfrxIO N1, idle 39s, uptime 10m39s, timeout 10m0s, bytes 4698, xlate id
0x2b8a6ec9b140
  Initiator: 192.168.1.9, Responder: 151.101.128.134
  Connection lookup keyid: 23610071

TCP out: 151.101.120.134/443 in: 192.168.1.9/51568,
    flags UfrxIO N1, idle 39s, uptime 10m40s, timeout 10m0s, bytes 5564, xlate id
0x2b8a6ec9ad40
  Initiator: 192.168.1.9, Responder: 151.101.120.134
 Connection lookup keyid: 23388003
```

The following is sample output from the **show conn** command when an orphan flow exists, as indicated by the V flag:

```
> show conn
16 in use, 19 most used
TCP out 192.168.110.251:7393 in 192.168.150.252:21 idle 0:00:00, bytes 1048, flags UOVB
TCP out 192.168.110.251:21137 in 192.168.150.252:21 idle 0:00:00, bytes 1048, flags UIOB
```

To limit the report to those connections that have orphan flows, add the **vpn_orphan** option to the **show conn state** command, as in the following example:

```
> show conn state vpn_orphan
14 in use, 19 most used
TCP out 192.168.110.251:7393 in 192.168.150.252:5013, idle 0:00:00, bytes 2841019, flags
UOVB
```

For clustering, to troubleshoot the connection flow, first see connections on all units by entering the **cluster exec show conn** command on the master unit. Look for flows that have the following flags: director (Y), backup (y), and forwarder (z). The following example shows an SSH connection from 172.18.124.187:22 to 192.168.103.131:44727 on all three devices; threat defense1 has the z flag showing it is a forwarder for the connection, threat defense3 has the Y flag showing it is the director for the connection, and threat defense2 has no special flags showing it is the owner. In the outbound direction, the packets for this connection enter the inside interface on threat defense2 and exit the outside interface. In the inbound direction, the packets for this connection enter the outside interface on threat defense1 and threat defense3, are forwarded over the cluster control link to threat defense2, and then exit the inside interface on threat defense2.

```
> cluster exec show conn
FTD1(LOCAL):**************************************************
18 in use, 22 most used
Cluster stub connections: 0 in use, 5 most used
TCP outside  172.18.124.187:22 inside  192.168.103.131:44727,
idle 0:00:00, bytes 37240828, flags z
FTD2:**************************************************
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
TCP outside  172.18.124.187:22 inside  192.168.103.131:44727,
idle 0:00:00, bytes 37240828, flags UIO
FTD3:**************************************************
10 in use, 12 most used
Cluster stub connections: 2 in use, 29 most used
TCP outside  172.18.124.187:22 inside  192.168.103.131:44727,
```

```
idle 0:00:03, bytes 0, flags  Y
```

The output of show conn detail on threat defense2 shows that the most recent forwarder was threat defense1:

```
> show conn detail
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       b - TCP state-bypass or nailed,
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, L - LISP triggered flow owner mobility,
       M - SMTP data, m - SIP media, n - GUP
       O - outbound data, o - offloaded,
       P - inside back connection,
       Q - Diameter, q - SQL*Net data,
       R - outside acknowledged FIN,
       R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
       V - VPN orphan, W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow
TCP outside: 172.18.124.187/22 inside: 192.168.103.131/44727,
    flags UIO , idle 0s, uptime 25s, timeout 1h0m, bytes 1036044,
cluster sent/rcvd bytes 0/1032983, cluster sent/rcvd total bytes 0/1080779, owners (1,255)
Traffic received at interface outside
        Locally received: 0 (0 byte/s)
        From most recent forwarder FTD1: 1032983 (41319 byte/s)
Traffic received at interface inside
        Locally received: 3061 (122 byte/s)
```

When you use the **detail** keyword, you can see information about Dead Connection Detection (DCD) probing, which shows how often the connection was probed by the initiator and responder. For example, the connection details for a DCD-enabled connection would look like the following:

```
TCP dmz: 10.5.4.11/5555 inside: 10.5.4.10/40299,
    flags UO , idle 1s, uptime 32m10s, timeout 1m0s, bytes 11828,
cluster sent/rcvd bytes 0/0, owners (0,255)
  Traffic received at interface dmz
        Locally received: 0 (0 byte/s)
  Traffic received at interface inside
        Locally received: 11828 (6 byte/s)
  Initiator: 10.5.4.10, Responder: 10.5.4.11
  DCD probes sent: Initiator 5, Responder 5
```

The following example shows how to view the status of connection data-rate tracking feature:

```
ciscoasa# show conn data-rate
Connection data rate tracking is currently enabled.
```

The following example shows how to filter the connection based on a specified data-rate:

```
firepower# show conn detail data-rate-filter ?
eq  Enter this keyword to show conns with data-rate equal to specified value
gt  Enter this keyword to show conns with data-rate greater than specified value
```

```
lt Enter this keyword to show conns with data-rate less than specified value
firepower# show conn detail data-rate-filter gt ?
<0-4294967295> Specify the data rate value in bytes per second
firepower# show conn detail data-rate-filter gt 123 | grep max rate
max rate: 3223223/399628 bytes/sec
max rate: 3500123/403260 bytes/sec
```

Following example is the output of **show conn** and **show conn detail** with the B flag. The B flag indicates that the TCP flow is used to obtain the TLS1.3 server certificate. When a request for TLS 1.3 certificate is obtained from the client to threat defense connection, another connection is established between the TLS 1.3 server and the threat defense. Thus, one connection is established between the threat defense and the client; another connection is established between the TLS 1.3 server and the threat defense.

```
>show conn
1 in use, 3 most used
Inspect Snort:
       preserve-connection: 1 enabled, 0 in effect, 1 most enabled, 0 most in effect
 TCP outside  33.33.33.2:80 inside  1.1.1.2:35226, idle 0:00:00, bytes 246324931, flags
UIOBN1

> show conn detail
1 in use, 3 most used
Inspect Snort:
       preserve-connection: 1 enabled, 0 in effect, 1 most enabled, 0 most in effect
Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,
       b - TCP state-bypass or nailed,
       B - TCP probe for server certificate
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - initiator FIN, f - responder FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media
       N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in
 effect)
       n - GUP, O - responder data, o - offloaded,
       P - inside back connection, p - passenger flow
       q - SQL*Net data, R - initiator acknowledged FIN,
       R - UDP SUNRPC, r - responder acknowledged FIN,
       T - SIP, t - SIP transient, U - up,
       V - VPN orphan, v - M3UA W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow

TCP outside: 33.33.33.2/80 inside: 1.1.1.2/35226,
    flags UIOBN1, idle 0s, uptime 12s, timeout 1h0m, bytes 698500915
  Initiator: 1.1.1.2, Responder: 33.33.33.2
  Connection lookup keyid: 865399
```

The following is sample output from the **show conn detail** command. This example shows N4, indicating that the snort inspection was bypassed for the Elephant Flow.

```
> show conn detail
0 in use, 19 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect
Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,
       B - TCP probe for server certificate,
       b - TCP state-bypass or nailed,
```

```
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - initiator FIN, f - responder FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media
       N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in
 effect,
            3 - elephant-flow, 4 - elephant-flow bypassed, 5 - elephant-flow throttled)
       n - GUP, O - responder data, o - offloaded,
       P - inside back connection, p - passenger flow
       q - SQL*Net data, R - initiator acknowledged FIN,
       R - UDP SUNRPC, r - responder acknowledged FIN,
       T - SIP, t - SIP transient, U - up,
       V - VPN orphan, v - M3UA W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow

TCP outside_https: 172.16.4.1/80 inside_https: 172.16.77.1/38992,
    flags UIO N1N4, idle 0s, uptime 2m24s, timeout 1h0m, bytes 1891172595
  Initiator: 172.16.77.1, Responder: 172.16.4.1
  Connection lookup keyid: 1556755610
```

This example shows N5 in the output to indicate dynamic rate limit policy (10% reduction) was applied on the Elephant Flow.

```
> show conn detail
0 in use, 19 most used
Inspect Snort:
preserve-connection: 0 enabled, 0 in effect, 1 most enabled, 0 most in effect
Flags: A - awaiting responder ACK to SYN, a - awaiting initiator ACK to SYN,
       B - TCP probe for server certificate,
       b - TCP state-bypass or nailed,
       C - CTIQBE media, c - cluster centralized,
       D - DNS, d - dump, E - outside back connection, e - semi-distributed,
       F - initiator FIN, f - responder FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - initiator data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, L - decap tunnel, M - SMTP data, m - SIP media
       N - inspected by Snort (1 - preserve-connection enabled, 2 - preserve-connection in
 effect,
            3 - elephant-flow, 4 - elephant-flow bypassed, 5 - elephant-flow throttled)
       n - GUP, O - responder data, o - offloaded,
       P - inside back connection, p - passenger flow
       q - SQL*Net data, R - initiator acknowledged FIN,
       R - UDP SUNRPC, r - responder acknowledged FIN,
       T - SIP, t - SIP transient, U - up,
       V - VPN orphan, v - M3UA W - WAAS,
       w - secondary domain backup,
       X - inspected by service module,
       x - per session, Y - director stub flow, y - backup stub flow,
       Z - Scansafe redirection, z - forwarding stub flow

TCP outside_https: 172.16.4.1/80 inside_https: 172.16.77.1/38822,
    flags UIO N1N5, qos-rule-id 20000, idle 0s, uptime 4m8s, timeout 1h0m, bytes 585732628
  Initiator: 172.16.77.1, Responder: 172.16.4.1
  Connection lookup keyid: 1933458538
```

Related Commands

Commands	Description
clear conn	Clears connections.
clear conn data-rate	Clears the current maximum data-rate stored.

show console-output

To display the currently captured console output, use the **show console-output** command.

show console-output

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show console-output** command.

```
> show console-output
Message #1 : Message #2 : Setting the offload CPU count to 0
Message #3 :
Compiled on Fri 20-May-16 13:36 PDT by builders
Message #4 :
Total NICs found: 14
Message #5 : i354 rev03 Gigabit Ethernet @ irq255 dev 20 index 08 MAC: e865.49b8.97f1
Message #6 : ivshmem rev03 Backplane Data Interface     @ index 09 MAC: 0000.0001.0002
Message #7 : en_vtun rev00 Backplane Control Interface  @ index 10 MAC: 0000.0001.0001
Message #8 : en_vtun rev00 Backplane Int-Mgmt Interface     @ index 11 MAC: 0000.0001.0003
Message #9 : en_vtun rev00 Backplane Ext-Mgmt Interface     @ index 12 MAC: 0000.0000.0000
Message #10 : en_vtun rev00 Backplane Tap Interface     @ index 13 MAC: 0000.0100.0001
Message #11 : Running Permanent Message
#12 : Activation Key: Message
#13 : 0x00000000 Message
#14 : 0x00000000 Message
#15 : 0x00000000 Message
#16 : 0x00000000 Message
#17 : 0x00000000 Message #18 :
Message #19 : The Running Activation Key is not valid, using default settings:
Message #20 :
(...output truncated...)
```

show coredump

To display the setting of packet-engine core dump generation, enter the **show coredump** command.

show coredump

Command Default

Packet-engine coredump generation is enabled by default.

Command History

Release	Modification
6.2.1	This command was introduced.
7.4.1, 7.2.6	This command was modified to display the following additional information about the Snort 3 core dump: • Mode of operation • Information about whether the next crash will produce a full core dump; and if temporarily disabled, when the full core dump will be enabled again.

Usage Guidelines

This command is only available on the Firepower 2100 series. When you run this command on an unsupported platform, the system returns the following message:

```
This command is not available on this platform.
```

Examples

The following example shows that packet-engine coredump generation is enabled:

```
> show coredump

Process Type: Coredump State:
packet-engine enabled
```

The following example shows the core dump state and when the next Snort 3 dump will produce a full core dump:

```
> show coredump

Process Type: Coredump State:
packet-engine enabled
The following programs have core dumps disabled:
None
snort3 will write core dump daily: core dump will be written on the first crash,
after which core dump will be disabled for the next 24 hours.
next snort3 crash will produce a full core dump
```

Related Commands

Command	Description
configure coredump packet-engine	Enables or disables packet-engine core dump generation.
configure coredump snort3	Enables or disables Snort 3 core dump generation.

show counters

To display the protocol stack counters, use the **show counters** command.

show counters [**all** | **summary** | **top** *N*] [**description**] [**detail**] [**protocol** *protocol_name* [**:***counter_name*]] [**threshold** *N*]

Syntax Description

all	Displays the filter details.
:*counter_name*	Specifies a counter by name.
description	Display the various counters and descriptions.
detail	Displays additional counters information.
protocol *protocol_name*	Displays the counters for the specified protocol. Enter ? for a list of options.
summary	Displays a counter summary.
threshold *N*	Displays only those counters at or above the specified threshold. The range is 1 through 4294967295.
top *N*	Displays the counters at or above the specified threshold. The range is 1 through 4294967295.

Command Default

The default is **show counters summary detail threshold 1**.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to display the default information.

