



CHAPTER

25

show asp drop through show curpriv Commands

show asp drop

To debug dropped packets or connections that take place in the control plane path, use the **show asp drop** command in privileged EXEC mode. This command only shows packet and flow drops for traffic that passes through the control plane path, including most inspected traffic, traffic destined directly to the FWSM, and all IPv6 traffic. Packets and flows that are processed and dropped in the FWSM hardware do not appear in the output.

```
show asp drop [flow drop_reason | frame drop_reason]
```

Syntax Description

flow	(Optional) Shows the dropped flows (connections).
frame	(Optional) Shows the dropped packets.
<i>drop_reason</i>	(Optional) Shows the flows or packets dropped by a particular process.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
3.1(1)	This command was introduced.

Usage Guidelines

The **show asp drop** command might help you troubleshoot a problem with the control plane. This information is used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Related Commands

Command	Description
clear asp drop	Clears drop statistics for the accelerated security path.
show conn	Shows information about connections.

show asp table arp

To debug the accelerated security path ARP tables, use the **show asp table arp** command in privileged EXEC mode.

```
show asp table arp [interface interface_name] [address ip_address [netmask mask]]
```

Syntax Description	Parameter	Description
	address <i>ip_address</i>	(Optional) Identifies an IP address for which you want to view ARP table entries.
	interface <i>interface_name</i>	(Optional) Identifies a specific interface for which you want to view the ARP table.
	netmask <i>mask</i>	(Optional) Sets the subnet mask for the IP address.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History	Release	Modification
	3.1(1)	This command was introduced.

Usage Guidelines The **show arp** command shows the contents of the control plane, while the **show asp table arp** command shows the contents of the accelerated security path, which might help you troubleshoot a problem. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide* for more information about the accelerated security path. These tables are used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Examples The following is sample output from the **show asp table arp** command:

```
hostname# show asp table arp

Context: single_vf, Interface: inside
10.86.194.50      Active  000f.66ce.5d46 hits 0
10.86.194.1      Active  00b0.64ea.91a2 hits 638
10.86.194.172    Active  0001.03cf.9e79 hits 0
10.86.194.204    Active  000f.66ce.5d3c hits 0
10.86.194.188    Active  000f.904b.80d7 hits 0
```

■ show asp table arp

```
Context: single_vf, Interface: identity
:: Active 0000.0000.0000 hits 0
0.0.0.0 Active 0000.0000.0000 hits 50208
```

Related Commands

Command	Description
show arp	Shows the ARP table.
show arp statistics	Shows ARP statistics.

show asp table classify

To debug the accelerated security path classifier tables, use the **show asp table classify** command in privileged EXEC mode. The classifier examines properties of incoming packets, such as protocol, and source and destination address, to match each packet to an appropriate classification rule. Each rule is labeled with a classification domain that determines what types of actions are performed, such as dropping a packet or allowing it through.

```
show asp table classify [crypto | domain domain_name | interface interface_name]
```

Syntax Description

domain <i>domain_name</i>	(Optional) Shows entries for a specific classifier domain. See “ Usage Guidelines ” for a list of domains.
interface <i>interface_name</i>	(Optional) Identifies a specific interface for which you want to view the classifier table.
crypto	(Optional) Shows the encrypt, decrypt, and ipsec tunnel flow domains only.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
3.1(1)	This command was introduced.

Usage Guidelines

The **show asp table classifier** command shows the classifier contents of the accelerated security path, which might help you troubleshoot a problem. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide* for more information about the accelerated security path. These tables are used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Classifier domains include the following:

```
aaa-acct
aaa-auth
aaa-user
accounting
arp
capture
capture
conn-nailed
conn-set
```

```
ctcp
decrypt
encrypt
established
filter-activex
filter-ftp
filter-https
filter-java
filter-url
host
inspect
inspect-ctiqbe
inspect-dns
inspect-dns-ids
inspect-ftp
inspect-ftp-data
inspect-gtp
inspect-h323
inspect-http
inspect-icmp
inspect-icmp-error
inspect-ils
inspect-mgcp
inspect-netbios
inspect-pptp
inspect-rsh
inspect-rtsp
inspect-sip
inspect-skinny
inspect-smtp
inspect-snmp
inspect-sqlnet
inspect-sqlnet-plus
inspect-sunrpc
inspect-tftp
inspect-xdmcp
ipsec-natt
ipsec-tunnel-flow
ipsec-user
limits
lu
mac-permit
mgmt-lockdown
mgmt-tcp-intercept
multicast
nat
nat-exempt
nat-exempt-reverse
nat-reverse
null
permit
permit-ip-option
permit-log
pim
ppp
punt
punt-l2
punt-root
shun
tcp-intercept
```