```
> show counters
Protocol     Counter                                    Value   Context
IP           IN_PKTS                                   785064   Summary
IP           OUT_PKTS                                   19196   Summary
IP           OUT_DROP_DWN                              177099   Summary
IP           TO_ARP                                    785064   Summary
TCP          OUT_PKTS                                   38378   Summary
TCP          SESS_CTOD                                  19189   Summary
TCP          OUT_CLSD                                   19189   Summary
TCP          HASH_ADD                                   19189   Summary
TCP          SND_SYN                                    19189   Summary
SSLERR       BAD_SIGNATURE                                  3   Summary
SSLDEV       NEW_CTX                                        3   Summary
VPIF         BAD_VALUE                                    673   Summary
VPIF         NOT_FOUND                              106843325   Summary
```

The following is sample output of the counters that are hit during a zero trust flow.

```
> show counters protocol zero_trust
Protocol  Counter                          Value     Context
ZERO_TRUST MAX_USERS_LIMIT                  1         Summary
ZERO_TRUST MAX_SESSIONS_PER_USER_LIMIT      3         Summary
ZERO_TRUST LONG_URL_LIMIT                   4         Summary
ZERO_TRUST DUPLICATE_ASSERTION              2         Summary
ZERO_TRUST DUPLICATE_SESSION                1         Summary
ZERO_TRUST COOKIE_DISABLED_BROWSER          3         Summary
ZERO_TRUST RELAY_STATE_FAILURE              1         Summary
ZERO_TRUST REDIRECTED_FOR_AUTHN             11        Summary
ZERO_TRUST TRAFFIC_ON_WRONG_INTERFACE       2         Summary
ZERO_TRUST NON_ZTNA_REQUEST                 6         Summary
ZERO_TRUST MISSING_URL_DATA                 3         Summary
ZERO_TRUST INVALID_GROUP_URL_PARAMS         3         Summary
ZERO_TRUST RANDOM_GEN_FAILURE               1         Summary
ZERO_TRUST INVALID_COOKIE                   3         Summary
ZERO_TRUST FORM_SUBMISSION_ERRORS           1         Summary
ZERO_TRUST HUGE_PAYLOAD                     1         Summary
```

Counter	Description
MAX_USERS_LIMIT	Number of times the maximum number of users per application limit was reached for a client IP
MAX_SESSIONS_PER_USER_LIMIT	Number of times the maximum number of sessions per user per application limit was reached
LONG_URL_LIMIT	Number of times the URL reached the maximum URL length limit
DUPLICATE_ASSERTION	Number of times duplicate assertion was received
DUPLICATE_SESSION	Number of times duplicate session was received
COOKIE_DISABLED_BROWSER	Number of times cookies were disabled by the browser
RELAY_STATE_FAILURE	Number of times relay state verification failed
REDIRECTED_FOR_AUTHN	Number of times connections were redirected for authentication
TRAFFIC_ON_WRONG_INTERFACE	Number of times traffic was on the wrong interface
NON_ZTNA_REQUEST	Number of non-zero trust requests
MISSING_URL_DATA	Number of times required data was missing in the URL
INVALID_GROUP_URL_PARAMS	Number of times group URL parameters were invalid
RANDOM_GEN_FAILURE	Number of times random number generation failed
INVALID_COOKIE	Number of times invalid cookie was seen
FORM_SUBMISSION_ERRORS	Number of times form submission error was seen

Counter	Description
HUGE_PAYLOAD	Number of times huge payload was seen

The following is a sample output of all HA specific counters prefixed with HA.

```
>show counters protocol zero_trust
Protocol       Counter              Value     Context
ZERO_TRUST HA_COOKIE_TX_SUCCESS      2         Summary
ZERO_TRUST HA_COOKIE_BULK_TX_SUCCESS 2         Summary
ZERO_TRUST HA_GRP_COOKIE_TX_SUCCESS  2         Summary
ZERO_TRUST HA_SALT_TX_SUCCESS        2         Summary
ZERO_TRUST HA_COOKIE_RX_SUCCESS      2         Summary
ZERO_TRUST HA_COOKIE_BULK_RX_SUCESS  2         Summary
ZERO_TRUST HA_GRP_COOKIE_RX_SUCCESS  2         Summary
ZERO_TRUST HA_SALT_RX_SUCCESS        2         Summary
```

Counter	Description
HA_COOKIE_TX_SUCCESS	Cookie messages were successfully sent from the active node
HA_COOKIE_TX_FAILURE	Cookie messages failed to be sent from the active node
HA_COOKIE_RX_SUCCESS	Cookie messages were successfully replicated on the standby node
HA_COOKIE_RX_FAILURE	Cookie messages failed to replicate on the standby node
HA_COOKIE_BULK_TX_SUCCESS	Cookie bulk sync messages were successfully sent from the active node
HA_COOKIE_BULK_TX_FAILURE	Cookie bulk sync messages failed to sent from the active node
HA_COOKIE_BULK_RX_SUCCESS	Cookie bulk sync replication was successful on the standby node
HA_COOKIE_BULK_RX_FAILURE	Cookie bulk sync replication failed on the standby node
HA_GRP_COOKIE_TX_SUCCESS	Group cookie messages were successfully sent from the active node
HA_GRP_COOKIE_TX_FAILURE	Group cookie messages failed to be sent from the active node
HA_GRP_COOKIE_RX_SUCCESS	Group cookie messages were successfully replicated on the standby node
HA_GRP_COOKIE_RX_FAILURE	Group cookie messages failed to replicate on the standby node
HA_SALT_TX_SUCCESS	Salt messages were successfully sent from the active node

Counter	Description
HA_SALT_TX_FAILURE	Salt messages failed to be sent from the active node
HA_SALT_RX_SUCCESS	Salt replication was successful on the standby node
HA_SALT_RX_FAILURE	Salt replication failed on the standby node

The following is a sample output of all cluster specific counters prefixed with CLUSTER.

```
> show counters protocol zero_trust
Protocol    Counter                      Value Context
ZERO_TRUST CLUSTER_COOKIE_TX_SUCCESS      2  Summary
ZERO_TRUST CLUSTER_COOKIE_TX_FAILURE      1  Summary
ZERO_TRUST CLUSTER_COOKIE_RX_SUCCESS      2  Summary
ZERO_TRUST CLUSTER_COOKIE_RX_FAILURE      3  Summary
ZERO_TRUST CLUSTER_COOKIE_BULK_TX_SUCCESS 2  Summary
ZERO_TRUST CLUSTER_COOKIE_BULK_TX_FAILURE 2  Summary
ZERO_TRUST CLUSTER_COOKIE_BULK_RX_SUCCESS 2  Summary
ZERO_TRUST CLUSTER_COOKIE_BULK_RX_FAILURE 2  Summary
ZERO_TRUST CLUSTER_GRP_COOKIE_TX_SUCCESS  3  Summary
ZERO_TRUST CLUSTER_GRP_COOKIE_TX_FAILURE  5  Summary
ZERO_TRUST CLUSTER_GRP_COOKIE_RX_SUCCESS  3  Summary
ZERO_TRUST CLUSTER_GRP_COOKIE_RX_FAILURE  3  Summary
ZERO_TRUST CLUSTER_SALT_TX_SUCCESS        4  Summary
ZERO_TRUST CLUSTER_SALT_TX_FAILURE        4  Summary
ZERO_TRUST CLUSTER_SALT_TX_SUCCESS        9  Summary
ZERO_TRUST CLUSTER_SALT_RX_FAILURE        4  Summary
```

Counter	Description
CLUSTER_COOKIE_TX_SUCCESS	Cookie messages were successfully sent from the control node
CLUSTER_COOKIE_TX_FAILURE	Cookie messages failed to be sent from the control node
CLUSTER_COOKIE_RX_SUCCESS	Cookie messages were successfully replicated to the data nodes
CLUSTER_COOKIE_RX_FAILURE	Cookie messages failed to replicate on the data nodes
CLUSTER_COOKIE_BULK_TX_SUCCESS	Bulk sync messages were successfully sent from the control node
CLUSTER_COOKIE_BULK_TX_FAILURE	Bulk sync messages failed to be sent from the control node
CLUSTER_COOKIE_BULK_RX_SUCCESS	Successful bulk syncs on the data nodes
CLUSTER_COOKIE_BULK_RX_FAILURE	Bulk sync failed on the data nodes
CLUSTER_GRP_COOKIE_TX_SUCCESS	Group cookie messages were successfully sent from the control node
CLUSTER_GRP_COOKIE_TX_FAILURE	Group cookie messages failed to be sent from the control node

Counter	Description
CLUSTER_GRP_COOKIE_RX_SUCCESS	Group cookie messages were successfully replicated on the data nodes
CLUSTER_GRP_COOKIE_RX_FAILURE	Group cookie messages failed to replicate on the data nodes
CLUSTER_SALT_TX_SUCCESS	Salt messages were successfully sent from the control node
CLUSTER_SALT_TX_FAILURE	Salt message failed to be sent from the control node
CLUSTER_SALT_RX_SUCCESS	Successful salt replications on the data nodes
CLUSTER_SALT_RX_FAILURE	Salt replication failed on the data nodes

Command	Description
clear counters	Clears the protocol stack counters.

show cpu

To display the CPU utilization information, use the **show cpu** command.

show cpu [**detailed** | **external** | **profile** [**dump**] | **system** [*processor_num*]]
show cpu core [**all** | *core_id*]
show cpu usage [**detailed** | **core** [**all** | *core_id*]]

Syntax Description

core [**all** | *core_id*]	Displays CPU statistics for each core. You can view all cores (the default) or specify a core by number. Use the keyword without a parameter to see the core numbers available on your device. Core numbers start at 0. The **show cpu core** and **show cpu usage core** commands provide the same information. **Note** On Secure Firewall 4200 series devices, core 0 is dedicated for control point, while the other cores are used to execute the data path processes.
detailed	(Optional) Displays the CPU usage internal details.
external	(Optional) Displays CPU usage for external processes.
profile [**dump**]	(Optional) Displays the CPU profiling data. Include the **dump** keyword to see a dump of the profiling data.
system [*processor_num*]	(Optional) Displays information related to the whole system. You can optionally include a processor number to see information for a specific processor. Use the command without the keyword to see the number of available processors, which are called CPUs. Processor numbers start at 0. Thus, if the output shows there are 8 CPUs, the valid numbers for your system are 0-7.
usage	(Optional) Displays the CPU usage. This is the default option.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The CPU usage is computed using an approximation of the load every five seconds, and by further feeding this approximation into two, following moving averages.

You can use the **show cpu profile dump** command in conjunction with the **cpu profile activate** command to collect information for TAC use in troubleshooting CPU issues. The **show cpu profile dump** command output is in hexadecimal format.

For the **detailed** and **core** views, it is not unusual to see a core with zero usage when overall CPU usage is low.

For the threat defense virtual, the **show cpu** command also shows whether the number of CPUs allotted to the VM is within the allowed limit based on the vCPU platform license limit. The status can be Compliant,

Noncompliant: Over-provisioned, or Noncompliant: Under-provisioned. This information might not be accurate.

Examples

The following example shows how to display the CPU utilization:

```
> show cpu
CPU utilization for 5 seconds = 18%; 1 minute: 18%; 5 minutes: 18%
```

The following example shows how to display detailed CPU utilization information:

```
> show cpu detailed
Break down of per-core data path versus control point cpu usage:
Core         5 sec            1 min            5 min
Core 0       0.0 (0.0 + 0.0)  3.3 (0.0 + 3.3)  2.4 (0.0 + 2.4)
Current control point elapsed versus the maximum control point elapsed for:
      5 seconds = 99.0%; 1 minute: 99.8%; 5 minutes: 95.9%
CPU utilization of external processes for:
      5 seconds = 0.2%; 1 minute: 0.0%; 5 minutes: 0.0%
Total CPU utilization for:
      5 seconds = 0.2%; 1 minute: 3.3%; 5 minutes: 2.5%
```

Note The "Current control point elapsed versus the maximum control point elapsed for" statement means that the current control point load is compared to the maximum load seen within the defined time period. This is a ratio instead of an absolute number. The figure of 99% for the 5-second interval means that the current control point load is at 99% of the maximum load that is visible over this 5-second interval. If the load continues to increase all the time, then it will always remain at 100%. However, the actual CPU may still have a lot of free capacity because the maximum absolute value has not been defined.

The following example shows how to display system-level CPU usage. Note the "(2 CPU)" indication in the first line. This is the number of processors on this device.

```
> show cpu system
Linux 3.10.62-ltsi-WR6.0.0.27_standard (ftd1.example.com)        10/20/16        _x86_64_    (2 CPU)

Time         CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
15:48:26     all   50.36    0.00   10.04    0.78    0.00    0.03    0.00    0.00    0.00   38.79
```

The following table explains the fields in the **show cpu system** output.

Table 8: Show CPU System Fields

Field	Description
Time	The time when these numbers were determined.
CPU	Processor number.

Field	Description
%user	Percentage of CPU utilization that occurred while executing at the user level (application).
%nice	Percentage of CPU utilization that occurred while executing at the user level with nice priority.
%sys	Percentage of CPU utilization that occurred while executing at the system level (kernel). This does not include time spent servicing interrupts or softirqs. A softirq (software interrupt) is one of up to 32 enumerated software interrupts that can run on multiple CPUs at once.
%iowait	Percentage of time that the CPUs were idle when the system had an outstanding disk I/O request.
%irq	Percentage of time spent by the CPUs to service interrupts.
%soft	Percentage of time spent by the CPUs to service softirqs.
%steal	Percentage of time spent in involuntary wait by the virtual CPUs while the hypervisor was servicing another virtual processor.
%guest	Percentage of time spent by the CPUs to run a virtual processor.
%gnice	Percentage of CPU utilization that occurred while executing at the guest level with nice priority for a virtual processor.
%idle	Percentage of time that the CPUs were idle and the system did not have an outstanding disk I/O request.

The following example activates the profiler and instructs it to store 1000 samples, the default. Next, the **show cpu profile** command shows that the profiling is in progress. After waiting some time, the next **show cpu profile** command shows that profiling has completed. Finally, we use the **show cpu profile dump** command to get the results. Copy the output and provide it to Cisco Technical Support. You might need to log your SSH session to get the full output.

```
> cpu profile activate
Activated CPU profiling for 1000 samples.
Use "show cpu profile" to display the progress or "show cpu profile dump" to interrupt
profiling and display the incomplete results.
> show cpu profile
CPU profiling started: 16:13:48.279 UTC Thu Oct 20 2016
CPU profiling currently in progress:
    Core 0: 501 out of 1000 samples collected.
    CP: 586 out of 1000 samples collected.
Use "show cpu profile dump" to see the results after it is complete or to interrupt
profiling and display the incomplete results.
> show cpu profile
CPU profiling started: 16:13:48.279 UTC Thu Oct 20 2016
CPU Profiling has stopped.
    Core 0 done with 1000 samples
    CP done with 1000 samples
Use "show cpu profile dump" to see the results.
> show cpu profile dump
(...output omitted...)
```

Related Commands

Command	Description
clear cpu profile	Clears CPU profiling data.
cpu profile activate	Activates CPU profiling.
show counters	Displays the protocol stack counters.

show crashinfo

To display the contents of the crash file stored in Flash memory, enter the **show crashinfo** command.

show crashinfo [**console** | **module** *number* | **save** | **webvpn** [**detailed**]]

Syntax Description

console	(Optional) Show the status of crashinfo console output.
module *number*	(Optional) Displays crash information retrieved from the specified module. Indicate the module by number, for example, 1.
save	(Optional) Displays whether the device is configured to save crash information to Flash memory.
webvpn [**detailed**]	(Optional) Displays threat defense crash recovery dumps.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

If the crash file is from a test crash (generated from the **crashinfo test** command), the first string of the crash file is ": Saved_Test_Crash" and the last string is ": End_Test_Crash". If the crash file is from a real crash, the first string of the crash file is ": Saved_Crash" and the last string is ": End_Crash". (This includes crashes from use of the **crashinfo force page-fault** or **crashinfo force watchdog** commands).

Compliance with FIPS 140-2 prohibits the distribution of Critical Security Parameters (keys, passwords, etc.) outside of the crypto boundary (chassis). When the device crashes, due to an assert or checkheaps failure, it is possible that the stack or memory regions dumped to the console contain sensitive data. This output must be suppressed in FIPS-mode.

Examples

The following example shows that there are no crashinfo information.

```
> show crashinfo
------------------ show crashinfo module 1 ------------------
INFO: This module has no crashinfo available.
```

The following example shows how to display the current crash information configuration:

```
> show crashinfo save
crashinfo save enable
```

The following example shows the status of crashinfo console output.

```
> show crashinfo console
crashinfo console enable
```

The following example shows the output for a crash file test. This test does not actually crash the threat defense device. It provides a simulated example file.

```
> crashinfo test
> show crashinfo
: Saved_Test_Crash
Thread Name: ci/console (Old pc 0x001a6ff5 ebp 0x00e88920)
Traceback:
0: 00323143
1: 0032321b
2: 0010885c
(...Remaining output truncated...)
```

Related Commands

Command	Description
clear crashinfo	Deletes the contents of the crash file.
crashinfo force	Forces a crash of the threat defense device.
crashinfo test	Tests the ability of the threat defense device to save crash information to a file in flash memory.

show crypto accelerator load-balance

To display the accelerator-specific load-balancing information from the hardware crypto accelerator MIB, use the **show crypto accelerator load-balance** command.

show crypto accelerator load-balance [**ipsec** | **ssl** | **detail** [**ipsec** | **ssl**]]

Syntax Description

detail	(Optional) Displays detailed information. You can include the **ipsec** or **ssl** keyword after this option.
ipsec	(Optional) Displays crypto accelerator IPSec load balancing details.
ssl	(Optional) Displays crypto accelerator SSL load balancing details.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows global crypto accelerator load balancing statistics:

```
> show crypto accelerator load-balance

                 Crypto IPSEC Load Balancing Stats:
                 ==================================

Engine      Crypto Cores             IPSEC Sessions          Active Session
                                                             Distribution (%)
======     ==============      ============================   ================
  0        IPSEC 1, SSL 1      Total:     0  Active:     0         0.0%

Commands Completed       1 second         5 second         60 second
==================       ========         ========         =========
 Engine 0 (load)            0.0%             0.0%             0.0%

Encrypted Data           1 second         5 second         60 second
==================       ========         ========         =========
 Engine 0 (load)            0.0%             0.0%             0.0%

Decrypted Data           1 second         5 second         60 second
==================       ========         ========         =========
 Engine 0 (load)            0.0%             0.0%             0.0%

Engine 0 Per Core Load Balancing Stats:
=======================================

Commands Completed       1 second         5 second        60 second
==================       ========         ========         =========
 IPSec ring 0   (load)      0.0%             0.0%             0.0%

Encrypted Data           1 second         5 second         60 second
==================       ========         ========         =========
 IPSec ring 0   (load)      0.0%             0.0%             0.0%
```

```
Decrypted Data           1 second         5 second        60 second
==================       ========         ========         =========
 IPSec ring 0   (load)      0.0%             0.0%             0.0%

                 Crypto SSL Load Balancing Stats:
                 ================================

Engine      Crypto Cores             SSL Sessions            Active Session
                                                             Distribution (%)
======     ==============      ============================   ================
  0        IPSEC 1, SSL 1      Total:     0  Active:     0         0.0%

Commands Completed       1 second         5 second         60 second
==================       ========         ========         =========
 Engine 0 (load)            0.0%             0.0%             0.0%

Encrypted Data           1 second         5 second         60 second
==================       ========         ========         =========
 Engine 0 (load)            0.0%             0.0%             0.0%

Decrypted Data           1 second         5 second         60 second
==================       ========         ========         =========
 Engine 0 (load)            0.0%             0.0%             0.0%

Engine 0 Per Core Load Balancing Stats:
=======================================

Commands Completed       1 second         5 second        60 second
==================       ========         ========         =========
 Admin ring 0   (load)      0.0%             0.0%             0.0%

Encrypted Data           1 second         5 second         60 second
==================       ========         ========         =========
 Admin ring 0   (load)      0.0%             0.0%             0.0%

Decrypted Data           1 second         5 second        60 second
==================       ========         ========         =========
 Admin ring 0   (load)      0.0%             0.0%             0.0%
```

Related Commands

Command	Description
clear crypto accelerator statistics	Clears the global and accelerator-specific statistics in the crypto accelerator MIB.
clear crypto protocol statistics	Clears the protocol-specific statistics in the crypto accelerator MIB.
show crypto protocol statistics	Displays the protocol-specific statistics from the crypto accelerator MIB.

show crypto accelerator statistics

To display the global and accelerator-specific statistics from the hardware crypto accelerator MIB, use the **show crypto accelerator statistics** command.

show crypto accelerator statistics

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The output statistics are defined as follows:

Accelerator 0 shows statistics for the software-based crypto engine.

Accelerator 1 shows statistics for the hardware-based crypto engine.

RSA statistics show RSA operations for 2048-bit keys, which are executed in software by default. This means that when you have a 2048-bit key, IKE/SSL VPN performs RSA operations in software during the IPsec/SSL negotiation phase. Actual IPsec/SSL traffic is still processed using hardware. This may cause high CPU if there are many simultaneous sessions starting at the same time, which may result in multiple RSA key operations and high CPU. If you run into a high CPU condition because of this, then you should use a 1024-bit key to process RSA key operations in hardware. To do so, you must reenroll the identity certificate. In releases 8.3(2) or later, you can also use the crypto engine large-mod-accel command on the 5510-5550 platforms to perform these operations in hardware.

If you are using a 2048-bit RSA key and the RSA processing is performed in software, you can use CPU profiling to determine which functions are causing high CPU usage. Generally, the bn_* and BN_* functions are math operations on the large data sets used for RSA, and are the most useful when examining CPU usage during an RSA operation in software. For example:

```
@@@@@@@@@@@@@@@@@@@@............................... 36.50% : _bn_mul_add_words
@@@@@@@@@@.......................................... 19.75% : _bn_sqr_comba8
```

Diffie-Hellman statistics show that any crypto operation with a modulus size greater than 1024 is performed in software (for example, DH5 (Diffie-Hellman group 5 uses 1536)). If so, a 2048-bit key certificate will be processed in software, which can result in high CPU usage when a lot of sessions are running.

DSA statistics show key generation in two phases. The first phase is a choice of algorithm parameters, which may be shared between different users of the system. The second phase computes private and public keys for a single user.

SSL statistics show records for the processor-intensive public key encryption algorithms involved in SSL transactions to the hardware crypto accelerator.

RNG statistics show records for a sender and receiver, which can generate the same set of random numbers automatically to use as keys.

Examples

The following example shows global crypto accelerator statistics:

```
> show crypto accelerator statistics

Crypto Accelerator Status
-------------------------
[Capacity]
   Supports hardware crypto: True
   Supports modular hardware crypto: False
   Max accelerators: 1
   Max crypto throughput: 100 Mbps
   Max crypto connections: 750
[Global Statistics]
   Number of active accelerators: 1
   Number of non-operational accelerators: 0
   Input packets: 700
   Input bytes: 753488
   Output packets: 700
   Output error packets: 0
   Output bytes: 767496
[Accelerator 0]
   Status: Active
   Software crypto engine
   Slot: 0
   Active time: 167 seconds
   Total crypto transforms: 7
   Total dropped packets: 0
   [Input statistics]
      Input packets: 0
      Input bytes: 0
      Input hashed packets: 0
      Input hashed bytes: 0
      Decrypted packets: 0
      Decrypted bytes: 0
   [Output statistics]
      Output packets: 0
      Output bad packets: 0
      Output bytes: 0
      Output hashed packets: 0
      Output hashed bytes: 0
      Encrypted packets: 0
      Encrypted bytes: 0
   [Diffie-Hellman statistics]
      Keys generated: 0
      Secret keys derived: 0
   [RSA statistics]
      Keys generated: 0
      Signatures: 0
      Verifications: 0
      Encrypted packets: 0
      Encrypted bytes: 0
      Decrypted packets: 0
      Decrypted bytes: 0
   [DSA statistics]
      Keys generated: 0
      Signatures: 0
      Verifications: 0
   [SSL statistics]
      Outbound records: 0
      Inbound records: 0
   [RNG statistics]
      Random number requests: 98
      Random number request failures: 0
[Accelerator 1]
   Status: Active
```