Examples

The following is sample output from the **show asp table classify** command:

```
hostname# show asp table classify

Interface test:
in id=0x36f3800, priority=10, domain=punt, deny=false
    hits=0, user_data=0x0, flags=0x0
    src ip=0.0.0.0, mask=0.0.0.0, port=0
    dst ip=10.86.194.60, mask=255.255.255.255, port=0
in id=0x33d3508, priority=99, domain=inspect, deny=false
    hits=0, user_data=0x0, use_real_addr, flags=0x0
    src ip=0.0.0.0, mask=0.0.0.0, port=0
    dst ip=0.0.0.0, mask=0.0.0.0, port=0
in id=0x33d3978, priority=99, domain=inspect, deny=false
    hits=0, user_data=0x0, use_real_addr, flags=0x0
    src ip=0.0.0.0, mask=0.0.0.0, port=53
    dst ip=0.0.0.0, mask=0.0.0.0, port=0
...
```

Related Commands

Command	Description
show asp drop	Shows the accelerated security path counters for dropped packets.

show asp table interfaces

To debug the accelerated security path interface tables, use the **show asp table interfaces** command in privileged EXEC mode.

show asp table interfaces

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	•

Command History	Release	Modification
	3.1(1)	This command was introduced.

Usage Guidelines The **show asp table interfaces** command shows the interface table contents of the accelerated security path, which might help you troubleshoot a problem. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide* for more information about the accelerated security path. These tables are used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Examples The following is sample output from the **show asp table interfaces** command:

```
hostname# show asp table interfaces

** Flags: 0x0001-DHCP, 0x0002-VMAC, 0x0010-Ident Ifc, 0x0020-HDB Initd,
0x0040-RPF Enabled
Soft-np interface 'dmz' is up
  context single_vE, nicnum 0, mtu 1500
    vlan 300, Not shared, seclvl 50
    0 packets input, 1 packets output
    flags 0x20

Soft-np interface 'foo' is down
  context single_vE, nicnum 2, mtu 1500
    vlan 301, Not shared, seclvl 0
    0 packets input, 0 packets output
    flags 0x20
```



```
Soft-np interface 'outside' is down
  context single_vf, nicnum 1, mtu 1500
  vlan 302, Not shared, seclvl 50
  0 packets input, 0 packets output
  flags 0x20

Soft-np interface 'inside' is up
  context single_vf, nicnum 0, mtu 1500
  vlan 303, Not shared, seclvl 100
  680277 packets input, 92501 packets output
  flags 0x20
...
```

Related Commands

Command	Description
interface	Configures an interface and enters interface configuration mode.
show interface	Displays the runtime status and statistics of interfaces.

show asp table mac-address-table

To debug the accelerated security path MAC address tables, use the **show asp table mac-address-table** command in privileged EXEC mode.

```
show asp table mac-address-table [interface interface_name]
```

Syntax Description

interface (Optional) Shows MAC address tables for a specific interface.
interface_name

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	—	•	•	•	•

Command History

Release	Modification
3.1(1)	This command was introduced.

Usage Guidelines

The **show asp table mac-address-table** command shows the MAC address table contents of the accelerated security path, which might help you troubleshoot a problem. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide* for more information about the accelerated security path. These tables are used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Examples

The following is sample output from the **show asp table mac-address-table** command:

```
hostname# show asp table mac-address-table

interface          mac address          flags
-----
insidel            0009.b74d.3800      None
insidel            0007.e903.ad6e      None
insidel            0007.e950.2067      None
insidel            0050.0499.3749      None
insidel            0012.d96f.e200      None
insidel            0001.02a7.f4ec      None
insidel            0001.032c.6477      None
insidel            0004.5a2d.a1c8      None
insidel            0003.4773.c87b      None
```

```
inside1          000d.88ef.5d1c  None
inside1          00c0.b766.adce  None
inside1          0050.5640.450d  None
inside1          0001.03cf.0431  None
...
```

Related Commands

Command	Description
show mac-address-table	Shows the MAC address table, including dynamic and static entries.

show asp table routing

To debug the accelerated security path routing tables, use the **show asp table routing** command in privileged EXEC mode. This command supports IPv4 and IPv6 addresses.

```
show asp table routing [input | output] [address ip_address [netmask mask] |
interface interface_name]
```

Syntax Description

address <i>ip_address</i>	Sets the IP address for which you want to view routing entries. For IPv6 addresses, you can include the subnet mask as a slash (/) followed by the prefix (0 to 128). For example, enter the following: <i>fe80::2e0:b6ff:fe01:3b7a/128</i>
input	Shows the entries from the input route table.
interface <i>interface_name</i>	(Optional) Identifies a specific interface for which you want to view the routing table.
netmask <i>mask</i>	For IPv4 addresses, specifies the subnet mask.
output	Shows the entries from the output route table.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
3.1(1)	This command was introduced.