```
   Encryption hardware device : Cisco ASA-55x0 on-board accelerator
(revision 0x0)
                            Boot microcode   : CNlite-MC-Boot-Cisco-1.2
                            SSL/IKE microcode: CNlite-MC-IPSEC-Admin-3.03
                            IPsec microcode  : CNlite-MC-IPSECm-MAIN-2.03
   Slot: 1
   Active time: 170 seconds
   Total crypto transforms: 1534
   Total dropped packets: 0
   [Input statistics]
      Input packets: 700
      Input bytes: 753544
      Input hashed packets: 700
      Input hashed bytes: 736400
      Decrypted packets: 700
      Decrypted bytes: 719944
   [Output statistics]
      Output packets: 700
      Output bad packets: 0
      Output bytes: 767552
      Output hashed packets: 700
      Output hashed bytes: 744800
      Encrypted packets: 700
      Encrypted bytes: 728352
   [Diffie-Hellman statistics]
      Keys generated: 97
      Secret keys derived: 1
   [RSA statistics]
      Keys generated: 0
      Signatures: 0
      Verifications: 0
      Encrypted packets: 0
      Encrypted bytes: 0
      Decrypted packets: 0
      Decrypted bytes: 0
   [DSA statistics]
      Keys generated: 0
      Signatures: 0
      Verifications: 0
   [SSL statistics]
      Outbound records: 0
      Inbound records: 0
   [RNG statistics]
      Random number requests: 1
      Random number request failures: 0
```

The following table explains the output.

Output	Description
Capacity	This section pertains to the crypto acceleration that the threat defense device can support.
Supports hardware crypto	(True/False) The threat defense device can support hardware crypto acceleration.
Supports modular hardware crypto	(True/False) Any supported hardware crypto accelerator can be inserted as a separate plug-in card or module.
Max accelerators	The maximum number of hardware crypto accelerators that the threat defense device supports.

Output	Description
Mac crypto throughput	The maximum rated VPN throughput for the device.
Max crypto connections	The maximum number of supported VPN tunnels for the device.
Global Statistics	This section pertains to the combined hardware crypto accelerators in the device.
Number of active accelerators	The number of active hardware accelerators. An active hardware accelerator has been initialized and is available to process crypto commands.
Number of non-operational accelerators	The number of inactive hardware accelerators. An inactive hardware accelerator has been detected, but either has not completed initialization or has failed and is no longer usable.
Input packets	The number of inbound packets processed by all hardware crypto accelerators.
Input bytes	The number of bytes of data in the processed inbound packets.
Output packets	The number of outbound packets processed by all hardware crypto accelerators.
Output error packets	The number of outbound packets processed by all hardware crypto accelerators in which an error has been detected.
Output bytes	The number of bytes of data in the processed outbound packets.
Accelerator 0	Each of these sections pertains to a crypto accelerator. The first one (Accelerator 0) is always the software crypto engine. Although not a hardware accelerator, the threat defense uses it to perform specific crypto tasks, and its statistics appear here. Accelerators 1 and higher are always hardware crypto accelerators.
Status	The status of the accelerator, which indicates whether the accelerator is being initialized, is active, or has failed.
Software crypto engine	The type of accelerator and firmware version (if applicable).
Slot	The slot number of the accelerator (if applicable).
Active time	The length of time that the accelerator has been in the active state.
Total crypto transforms	The total number of crypto commands that were performed by the accelerator.
Total dropped packets	The total number of packets that were dropped by the accelerator because of errors.
Input statistics	This section pertains to input traffic that was processed by the accelerator. Input traffic is considered to be ciphertext that must be decrypted and/or authenticated.

Output	Description
Input packets	The number of input packets that have been processed by the accelerator.
Input bytes	The number of input bytes that have been processed by the accelerator
Input hashed packets	The number of packets for which the accelerator has performed hash operations.
Input hashed bytes	The number of bytes over which the accelerator has performed hash operations.
Decrypted packets	The number of packets for which the accelerator has performed symmetric decryption operations.
Decrypted bytes	The number of bytes over which the accelerator has performed symmetric decryption operations.
Output statistics	This section pertains to output traffic that has been processed by the accelerator. Input traffic is considered clear text that must be encrypted and/or hashed.
Output packets	The number of output packets that have been processed by the accelerator.
Output bad packets	The number of output packets that have been processed by the accelerator in which an error has been detected.
Output bytes	The number of output bytes that have been processed by the accelerator.
Output hashed packets	The number of packets for which the accelerator has performed outbound hash operations.
Output hashed bytes	The number of bytes over which the accelerator has performed outbound hash operations.
Encyrpted packets	The number of packets for which the accelerator has performed symmetric encryption operations.
Encyrpted bytes	The number of bytes over which the accelerator has performed symmetric encryption operations.
Diffie-Hellman statistics	This section pertains to Diffie-Hellman key exchange operations.
Keys generated	The number of Diffie-Hellman key sets that have been generated by the accelerator.
Secret keys derived	The number of Diffie-Hellman shared secrets that have been derived by the accelerator.
RSA statistics	This section pertains to RSA crypto operations.
Keys generated	The number of RSA key sets that have been generated by the accelerator.

Output	Description
Signatures	The number of RSA signature operations that have been performed by the accelerator.
Verifications	The number of RSA signature verifications that have been performed by the accelerator.
Encrypted packets	The number of packets for which the accelerator has performed RSA encryption operations.
Decrypted packets	The number of packets for which the accelerator has performed RSA decryption operations.
Decrypted bytes	The number of bytes of data over which the accelerator has performed RSA decryption operations.
DSA statistics	This section pertains to DSA operations. Note that DSA is not supported as of Version 8.2, so these statistics are no longer displayed.
Keys generated	The number of DSA key sets that have been generated by the accelerator.
Signatures	The number of DSA signature operations that have been performed by the accelerator.
Verifications	The number of DSA signature verifications that have been performed by the accelerator.
SSL statistics	This section pertains to SSL record processing operations.
Outbound records	The number of SSL records that have been encrypted and authenticated by the accelerator.
Inbound records	The number of SSL records that have been decrypted and authenticated by the accelerator.
RNG statistics	This section pertains to random number generation.
Random number requests	The number of requests to the accelerator for a random number.
Random number request failures	The number of random number requests to the accelerator that did not succeed.

On platforms that support IPsec flow offload, the output shows the statistics for offloaded flows while the global counters show the total of all offloaded and non-offloaded flows for all accelerator engines on the device.

```
> show crypto accelerator statistics

Crypto Accelerator Status
-------------------------
[Capability]
   Supports hardware crypto: True
   Supported TLS Offload Mode: HARDWARE
```

```
   Supports modular hardware crypto: False
   Max accelerators: 3
   Max crypto throughput: 3000 Mbps
   Max crypto connections: 3000
[Global Statistics]
   Number of active accelerators: 2
   Number of non-operational accelerators: 0
   Input packets: 108
   Input bytes: 138912
   Output packets: 118
   Output error packets: 0
   Output bytes: 142329

[Accelerator 0]
   Status: OK
   Software crypto engine
   Slot: 0
   Active time: 489 seconds
   Total crypto transforms: 2770
   Total dropped packets: 0
   [Input statistics]
      Input packets: 0
      Input bytes: 19232
      Input hashed packets: 0
      Input hashed bytes: 0
      Decrypted packets: 0
      Decrypted bytes: 19232
   [Output statistics]
      Output packets: 0
      Output bad packets: 0
      Output bytes: 18784
      Output hashed packets: 0
      Output hashed bytes: 0
      Encrypted packets: 0
      Encrypted bytes: 18784
   [Diffie-Hellman statistics]
      Keys generated: 0
      Secret keys derived: 0
   [RSA statistics]
      Keys generated: 1
      Signatures: 1
      Verifications: 1
      Encrypted packets: 1
      Encrypted bytes: 28
      Decrypted packets: 1
      Decrypted bytes: 256
   [ECDSA statistics]
      Keys generated: 13
      Signatures: 12
      Verifications: 15
   [EDDSA statistics]
      Keys generated: 0
      Signatures: 0
      Verifications: 0
   [SSL statistics]
      Outbound records: 0
      Inbound records: 0
   [RNG statistics]
      Random number requests: 0
      Random number request failures: 0
   [HMAC statistics]
      HMAC requests: 54

[Accelerator 1]
```

```
Status: OK
Encryption hardware device : Cisco ASA Crypto on-board accelerator (revision 0x1)
                         AE microcode        : CNN5x-MC-AE-MAIN-0007
                         SE SSL microcode    : CNN5x-MC-SE-SSL-0018
Slot: 1
Active time: 497 seconds
Total crypto transforms: 2910
Total dropped packets: 0
[Input statistics]
   Input packets: 4
   Input bytes: 13056
   Input hashed packets: 0
   Input hashed bytes: 0
   Decrypted packets: 4
   Decrypted bytes: 6528
[Output statistics]
   Output packets: 14
   Output bad packets: 0
   Output bytes: 20786
   Output hashed packets: 0
   Output hashed bytes: 0
   Encrypted packets: 14
   Encrypted bytes: 10393
[Offloaded Input statistics]
   Input packets: 106
   Input bytes: 115328
   Input hashed packets: 0
   Input hashed bytes: 0
   Decrypted packets: 107
   Decrypted bytes: 112992
[Offloaded Output statistics]
   Output packets: 107
   Output bytes: 116416
   Output hashed packets: 0
   Output hashed bytes: 0
   Encrypted packets: 107
   Encrypted bytes: 112992
Total dropped packets: 0
[Diffie-Hellman statistics]
   Keys generated: 194
   Secret keys derived: 1
[RSA statistics]
   Keys generated: 0
   Signatures: 2
   Verifications: 1
   Encrypted packets: 3
   Encrypted bytes: 162
   Decrypted packets: 2
   Decrypted bytes: 512
[ECDSA statistics]
   Keys generated: 0
   Signatures: 0
   Verifications: 0
[EDDSA statistics]
   Keys generated: 0
   Signatures: 0
   Verifications: 0
[SSL statistics]
   Outbound records: 14
   Inbound records: 4
[RNG statistics]
   Random number requests: 34
   Random number request failures: 0
```

```
[HMAC statistics]
   HMAC requests: 26
```

Related Commands

Command	Description
clear crypto accelerator statistics	Clears the global and accelerator-specific statistics in the crypto accelerator MIB.
clear crypto protocol statistics	Clears the protocol-specific statistics in the crypto accelerator MIB.
show crypto protocol statistics	Displays the protocol-specific statistics from the crypto accelerator MIB.

show crypto accelerator usage

This command allows you to view TLS crypto acceleration core usage and average utilization across all cores. This command is not available on all hardware platforms.

For guidelines and limitations of TLS crypto acceleration, see the *Management Center Configuration Guide*.

show crypto accelerator usage [**detail**]

Syntax Description

detail	(Optional.) Displays more detail, which is useful if your managed device has threat defense container instances.

Command History

Release	Modification
6.6	This command was introduced.

Usage Guidelines

Displays the core usage on each core and the average utlization of each core. Depending on your hardware model, the command might not be available and might display different statistics.

Examples

Following is an example of viewing the core usage of TLS crypto acceleration:

```
> show crypto accelerator usage
Crypto engine 0: 64 ADMIN SE cores, utilization 18.8%
Crypto engine 1: 64 ADMIN SE cores, utilization 17.2%
Total 128 ADMIN SE cores, utilization18%
Crypto engine 0: 64 ADMIN AE cores, utilization 0%
Crypto engine 1: 64 ADMIN AE cores, utilization 0%
Total 128 ADMIN AE cores, utilization0%
```

Following is an example of viewing detailed usage information:

```
 show crypto accelerator usage detail
Crypto engine 0: 64 IPSec/SSL crypto cores, utilization 18.8%
Crypto engine 1: 64 IPSec/SSL crypto cores, utilization 17.2%
Total 128 IPSec/SSL cryto cores, utilization 18%
Crypto engine 0: 64 Asymmetric crypto cores, utilization 0%
Crypto engine 1: 64 Asymmetric crypto cores, utilization 0%
Total 128 Asymmetric crypto cores, utilization 0%
```

show crypto ca certificates

To display the certificates associated with a specific trustpoint or to display all the certificates installed on the system, use the **show crypto ca certificates** command.

show crypto ca certificates [*trustpointname*]

Syntax Description

trustpointname	(Optional) The name of a trustpoint. If you do not specify a name, this command displays all certificates installed on the threat defense device.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show crypto ca certificates** command:

```
>show crypto ca certificates tp1
CA Certificate
    Status: Available
    Certificate Serial Number 2957A3FF296EF854FD0D6732FE25B45
    Certificate Usage: Signature
    Issuer:
        CN = ms-root-sha-06-2004
        OU = rootou
        O = cisco
        L = franklin
        ST - massachusetts
        C = US
        EA = a@b.con
   Subject:
        CN = ms-root-sha-06-2004
        OU = rootou
        O = cisco
        L = franklin
        ST = massachusetts
        C = US
        EA = example.com
   CRL Distribution Point
        ldap://w2kadvancedsrv/CertEnroll/ms-root-sha-06-2004.crl
   Validity Date:
        start date: 14:11:40 UTC Jun 26 2004
        end date: 14:01:30 UTC Jun 4 2022
   Associated Trustpoints: tp2 tp1
```

show crypto ca crls

To display all cached certificate revocation lists (CRLs) or to display all CRLs cached for a specified trustpoint, use the **show crypto ca crl** command.

show crypto ca crls [**trustpool** | **trustpoint** *trustpointname*]

Syntax Description

trustpoint *trustpointname*	(Optional) The name of a trustpoint. If you do not specify a name, this command displays all CRLs cached on the threat defense device.
trustpool	Displays all trustpool-related CRLs.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following is sample output from the **show crypto ca crl** command:

```
> show crypto ca crl trustpoint tp1
CRL Issuer Name:
    cn=ms-sub1-ca-5-2004,ou=Franklin DevTest,o=Cisco
Systems,l=Franklin,st=MA,c=US,ea=user@example.com
    LastUpdate: 19:45:53 UTC Dec 24 2004
    NextUpdate: 08:05:53 UTC Jan 1 2005
    Retrieved from CRL Distribution Point:
      http://win2k-ad2.frk-ms-pki.cisco.com/CertEnroll/ms-sub1-ca-5-2004.crl
    Associated Trustpoints: tp1
```

show crypto ca trustpoints

To display the CA trustpoints, use the **show crypto ca trustpoints** command.

show crypto ca trustpoints [*trustpoint_name*]

Syntax Description

trustpoint_name	(Optional) The name of a trustpoint to display.

Command Default

If you do not specify a trustpoint, all trustpoints are shown.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to display the CA trustpoints.

```
> show crypto ca trustpoints
Trustpoint ftd-self:
    Configured for self-signed certificate generation.
```

show crypto ca trustpool

To display the certificates that constitute the trustpool, use the **show crypto ca trustpool** command.

show crypto ca trustpool [**detail** | **policy**]

Syntax Description

detail	(Optional) Displays certificate details.
policy	(Optional) Displays the configured trustpool policy.

Command Default

This command shows an abbreviated display of all the trustpool certificates. When the **detail** option is specified, more information is included.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The output of the **show crypto ca trustpool** command includes the fingerprint value of each certificate. These values are required for removal operation.

Examples

The following example shows how to display the certificates in the trustpool.

```
> show crypto ca trustpool
CA Certificate
Status: Available
Certificate Serial Number: 6c386c409f4ff4944154635da520ed4c
Certificate Usage: Signature
Public Key Type: RSA (2048 bits)
Signature Algorithm: SHA1 with RSA Encryption
Issuer Name: cn=bxb2008-root
dc=bdb2008
dc=mycompany
dc=com
Subject Name:
cn=bxb2008-root
dc=bxb2008
dc=cisco
dc=com
Validity Date:
start date:17:21:06 EST Jan 14 2009
end date:17:31:06 EST Jan 14 2024
CA Certificate
Status: Available
Certificate Serial Number: 58d1c756000000000059
Certificate Usage: Signature
Public Key Type: RSA (2048 bits)
Signature Algorithm: SHA1 with RSA Encryption
Issuer Name:
cn=bxb2008-root
dc=bxb2008
dc=mycompany
dc=com
```

```
Subject Name:
cn=BXB2008SUB1-CA
dc=bxb2008
dc=cisco
dc=com
OCSP AIA:
URL: http://bxb2008-1.bxb2008.mycompany.com/ocsp
CRL Distribution Points:
(1) http://bxb2008-1.bxb2008.mycompany.com/CertEnroll/bxb2008-root.crl
Validity Date:
start date:11:54:34 EST May 18 2009
end date:12:04:34 EST May 18 2011
```

The following example shows how to display the trustpool policy.

```
> show crypto ca trustpool policy
800 trustpool certificates installed
Trustpool auto import statistics:
 Last import result: SUCCESS
 Next scheduled import at 22:00:00 Tues Jul 21 2015
Trustpool Policy
Trustpool revocation checking is disabled
CRL cache time: 123 seconds
CRL next update field: required and forced
Automatic import of trustpool certificates is enabled
Automatic import URL: http://www.thawte.com
Download time: 22:00:00
Policy overrides:
map: map1
match:issuer-name eq cn=Mycompany Manufacturing CA
match:issuer-name eq cn=Mycompany CA
action:skip revocation-check
map: map2
match: issuer-name eq cn=mycompany Manufacturing CA
match: issuer-name eq cn=mycompany CA2
action: allowed expired certificates
```

Related Commands

Command	Description
clear crypto ca trustpool	Removes all certificates from the trustpool.

show crypto debug-condition

To display the currently configured filters, the unmatched states, and the error states for IPsec and ISAKMP debugging messages, use the **show crypto debug-condition** command.

show crypto debug-condition

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows the filtering conditions:

```
> show crypto debug-condition
Crypto conditional debug is turned ON
IKE debug context unmatched flag:  OFF
IPsec debug context unmatched flag:  ON
IKE peer IP address filters:
1.1.1.0/24   2.2.2.2
IKE user name filters:
my_user
```

Related Commands

Command	Description
debug crypto condition	Sets filtering conditions for IPsec and ISAKMP debugging messages.
debug crypto condition error	Shows debugging messages whether or not filtering conditions have been specified.
debug crypto condition unmatched	Shows debugging messages for IPsec and ISAKMP that do not include sufficient context information for filtering.

show crypto ikev1

To display the information about Internet Key Exchange version 1 (IKEv1), use the **show crypto ikev1** command.

show crypto ikev1 {**ipsec-over-tcp** | **sa** [**detail**] | **stats**}

Syntax Description

ipsec-over-tcp	Displays the IPsec over TCP data.
sa [**detail**]	Displays information about the IKEv1 runtime security association (SA) database. Include the **detail** keyword to display detailed output about the SA database.
stats	Displays the IKEv1 statistics.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays detailed information about the SA database. If you do not include the detail keyword, only the IKE Peer, Type, Dir, Rky, and State columns are shown.

```
> show crypto ikev1 sa detail
IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
1 209.165.200.225 User  Resp  No   AM_Active  3des    SHA   preshrd 86400

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
2 209.165.200.226 User  Resp  No   AM_ACTIVE  3des    SHA   preshrd 86400

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
3 209.165.200.227 User  Resp  No   AM_ACTIVE  3des    SHA   preshrd 86400

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
4 209.165.200.228 User  Resp  No   AM_ACTIVE  3des    SHA   preshrd 86400
```

The following example displays the IPsec over TCP data:

```
> show crypto ikev1 ipsec-over-tcp
Global IKEv1 IPSec over TCP Statistics
-------------------------------
Embryonic connections: 0
Active connections: 0
Previous connections: 0
Inbound packets: 0
Inbound dropped packets: 0
Outbound packets: 0
Outbound dropped packets: 0
RST packets: 0
Recevied ACK heart-beat packets: 0
Bad headers: 0
Bad trailers: 0
Timer failures: 0
```

```
Checksum errors: 0
Internal errors: 0
```

The following example displays the Global IKEv1 statistics:

```
> show crypto ikev1 stats
Global IKEv1 Statistics
  Active Tunnels:              0
  Previous Tunnels:            0
  In Octets:                   0
  In Packets:                  0
  In Drop Packets:             0
  In Notifys:                  0
  In P2 Exchanges:             0
  In P2 Exchange Invalids:     0
  In P2 Exchange Rejects:      0
  In P2 Sa Delete Requests:    0
  Out Octets:                  0
  Out Packets:                 0
  Out Drop Packets:            0
  Out Notifys:                 0
  Out P2 Exchanges:            0
  Out P2 Exchange Invalids:    0
  Out P2 Exchange Rejects:     0
  Out P2 Sa Delete Requests:   0
  Initiator Tunnels:           0
  Initiator Fails:             0
  Responder Fails:             0
  System Capacity Fails:       0
  Auth Fails:                  0
  Decrypt Fails:               0
  Hash Valid Fails:            0
  No Sa Fails:                 0

IKEV1 Call Admission Statistics
  Max In-Negotiation SAs:                50
  In-Negotiation SAs:                     0
  In-Negotiation SAs Highwater:           0
  In-Negotiation SAs Rejected:            0
```

Related Commands

Command	Description
show crypto ikev2 sa	Displays the IKEv2 runtime SA database.
show running-config crypto isakmp	Displays all the active ISAKMP configuration.

show crypto ikev2

To display the information about Internet Key Exchange version 2 (IKEv2), use the **show crypto ikev2** command.

show crypto ikev2 {**sa** [**detail**] | **stats**}

Syntax Description

sa [**detail**]	Displays information about the IKEv2 runtime security association (SA) database. Include the **detail** keyword to display detailed output about the SA database.
stats	Displays the IKEv2 statistics.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays detailed information about the SA database:

```
> show crypto ikev2 sa detail
IKEv2 SAs:
Session-id:1, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id            Local             Remote              Status   Role
671069399            10.0.0.0/500      10.255.255.255/500  READY    INITIATOR
      Encr: AES-GCM, keysize: 256, Hash: N/A, DH Grp:20, Auth sign: PSK, Auth verify: PSK
      Life/Active Time: 86400/188 sec
      Session-id: 1
      Status Description: Negotiation done
      Local spi: 80173A0373C2D403       Remote spi: AE8AEFA1B97DBB22
      Local id: asa
      Remote id: asa1
      Local req mess id: 8              Remote req mess id: 7
      Local next mess id: 8             Remote next mess id: 7
      Local req queued: 8               Remote req queued: 7
      Local window: 1                   Remote window: 1
      DPD configured for 10 seconds, retry 2
      NAT-T is not detected
Child sa: local selector  0.0.0.0/0 - 255.255.255.255/65535
          remote selector 0.0.0.0/0 - 255.255.255.255/65535
          ESP spi in/out: 0x242a3da5/0xe6262034
          AH spi in/out: 0x0/0x0
          CPI in/out: 0x0/0x0
          Encr: AES-GCM, keysize: 128, esp_hmac: N/A
          ah_hmac: None, comp: IPCOMP_NONE, mode tunnel
```

The following example displays the IKEv2 statistics:

```
> show crypto ikev2 stats
Global IKEv2 Statistics
  Active Tunnels:                         0
  Previous Tunnels:                       0
  In Octets:                              0
```

```
  In Packets:                             0
  In Drop Packets:                        0
  In Drop Fragments:                      0
  In Notifys:                             0
  In P2 Exchange:                         0
  In P2 Exchange Invalids:                0
  In P2 Exchange Rejects:                 0
  In IPSEC Delete:                        0
  In IKE Delete:                          0
  Out Octets:                             0
  Out Packets:                            0
  Out Drop Packets:                       0
  Out Drop Fragments:                     0
  Out Notifys:                            0
  Out P2 Exchange:                        0
  Out P2 Exchange Invalids:               0
  Out P2 Exchange Rejects:                0
  Out IPSEC Delete:                       0
  Out IKE Delete:                         0
  SAs Locally Initiated:                  0
  SAs Locally Initiated Failed:           0
  SAs Remotely Initiated:                 0
  SAs Remotely Initiated Failed:          0
  System Capacity Failures:               0
  Authentication Failures:                0
  Decrypt Failures:                       0
  Hash Failures:                          0
  Invalid SPI:                            0
  In Configs:                             0
  Out Configs:                            0
  In Configs Rejects:                     0
  Out Configs Rejects:                    0
  Previous Tunnels:                       0
  Previous Tunnels Wraps:                 0
  In DPD Messages:                        0
  Out DPD Messages:                       0
  Out NAT Keepalives:                     0
  IKE Rekey Locally Initiated:            0
  IKE Rekey Remotely Initiated:           0
  CHILD Rekey Locally Initiated:          0
  CHILD Rekey Remotely Initiated:         0