Usage Guidelines

The **show asp table routing** command shows the routing table contents of the accelerated security path, which might help you troubleshoot a problem. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide* for more information about the accelerated security path. These tables are used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Examples

The following is sample output from the **show asp table routing** command:

```
hostname# show asp table routing

in 255.255.255.255 255.255.255.255 identity
```

```

in 224.0.0.9      255.255.255.255 identity
in 10.86.194.60   255.255.255.255 identity
in 10.86.195.255  255.255.255.255 identity
in 10.86.194.0    255.255.255.255 identity
in 209.165.202.159 255.255.255.255 identity
in 209.165.202.255 255.255.255.255 identity
in 209.165.201.30  255.255.255.255 identity
in 209.165.201.0   255.255.255.255 identity
in 10.86.194.0     255.255.254.0    inside
in 224.0.0.0       240.0.0.0        identity
in 0.0.0.0         0.0.0.0          inside
out 255.255.255.255 255.255.255.255 foo
out 224.0.0.0       240.0.0.0        foo
out 255.255.255.255 255.255.255.255 test
out 224.0.0.0       240.0.0.0        test
out 255.255.255.255 255.255.255.255 inside
out 10.86.194.0     255.255.254.0    inside
out 224.0.0.0       240.0.0.0        inside
out 0.0.0.0         0.0.0.0          via 10.86.194.1, inside
out 0.0.0.0         0.0.0.0          via 0.0.0.0, identity
out ::              ::               via 0.0.0.0, identity

```

Related Commands

Command	Description
show route	Shows the routing table in the control plane.

show asp table vpn-context

To debug the accelerated security path VPN context tables, use the **show asp table vpn-context** command in privileged EXEC mode.

show asp table vpn-context [detail]

Syntax Description	detail (Optional) Shows additional detail for the VPN context tables.
---------------------------	--

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	•

Command History	Release	Modification
	3.1(1)	This command was introduced.

Usage Guidelines The **show asp table vpn-context** command shows the VPN context contents of the accelerated security path, which might help you troubleshoot a problem. See the *Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Configuration Guide* for more information about the accelerated security path. These tables are used for debugging purposes only, and the information output is subject to change. Consult Cisco TAC to help you debug your system with this command.

Examples The following is sample output from the **show asp table vpn-context** command:

```
hostname# show asp table vpn-context

VPN ID=0058070576, DECR+ESP, UP, pk=0000000000, rk=0000000000, gc=0
VPN ID=0058193920, ENCR+ESP, UP, pk=0000000000, rk=0000000000, gc=0
VPN ID=0058168568, DECR+ESP, UP, pk=0000299627, rk=0000000061, gc=2
VPN ID=0058161168, ENCR+ESP, UP, pk=0000305043, rk=0000000061, gc=1
VPN ID=0058153728, DECR+ESP, UP, pk=0000271432, rk=0000000061, gc=2
VPN ID=0058150440, ENCR+ESP, UP, pk=0000285328, rk=0000000061, gc=1
VPN ID=0058102088, DECR+ESP, UP, pk=0000268550, rk=0000000061, gc=2
VPN ID=0058134088, ENCR+ESP, UP, pk=0000274673, rk=0000000061, gc=1
VPN ID=0058103216, DECR+ESP, UP, pk=0000252854, rk=0000000061, gc=2
...
```

The following is sample output from the **show asp table vpn-context detail** command:

```
hostname# show asp table vpn-context detail
```

```

VPN Ctx = 0058070576 [0x03761630]
State = UP
Flags = DECR+ESP
SA = 0x037928F0
SPI = 0xEA0F21F0
Group = 0
Pkts = 0
Bad Pkts = 0
Bad SPI = 0
Spoof = 0
Bad Crypto = 0
Rekey Pkt = 0
Rekey Call = 0

VPN Ctx = 0058193920 [0x0377F800]
State = UP
Flags = ENCR+ESP
SA = 0x037B4B70
SPI = 0x900FDC32
Group = 0
Pkts = 0
Bad Pkts = 0
Bad SPI = 0
Spoof = 0
Bad Crypto = 0
Rekey Pkt = 0
Rekey Call = 0
...

```

Related Commands

Command	Description
show asp drop	Shows the accelerated security path counters for dropped packets.

show asr

To display the members of ASR groups, use the **show asr** command in privileged EXEC mode.

```
show asr {group_id | all}
```

Syntax Description

<i>group_id</i>	Displays the VLANs that are members of the specified ASR group. Valid values are 1 through 32.
all	Displays the membership for all 32 ASR groups.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	—	•	—	—

Command History

Release	Modification
3.1(1)	This command was introduced.