IKEV2 Call Admission Statistics
  Max Active SAs:                    No Limit
  Max In-Negotiation SAs:                 250
  Cookie Challenge Threshold:           Never
  Active SAs:                             0
  In-Negotiation SAs:                     0
  Incoming Requests:                      0
  Incoming Requests Accepted:             0
  Incoming Requests Rejected:             0
  Outgoing Requests:                      0
  Outgoing Requests Accepted:             0
  Outgoing Requests Rejected:             0
  Rejected Requests:                      0
  Rejected Over Max SA limit:             0
  Rejected Low Resources:                 0
  Rejected Reboot In Progress:            0
  Cookie Challenges:                      0
  Cookie Challenges Passed:               0
  Cookie Challenges Failed:               0
```

Related Commands

Command	Description
show crypto ikev1 sa	Displays the IKEv1 runtime SA database.
show running-config crypto isakmp	Displays all the active ISAKMP configuration.

show crypto ipsec df-bit

To display the IPsec do-not-fragment (DF-bit) policy for IPsec packets for a specified interface, use the **show crypto ipsec df-bit** command. You can also use the command synonym **show ipsec df-bit**.

show crypto ipsec df-bit *interface*

Syntax Description

interface	Specifies an interface name.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The df-bit setting determines how the system handles the do-not-fragment (DF) bit in the encapsulated header. The DF bit within the IP header determines whether or not a device is allowed to fragment a packet. Based on this setting, the system either clears, sets, or copies the DF-bit setting of the clear-text packet to the outer IPsec header when applying encryption.

Examples

The following example displays the IPsec DF-bit policy for interface named inside:

```
> show crypto ipsec df-bit inside
df-bit inside copy
```

Related Commands

Command	Description
show crypto ipsec fragmentation	Displays the fragmentation policy for IPsec packets.

show crypto ipsec fragmentation

To display the fragmentation policy for IPsec packets, use the **show crypto ipsec fragmentation** command. You can also use the command synonym **show ipsec fragmentation**.

show crypto ipsec fragmentation *interface*

Syntax Description

interface	Specifies an interface name.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

When encrypting packets for a VPN, the system compares the packet length with the MTU of the outbound interface. If encrypting the packet will exceed the MTU, the packet must be fragmented. This command shows whether the system will fragment the packet after encrypting it (after-encryption), or before encrypting it (before-encryption). Fragmenting the packet before encryption is also called prefragmentation, and is the default system behavior because it improves overall encryption performance.

Examples

The following example displays the IPsec fragmentation policy for an interface named inside:

```
> show crypto ipsec fragmentation inside
fragmentation inside before-encryption
```

Related Commands

Command	Description
show crypto ipsec df-bit	Displays the DF-bit policy for a specified interface.

show crypto ipsec policy

To display IPsec secure socket API (SS API) security policy configure for OSPFv3, use the **show crypto ipsec policy** command. You can also use the alternate form of this command: **show ipsec policy**.

show crypto ipsec policy

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows the OSPFv3 authentication and encryption policy.

```
> show crypto ipsec policy
Crypto IPsec client security policy data

Policy name:      OSPFv3-1-256
Policy refcount:  1
Policy flags:     0x00000000
SA handles:       sess 268382208 (0xfff3000) / in 55017 (0xd6e9) / out 90369 (0x16101)
Inbound  ESP SPI:        256 (0x100)
Outbound ESP SPI:        256 (0x100)
Inbound  ESP Auth Key:   12345678901234567890123456789012345678901234567890
Outbound ESP Auth Key:   12345678901234567890123456789012345678901234567890
Inbound  ESP Cipher Key: 12345678901234567890123456789012
Outbound ESP Cipher Key: 12345678901234567890123456789012
Transform set:    esp-aes esp-sha-hmac
```

Related Commands

Command	Description
show ipv6 ospf interface	Displays information about OSPFv3 interfaces.
show crypto sockets	Displays secure socket information.

show crypto ipsec sa

To display a list of IPsec SAs, use the **show crypto ipsec sa** command. You can also use the alternate form of this command: **show ipsec sa**.

show crypto ipsec sa [**assigned-address** | **entry** | **identity** | **inactive** | **map** *map-name* | **peer** *peer-addr* | **spi** | **summary** | **user**] [**detail**]

Syntax Description

assigned-address	(Optional) Displays IPsec SAs for an assigned address.
detail	(Optional) Displays detailed error information on what is displayed.
entry	(Optional) Displays IPsec SAs sorted by peer address
identity	(Optional) Displays IPsec SAs for sorted by identity, not including ESPs. This is a condensed form.
inactive	(Optional) Displays inactive IPsec SAs.
map *map-name*	(Optional) Displays IPsec SAs for the specified crypto map.
peer *peer-addr*	(Optional) Displays IPsec SAs for specified peer IP addresses.
spi	(Optional) Displays IPsec SAs for an SPI
summary	(Optional) Displays IPsec SAs summary by type
user	(Optional) Displays IPsec SAs for a user.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays IPsec SAs that include a tunnel identified as OSPFv3.

```
> show crypto ipsec sa
interface: outside2
    Crypto map tag: def, local addr: 10.132.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (172.20.0.21/255.255.255.255/0/0)
      current_peer: 172.20.0.21
      dynamic allocated peer ip: 10.135.1.5

      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 1145, #pkts decrypt: 1145, #pkts verify: 1145
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
      #pre-frag successes: 2, #pre-frag failures: 1, #fragments created: 10
      #PMTUs sent: 5, #PMTUs rcvd: 2, #decapstulated frags needing reassembly: 1
      #send errors: 0, #recv errors: 0
```

```
    local crypto endpt.: 10.132.0.17, remote crypto endpt.: 172.20.0.21

    path mtu 1500, ipsec overhead 60, media mtu 1500
    current outbound spi: DC15BF68

  inbound esp sas:
    spi: 0x1E8246FC (511854332)
       transform: esp-3des esp-md5-hmac
       in use settings ={L2L, Transport, Manual key, (OSPFv3), }
       slot: 0, conn_id: 3, crypto-map: def
       sa timing: remaining key lifetime (sec): 548
       IV size: 8 bytes
       replay detection support: Y
  outbound esp sas:
    spi: 0xDC15BF68 (3692412776)
       transform: esp-3des esp-md5-hmac
       in use settings ={L2L, Transport, Manual key, (OSPFv3), }
       slot: 0, conn_id: 3, crypto-map: def
       sa timing: remaining key lifetime (sec): 548
       IV size: 8 bytes
       replay detection support: Y

  Crypto map tag: def, local addr: 10.132.0.17

    local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
```

Note Fragmentation statistics are pre-fragmentation statistics if the IPsec SA policy states that fragmentation occurs before IPsec processing. Post-fragmentation statistics appear if the SA policy states that fragmentation occurs after IPsec processing.

The following example displays IPsec SAs for a crypto map named def.

```
> show crypto ipsec sa map def
cryptomap: def
    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (10.132.0.21/255.255.255.255/0/0)
      current_peer: 10.132.0.21
      dynamic allocated peer ip: 90.135.1.5

      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 1146, #pkts decrypt: 1146, #pkts verify: 1146
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
      #send errors: 0, #recv errors: 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.132.0.21

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: DC15BF68

    inbound esp sas:
      spi: 0x1E8246FC (511854332)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 3, crypto-map: def
         sa timing: remaining key lifetime (sec): 480
```

```
        IV size: 8 bytes
        replay detection support: Y
    outbound esp sas:
      spi: 0xDC15BF68 (3692412776)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 3, crypto-map: def
         sa timing: remaining key lifetime (sec): 480
         IV size: 8 bytes
         replay detection support: Y

    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (192.168.132.0/255.255.255.0/0/0)
      current_peer: 10.135.1.8
      dynamic allocated peer ip: 0.0.0.0

      #pkts encaps: 73672, #pkts encrypt: 73672, #pkts digest: 73672
      #pkts decaps: 78824, #pkts decrypt: 78824, #pkts verify: 78824
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 73672, #pkts comp failed: 0, #pkts decomp failed: 0
      #send errors: 0, #recv errors: 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.135.1.8

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: 3B6F6A35

    inbound esp sas:
      spi: 0xB32CF0BD (3006066877)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 4, crypto-map: def
         sa timing: remaining key lifetime (sec): 263
         IV size: 8 bytes
         replay detection support: Y
    outbound esp sas:
      spi: 0x3B6F6A35 (997157429)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 4, crypto-map: def
         sa timing: remaining key lifetime (sec): 263
         IV size: 8 bytes
         replay detection support: Y
```

The following example shows IPsec SAs for the keyword **entry**.

```
> show crypto ipsec sa entry
peer address: 10.132.0.21
    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (10.132.0.21/255.255.255.255/0/0)
      current_peer: 10.132.0.21
      dynamic allocated peer ip: 90.135.1.5

      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 1147, #pkts decrypt: 1147, #pkts verify: 1147
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
      #send errors: 0, #recv errors: 0
```

```
      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.132.0.21

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: DC15BF68

    inbound esp sas:
      spi: 0x1E8246FC (511854332)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 3, crypto-map: def
         sa timing: remaining key lifetime (sec): 429
         IV size: 8 bytes
         replay detection support: Y
    outbound esp sas:
      spi: 0xDC15BF68 (3692412776)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 3, crypto-map: def
         sa timing: remaining key lifetime (sec): 429
         IV size: 8 bytes
         replay detection support: Y

peer address: 10.135.1.8
    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (192.168.132.0/255.255.255.0/0/0)
      current_peer: 10.135.1.8
      dynamic allocated peer ip: 0.0.0.0

      #pkts encaps: 73723, #pkts encrypt: 73723, #pkts digest: 73723
      #pkts decaps: 78878, #pkts decrypt: 78878, #pkts verify: 78878
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 73723, #pkts comp failed: 0, #pkts decomp failed: 0
      #send errors: 0, #recv errors: 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.135.1.8

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: 3B6F6A35

    inbound esp sas:
      spi: 0xB32CF0BD (3006066877)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 4, crypto-map: def
         sa timing: remaining key lifetime (sec): 212
         IV size: 8 bytes
         replay detection support: Y
    outbound esp sas:
      spi: 0x3B6F6A35 (997157429)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 4, crypto-map: def
         sa timing: remaining key lifetime (sec): 212
         IV size: 8 bytes
         replay detection support: Y
```

The following example shows IPsec SAs with the keywords **entry detail**.

```
> show crypto ipsec sa entry detail
peer address: 10.132.0.21
    Crypto map tag: def, local addr: 172.20.0.17
```

```
      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (10.132.0.21/255.255.255.255/0/0)
      current_peer: 10.132.0.21
      dynamic allocated peer ip: 90.135.1.5

      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 1148, #pkts decrypt: 1148, #pkts verify: 1148
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
      #pkts no sa (send): 0, #pkts invalid sa (rcv): 0
      #pkts encaps failed (send): 0, #pkts decaps failed (rcv): 0
      #pkts invalid prot (rcv): 0, #pkts verify failed: 0
      #pkts invalid identity (rcv): 0, #pkts invalid len (rcv): 0
      #pkts replay rollover (send): 0, #pkts replay rollover (rcv): 0
      #pkts replay failed (rcv): 0
      #pkts internal err (send): 0, #pkts internal err (rcv): 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.132.0.21

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: DC15BF68

    inbound esp sas:
      spi: 0x1E8246FC (511854332)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 3, crypto-map: def
         sa timing: remaining key lifetime (sec): 322
         IV size: 8 bytes
         replay detection support: Y
    outbound esp sas:
      spi: 0xDC15BF68 (3692412776)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 3, crypto-map: def
         sa timing: remaining key lifetime (sec): 322
         IV size: 8 bytes
         replay detection support: Y

peer address: 10.135.1.8
    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (192.168.132.0/255.255.255.0/0/0)
      current_peer: 10.135.1.8
      dynamic allocated peer ip: 0.0.0.0

      #pkts encaps: 73831, #pkts encrypt: 73831, #pkts digest: 73831
      #pkts decaps: 78989, #pkts decrypt: 78989, #pkts verify: 78989
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 73831, #pkts comp failed: 0, #pkts decomp failed: 0
      #pkts no sa (send): 0, #pkts invalid sa (rcv): 0
      #pkts encaps failed (send): 0, #pkts decaps failed (rcv): 0
      #pkts invalid prot (rcv): 0, #pkts verify failed: 0
      #pkts invalid identity (rcv): 0, #pkts invalid len (rcv): 0
      #pkts replay rollover (send): 0, #pkts replay rollover (rcv): 0
      #pkts replay failed (rcv): 0
      #pkts internal err (send): 0, #pkts internal err (rcv): 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.135.1.8

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: 3B6F6A35
```

```
    inbound esp sas:
      spi: 0xB32CF0BD (3006066877)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 4, crypto-map: def
         sa timing: remaining key lifetime (sec): 104
         IV size: 8 bytes
         replay detection support: Y
    outbound esp sas:
      spi: 0x3B6F6A35 (997157429)
         transform: esp-3des esp-md5-hmac
         in use settings ={RA, Tunnel, }
         slot: 0, conn_id: 4, crypto-map: def
         sa timing: remaining key lifetime (sec): 104
         IV size: 8 bytes
         replay detection support: Y
```

The following example shows IPsec SAs with the keyword **identity**.

```
> show crypto ipsec sa identity
interface: outside2
    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (10.132.0.21/255.255.255.255/0/0)
      current_peer: 10.132.0.21
      dynamic allocated peer ip: 90.135.1.5

      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 1147, #pkts decrypt: 1147, #pkts verify: 1147
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
      #send errors: 0, #recv errors: 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.132.0.21

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: DC15BF68

    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (192.168.132.0/255.255.255.0/0/0)
      current_peer: 10.135.1.8
      dynamic allocated peer ip: 0.0.0.0

      #pkts encaps: 73756, #pkts encrypt: 73756, #pkts digest: 73756
      #pkts decaps: 78911, #pkts decrypt: 78911, #pkts verify: 78911
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 73756, #pkts comp failed: 0, #pkts decomp failed: 0
      #send errors: 0, #recv errors: 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.135.1.8

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: 3B6F6A35
```

The following example shows IPsec SAs with the keywords **identity** and **detail**.

```
> show crypto ipsec sa identity detail
```

```
interface: outside2
    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (10.132.0.21/255.255.255.255/0/0)
      current_peer: 10.132.0.21
      dynamic allocated peer ip: 90.135.1.5

      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 1147, #pkts decrypt: 1147, #pkts verify: 1147
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0
      #pkts no sa (send): 0, #pkts invalid sa (rcv): 0
      #pkts encaps failed (send): 0, #pkts decaps failed (rcv): 0
      #pkts invalid prot (rcv): 0, #pkts verify failed: 0
      #pkts invalid identity (rcv): 0, #pkts invalid len (rcv): 0
      #pkts replay rollover (send): 0, #pkts replay rollover (rcv): 0
      #pkts replay failed (rcv): 0
      #pkts internal err (send): 0, #pkts internal err (rcv): 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.132.0.21

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: DC15BF68

    Crypto map tag: def, local addr: 172.20.0.17

      local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
      remote ident (addr/mask/prot/port): (192.168.132.0/255.255.255.0/0/0)
      current_peer: 10.135.1.8
      dynamic allocated peer ip: 0.0.0.0

      #pkts encaps: 73771, #pkts encrypt: 73771, #pkts digest: 73771
      #pkts decaps: 78926, #pkts decrypt: 78926, #pkts verify: 78926
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 73771, #pkts comp failed: 0, #pkts decomp failed: 0
      #pkts no sa (send): 0, #pkts invalid sa (rcv): 0
      #pkts encaps failed (send): 0, #pkts decaps failed (rcv): 0
      #pkts invalid prot (rcv): 0, #pkts verify failed: 0
      #pkts invalid identity (rcv): 0, #pkts invalid len (rcv): 0
      #pkts replay rollover (send): 0, #pkts replay rollover (rcv): 0
      #pkts replay failed (rcv): 0
      #pkts internal err (send): 0, #pkts internal err (rcv): 0

      local crypto endpt.: 172.20.0.17, remote crypto endpt.: 10.135.1.8

      path mtu 1500, ipsec overhead 60, media mtu 1500
      current outbound spi: 3B6F6A35
```

Related Commands

Command	Description
clear isakmp sa	Clears the IKE runtime SA database.
show running-config isakmp	Displays all the active ISAKMP configuration.

show crypto ipsec stats

To display a list of IPsec statistics, use the **show crypto ipsec stats** command.

show crypto ipsec stats

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays IPsec statistics:

```
> show crypto ipsec stats

IPsec Global Statistics
-----------------------
Active tunnels: 2
Previous tunnels: 9
Inbound
    Bytes: 4933013
    Decompressed bytes: 4933013
    Packets: 80348
    Dropped packets: 0
    Replay failures: 0
    Authentications: 80348
    Authentication failures: 0
    Decryptions: 80348
    Decryption failures: 0
    Decapsulated fragments needing reassembly: 0
Outbound
    Bytes: 4441740
    Uncompressed bytes: 4441740
    Packets: 74029
    Dropped packets: 0
    Authentications: 74029
    Authentication failures: 0
    Encryptions: 74029
    Encryption failures: 0
 Fragmentation successes: 3
  Pre-fragmentation successes:2
  Post-fragmentation successes: 1
 Fragmentation failures: 2
  Pre-fragmentation failures:1
  Post-fragmentation failures: 1
 Fragments created: 10
 PMTUs sent: 1
 PMTUs recvd: 2
Protocol failures: 0
Missing SA failures: 0
System capacity failures: 0
```

Related Commands

Command	Description
clear ipsec sa	Clears IPsec SAs or counters based on specified parameters.
show ipsec sa	Displays IPsec SAs based on specified parameters.
show ipsec sa summary	Displays a summary of IPsec SAs.

show crypto isakmp

To display the ISAKMP information for both IKEv1 and IKEv2, use the **show crypto isakmp** command.

show crypto isakmp {**sa** [**detail**] | **stats**}

Syntax Description

sa [**detail**]	Displays information about the runtime security association (SA) database. Include the **detail** keyword to display detailed output about the SA database.
stats	Displays the IKEv1 and IKEv2 statistics.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **show crypto isakmp** commands combine the output of the equivalent **show crypto ikev1** and **show crypto ikev2** commands.

Following are some tips for reading the SA information.

- Rky can be No or Yes. If yes, a rekey is occurring, and a second matching SA will be in a different state until the rekey completes.
- Role is Initiator or Responder State. This is the current state of the state machine for the SA.
- State—A tunnel that is up and passing data has a value of either MM_ACTIVE or AM_ACTIVE.

Examples

The following example displays detailed information about the SA database.

```
> show crypto isakmp sa detail

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
1 209.165.200.225 User  Resp  No   AM_Active  3des    SHA   preshrd 86400

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
2 209.165.200.226 User  Resp  No   AM_ACTIVE  3des    SHA   preshrd 86400

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
3 209.165.200.227 User  Resp  No   AM_ACTIVE  3des    SHA   preshrd 86400

IKE Peer   Type  Dir   Rky  State      Encrypt Hash  Auth    Lifetime
4 209.165.200.228 User  Resp  No   AM_ACTIVE  3des    SHA   preshrd 86400
```

The following example displays ISAKMP statistics. IKEv1 and IKEv2 are shown separately.

```
> show crypto isakmp stats

Global IKEv1 Statistics
  Active Tunnels:             136
  Previous Tunnels:           0
```

```
  In Octets:                    0
  In Packets:                   0
  In Drop Packets:              0
  In Notifys:                   0
  In P2 Exchanges:              0
  In P2 Exchange Invalids:      0
  In P2 Exchange Rejects:       0
  In P2 Sa Delete Requests:     0
  Out Octets:                1344
  Out Packets:                  8
  Out Drop Packets:             0
  Out Notifys:                  0
  Out P2 Exchanges:             0
  Out P2 Exchange Invalids:     0
  Out P2 Exchange Rejects:      0
  Out P2 Sa Delete Requests:    0
  Initiator Tunnels:            2
  Initiator Fails:              2
  Responder Fails:              0
  System Capacity Fails:        0
  Auth Fails:                   0
  Decrypt Fails:                0
  Hash Valid Fails:             0
  No Sa Fails:                  0

IKEV1 Call Admission Statistics
  Max In-Negotiation SAs:                50
  In-Negotiation SAs:                     0
  In-Negotiation SAs Highwater:           0
  In-Negotiation SAs Rejected:            0
In Drop Packets: 925

Global IKEv2 Statistics
  Active Tunnels:                         132
  Previous Tunnels:                       132
  In Octets:                              195471
  In Packets:                             1854
  In Drop Packets:                        925
  In Drop Fragments:                      0
  In Notifys:                             0
  In P2 Exchange:                         132
  In P2 Exchange Invalids:                0
  In P2 Exchange Rejects:                 0
  In IPSEC Delete:                        0
  In IKE Delete:                          0
  Out Octets:                             119029
  Out Packets:                            796
  Out Drop Packets:                       0
  Out Drop Fragments:                     0
  Out Notifys:                            264
  Out P2 Exchange:                        0
Out P2 Exchange Invalids:               0
  Out P2 Exchange Rejects:                0
  Out IPSEC Delete:                       0
  Out IKE Delete:                         0
  SAs Locally Initiated:                  0
  SAs Locally Initiated Failed:           0
  SAs Remotely Initiated:                 0
  SAs Remotely Initiated Failed:          0
  System Capacity Failures:               0
  Authentication Failures:                0
  Decrypt Failures:                       0
  Hash Failures:                          0
  Invalid SPI:                            0
```

```
  In Configs:                             0
  Out Configs:                            0
  In Configs Rejects:                     0
  Out Configs Rejects:                    0
  Previous Tunnels:                       0
  Previous Tunnels Wraps:                 0
  In DPD Messages:                        0
  Out DPD Messages:                       0
  Out NAT Keepalives:                     0
  IKE Rekey Locally Initiated:            0
  IKE Rekey Remotely Initiated:           0
  CHILD Rekey Locally Initiated:          0
  CHILD Rekey Remotely Initiated:         0

IKEV2 Call Admission Statistics
  Max Active SAs:                  No Limit
  Max In-Negotiation SAs:               300
  Cookie Challenge Threshold:           150
  Active SAs:                             0
  In-Negotiation SAs:                     0
  Incoming Requests:                      0
  Incoming Requests Accepted:             0
  Incoming Requests Rejected:             0
  Outgoing Requests:                      0
  Outgoing Requests Accepted:             0
  Outgoing Requests Rejected:             0
  Rejected Requests:                      0
  Rejected Over Max SA limit:             0
  Rejected Low Resources:                 0
  Rejected Reboot In Progress:            0
  Cookie Challenges:                      0
  Cookie Challenges Passed:               0
  Cookie Challenges Failed:               0
```

Related Commands

Command	Description
clear crypto isakmp sa	Clears the IKE runtime SA database.
show running-config crypto isakmp	Displays all the active ISAKMP configuration.

show crypto key mypubkey

To display the key name, usage, and elliptic curve size for ECDSA or RSA keys, use the **show crypto key mypubkey** command.

show crypto key mypubkey {**ecdsa** | **rsa**}

Syntax Description

ecdsa	Displays ECDSA public keys.
rsa	Displays RSA public keys.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays the RSA public key:

```
> show crypto key mypubkey rsa
Key pair was generated at: 18:19:26 UTC May 26 2016
Key name: <Default-RSA-Key>
 Usage: General Purpose Key
 Modulus Size (bits): 1024
 Key Data:

  30819f30 0d06092a 864886f7 0d010101 05000381 8d003081 89028181 00c0bf77
  d651ead6 fca31c72 12064272 36f699b9 e971e198 1503ba6b f0112b63 97252a26
  38827d83 cd71863e b8962da5 bb905a47 666452a1 9eb1a36e dd8aab00 0e4493f1
  4422bf09 4bcfcb95 a83d38a9 7b9caba6 83c9b5b2 cff251f8 a0422a68 3690c9e5
  0cbbe83b 1a8b2460 1f83b43b a9b06912 7cc9f7f9 f596b81e e2a7bde7 8f020301
  0001
>
```

show crypto protocol statistics

To display the protocol-specific statistics in the crypto accelerator MIB, use the **show crypto protocol statistics** command.

show crypto protocol statistics *protocol*

Syntax Description

protocol	Specifies the name of the protocol for which to display statistics. Protocol choices are as follows: **ikev1**—Internet Key Exchange version 1. **ikev2**— Internet Key Exchange version 2. **ipsec**—IP Security Phase-2 protocols. **ssl**—Secure Sockets Layer. **ssh**—Secure Shell protocol **srtp**—Secure Real-time transport protocol **other**—Reserved for new protocols. **all**—All protocols currently supported.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays crypto accelerator statistics for all protocols:

```
> show crypto protocol statistics all
[IKEv1 statistics]
   Encrypt packet requests: 46
   Encapsulate packet requests: 46
   Decrypt packet requests: 40
   Decapsulate packet requests: 40
   HMAC calculation requests: 91
   SA creation requests: 1
   SA rekey requests: 3
   SA deletion requests: 3
   Next phase key allocation requests: 2
   Random number generation requests: 0
   Failed requests: 0
[IKEv2 statistics]
   Encrypt packet requests: 0
   Encapsulate packet requests: 0
   Decrypt packet requests: 0
   Decapsulate packet requests: 0
   HMAC calculation requests: 0
   SA creation requests: 0
   SA rekey requests: 0
   SA deletion requests: 0
```

```
   Next phase key allocation requests: 0
   Random number generation requests: 0
   Failed requests: 0
[IPsec statistics]
   Encrypt packet requests: 700
   Encapsulate packet requests: 700
   Decrypt packet requests: 700
   Decapsulate packet requests: 700
   HMAC calculation requests: 1400
   SA creation requests: 2
   SA rekey requests: 0
   SA deletion requests: 0
   Next phase key allocation requests: 0
   Random number generation requests: 0
   Failed requests: 0
[SSL statistics]
   Encrypt packet requests: 0
   Encapsulate packet requests: 0
   Decrypt packet requests: 0
   Decapsulate packet requests: 0
   HMAC calculation requests: 0
   SA creation requests: 0
   SA rekey requests: 0
   SA deletion requests: 0
   Next phase key allocation requests: 0
   Random number generation requests: 0
   Failed requests: 0
[SSH statistics are not supported]
[SRTP statistics are not supported]
[Other statistics]
   Encrypt packet requests: 0
   Encapsulate packet requests: 0
   Decrypt packet requests: 0
   Decapsulate packet requests: 0
   HMAC calculation requests: 0
   SA creation requests: 0
   SA rekey requests: 0
   SA deletion requests: 0
   Next phase key allocation requests: 0
   Random number generation requests: 99
   Failed requests: 0
>
```

Related Commands

Command	Description
clear crypto accelerator statistics	Clears the global and accelerator-specific statistics in the crypto accelerator MIB.
clear crypto protocol statistics	Clears the protocol-specific statistics in the crypto accelerator MIB.
show crypto accelerator statistics	Displays the global and accelerator-specific statistics from the crypto accelerator MIB.

show crypto sockets

To display crypto secure socket information, use the **show crypto sockets** command.

show crypto sockets

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example displays crypto secure socket information:

```
> show crypto sockets
Number of Crypto Socket connections 1

       Gi0/1  Peers: (local): 2001:1::1
                     (remote): ::
              Local Ident (addr/plen/port/prot): (2001:1::1/64/0/89)
              Remote Ident (addr/plen/port/prot): (::/0/0/89)
              IPsec Profile: "CSSU-UTF"
              Socket State: Open
              Client: "CSSU_App(UTF)" (Client State: Active)

       Crypto Sockets in Listen state:
```

The following table describes the fields in the **show crypto sockets** command output.

Field	Description
Number of Crypto Socket connections	Number of crypto sockets in the system.
Socket State	This state can be Open, which means that active IPsec security associations (SAs) exist, or it can be Closed, which means that no active IPsec SAs exist.
Client	Application name and its state.
Flags	If this field says “shared,” the socket is shared with more than one tunnel interface.
Crypto Sockets in Listen state	Name of the crypto IPsec profile.

Related Commands

Command	Description
show crypto ipsec policy	Displays the crypto secure socket API installed policy information.

show crypto ssl

To display information about the active SSL sessions on the threat defense device, use the **show crypto ssl** command

show crypto ssl [**cache** | **ciphers** | **errors** [**trace**] | **mib** [**64**] | **objects**]

Syntax Description

cache	(Optional) Displays SSL session cache statistics.
ciphers	(Optional) Displays SSL ciphers available for use.
errors	(Optional) Displays SSL errors.
trace	(Optional) Displays SSL error trace information.
mib	(Optional) Displays SSL MIB statistics.
64	(Optional) Displays SSL MIB 64-bit counter statistics.
objects	(Optional) Displays SSL object statistics.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command shows information about the current SSLv3 or greater sessions, including the enabled cipher order, which ciphers are disabled, SSL trustpoints being used, and whether certificate authentication is enabled.

Examples

The following is sample output from the **show ssl** command:

```
> show crypto ssl

Accept connections using SSLv3 or greater and negotiate to TLSv1 or greater
Start connections using TLSv1 and negotiate to TLSv1 or greater
SSL DH Group: group2 (1024-bit modulus)
SSL ECDH Group: group19 (256-bit EC)

SSL trust-points:
  Self-signed (RSA 2048 bits RSA-SHA256) certificate available
  Self-signed (EC 256 bits ecdsa-with-SHA256) certificate available
Certificate authentication is not enabled
```

To display SSL session cache statistics, use the **show crypto ssl cache** command

```
> show crypto ssl cache

SSL session cache statistics:
  Maximum cache size:          100    Current cache size:           0
  Cache hits:                    0    Cache misses:                 0
  Cache timeouts:                0    Cache full:                   0
```

```
  Accept attempts:               0    Accepts successful:           0
  Accept renegotiates:           0
  Connect attempts:              0    Connects successful:          0
  Connect renegotiates:          0
SSL VPNLB session cache statistics:
  Maximum cache size:           10    Current cache size:           0
  Cache hits:                    0    Cache misses:                 0
  Cache timeouts:                0    Cache full:                   0
  Accept attempts:               0    Accepts successful:           0
  Accept renegotiates:           0
  Connect attempts:              0    Connects successful:          0
  Connect renegotiates:          0
SSLDEV session cache statistics:
  Maximum cache size:           20    Current cache size:           0
  Cache hits:                    0    Cache misses:                 0
  Cache timeouts:                0    Cache full:                   0
  Accept attempts:               0    Accepts successful:           0
  Accept renegotiates:           0
  Connect attempts:              0    Connects successful:          0
  Connect renegotiates:          0
DTLS session cache statistics:
  Maximum cache size:          100    Current cache size:           0
  Cache hits:                    0    Cache misses:                 0
  Cache timeouts:                0    Cache full:                   0
  Accept attempts:               0    Accepts successful:           0
  Accept renegotiates:           0
  Connect attempts:              0    Connects successful:          0
  Connect renegotiates:          0
```

To display SSL cipher lists, use the **show crypto ssl cipher** command

```
> show crypto ssl cipher

Current cipher configuration:
default (medium):
  ECDHE-ECDSA-AES256-GCM-SHA384
  ECDHE-RSA-AES256-GCM-SHA384
  DHE-RSA-AES256-GCM-SHA384
  AES256-GCM-SHA384
  ECDHE-ECDSA-AES256-SHA384
  ECDHE-RSA-AES256-SHA384
  DHE-RSA-AES256-SHA256
  AES256-SHA256
  ECDHE-ECDSA-AES128-GCM-SHA256
  ECDHE-RSA-AES128-GCM-SHA256
  DHE-RSA-AES128-GCM-SHA256
  AES128-GCM-SHA256
  ECDHE-ECDSA-AES128-SHA256
  ECDHE-RSA-AES128-SHA256
  DHE-RSA-AES128-SHA256
  AES128-SHA256
  DHE-RSA-AES256-SHA
  AES256-SHA
  DHE-RSA-AES128-SHA
  AES128-SHA
  DES-CBC3-SHA
tlsv1 (medium):
  DHE-RSA-AES256-SHA
  AES256-SHA
  DHE-RSA-AES128-SHA
  AES128-SHA
  DES-CBC3-SHA
tlsv1.1 (medium):
```

```
    DHE-RSA-AES256-SHA
    AES256-SHA
    DHE-RSA-AES128-SHA
    AES128-SHA
    DES-CBC3-SHA
  tlsv1.2 (medium):
    ECDHE-ECDSA-AES256-GCM-SHA384
    ECDHE-RSA-AES256-GCM-SHA384
    DHE-RSA-AES256-GCM-SHA384
    AES256-GCM-SHA384
    ECDHE-ECDSA-AES256-SHA384
    ECDHE-RSA-AES256-SHA384
    DHE-RSA-AES256-SHA256
    AES256-SHA256
    ECDHE-ECDSA-AES128-GCM-SHA256
    ECDHE-RSA-AES128-GCM-SHA256
    DHE-RSA-AES128-GCM-SHA256
    AES128-GCM-SHA256
    ECDHE-ECDSA-AES128-SHA256
    ECDHE-RSA-AES128-SHA256
    DHE-RSA-AES128-SHA256
    AES128-SHA256
    DHE-RSA-AES256-SHA
    AES256-SHA
    DHE-RSA-AES128-SHA
    AES128-SHA
    DES-CBC3-SHA
  dtlsv1 (medium):
    DHE-RSA-AES256-SHA
    AES256-SHA
    DHE-RSA-AES128-SHA
    AES128-SHA
    DES-CBC3-SHA
```

show ctiqbe

To display information about CTIQBE sessions established across the threat defense device, use the **show ctiqbe** command.

show ctiqbe

Command History

Release	Modification
6.2	This command was introduced.

Examples

The following is sample output from the **show ctiqbe** command under the following conditions. There is only one active CTIQBE session setup across the device. It is established between an internal CTI device (for example, a Cisco IP SoftPhone) at local address 10.0.0.99 and an external Cisco Call Manager at 172.29.1.77, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
> show ctiqbe

Total: 1
        LOCAL           FOREIGN         STATE   HEARTBEAT
-----------------------------------------------------------------
1       10.0.0.99/1117  172.29.1.77/2748        1       120
        ---------------------------------------------
        RTP/RTCP: PAT xlates: mapped to 172.29.1.99(1028 - 1029)
        ---------------------------------------------
        MEDIA: Device ID 27     Call ID 0
               Foreign 172.29.1.99      (1028 - 1029)
               Local   172.29.1.88      (26822 - 26823)
        ---------------------------------------------
```

The CTI device has already registered with the CallManager. The device internal address and RTP listening port is PATed to 172.29.1.99 UDP port 1028. Its RTCP listening port is PATed to UDP 1029.

The line beginning with “RTP/RTCP: PAT xlates:” appears only if an internal CTI device has registered with an external CallManager and the CTI device address and ports are PATed to that external interface. This line does not appear if the CallManager is located on an internal interface, or if the internal CTI device address and ports are NATed to the same external interface that is used by the CallManager.

The output indicates a call has been established between this CTI device and another phone at 172.29.1.88. The RTP and RTCP listening ports of the other phone are UDP 26822 and 26823. The other phone locates on the same interface as the CallManager because the threat defense device does not maintain a CTIQBE session record associated with the second phone and CallManager. The active call leg on the CTI device side can be identified with Device ID 27 and Call ID 0.

Related Commands

Commands	Description
inspect ctiqbe	Enables CTIQBE application inspection.
show service-policy	Shows service policy information and statistics.
show conn	Displays the connection state for different connection types.

show ctl-provider

To display the configuration of CTL providers used in unified communications, use the **show ctl-provider** command.

show ctl-provider [*name*]

Syntax Description

name	(Optional) Shows information for this CTL provider only.

Command History

Release	Modification
6.3	This command was introduced.

Examples

This example shows how to display the configuration of the CTL providers.

```
> show ctl-provider
!
ctl-provider my-ctl
 client interface inside address 192.168.1.55
 client interface inside address 192.168.1.56
 client username admin password gWe.oMSKmeGtelxS encrypted
 export certificate ccm-proxy
!
```

show curpriv

To display the current user privileges for a Diagnostic CLI session, use the **show curpriv** command:

show curpriv

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **show curpriv** command displays the current privilege level. Lower privilege level numbers indicate lower privilege levels.

This information does not apply to the users defined by the **configure user** command. Instead, these are the privileges of a user within the **system support diagnotistic-cli** session. You cannot change these privileges.

Examples

The following example shows how to view the privileges for the logged-in user. These privileges apply to the Diagnostic CLI; they do not apply to the ability to use configure commands. You cannot configure permissions for the enable_1 user. These privileges are the same for both **Basic** and **Config** permissions.

```
> show curpriv
Username : enable_1
Current privilege level : 15
Current Mode/s : P_PRIV P_CONF
```