Usage Guidelines

An ASR group can contain up to 8 members. A “0” (zero) in the output indicates an empty slot. The **show asr** command provides the same output as the **show np asr** command.

Examples

The following is sample output from the **show asr** command. It limits the display to VLANs that are members of ASR group 1.

```
hostname# sh asr 1
```

```
ASR Group | Vlan Entries in ASR Group (0 denotes empty slot)
-----|-----
      1 | 10 20 0 0 0 0 0 0

```

The following is sample output from the **show asr** command. It displays VLAN membership for all possible ASR groups. In this example, only ASR group 1 has member VLANs.

```
hostname# sh asr all
```

```
ASR Group | Vlan Entries in ASR Group (0 denotes empty slot)
-----|-----
      1 | 10 20 0 0 0 0 0 0
      2 | 0 0 0 0 0 0 0 0
      3 | 0 0 0 0 0 0 0 0
      4 | 0 0 0 0 0 0 0 0

```



```

5 | 0 0 0 0 0 0 0 0 0
6 | 0 0 0 0 0 0 0 0 0
7 | 0 0 0 0 0 0 0 0 0
8 | 0 0 0 0 0 0 0 0 0
9 | 0 0 0 0 0 0 0 0 0
10 | 0 0 0 0 0 0 0 0 0
11 | 0 0 0 0 0 0 0 0 0
12 | 0 0 0 0 0 0 0 0 0
13 | 0 0 0 0 0 0 0 0 0
14 | 0 0 0 0 0 0 0 0 0
15 | 0 0 0 0 0 0 0 0 0
16 | 0 0 0 0 0 0 0 0 0
17 | 0 0 0 0 0 0 0 0 0
18 | 0 0 0 0 0 0 0 0 0
19 | 0 0 0 0 0 0 0 0 0
20 | 0 0 0 0 0 0 0 0 0
21 | 0 0 0 0 0 0 0 0 0
22 | 0 0 0 0 0 0 0 0 0
23 | 0 0 0 0 0 0 0 0 0
24 | 0 0 0 0 0 0 0 0 0
25 | 0 0 0 0 0 0 0 0 0
26 | 0 0 0 0 0 0 0 0 0
27 | 0 0 0 0 0 0 0 0 0
28 | 0 0 0 0 0 0 0 0 0
29 | 0 0 0 0 0 0 0 0 0
30 | 0 0 0 0 0 0 0 0 0
31 | 0 0 0 0 0 0 0 0 0
32 | 0 0 0 0 0 0 0 0 0

```

Related Commands

Command	Description
asr-group	Specifies an interface as a member of an ASR group.

show auto-update

To view the Auto Update Server configuration, use the **show auto-update** command in privileged EXEC mode.

show auto-update

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	—	—

Command History	Release	Modification
	3.1(1)	This command was introduced.

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

```
hostname# show arp-inspection
Poll period: 1 minutes, retry count: 1, retry period: 5 minutes
Timeout: none
Device ID: host name [farscape]
```

Related Commands	Command	Description
	auto-update device-id	Sets the FWSM device ID for use with an Auto Update Server.
	auto-update poll-period	Sets how often the FWSM checks for updates from an Auto Update Server.
	auto-update server	Identifies the Auto Update Server.
	auto-update timeout	Stops traffic from passing through the FWSM if the Auto Update Server is not contacted within the timeout period.
	clear configure auto-update	Clears the Auto Update Server configuration

show blocks

To show the packet buffer utilization, use the **show blocks** command in privileged EXEC mode.

```
show blocks [{address hex | all | assigned | free | old | pool size [summary]}] [diagnostics |
dump | header | packet] | queue history [detail]
```

Syntax Description

address <i>hex</i>	(Optional) Shows a block corresponding to this address, in hexadecimal.
all	(Optional) Shows all blocks.
assigned	(Optional) Shows blocks that are assigned and in use by an application.
detail	(Optional) Shows a portion (128 bytes) of the first block for each unique queue type.
dump	(Optional) Shows the entire block contents, including the header and packet information. The difference between dump and packet is that dump includes additional information between the header and the packet.
diagnostics	(Optional) Shows block diagnostics.
free	(Optional) Shows blocks that are available for use.
header	(Optional) Shows the header of the block.
old	(Optional) Shows blocks that were assigned more than a minute ago.
packet	(Optional) Shows the header of the block as well as the packet contents.
pool <i>size</i>	(Optional) Shows blocks of a specific size.
queue history	(Optional) Shows where blocks are assigned when the FWSM runs out of blocks. Sometimes, a block is allocated from the pool but never assigned to a queue. In that case, the location is the code address that allocated the block.
summary	(Optional) Shows detailed information about block usage sorted by the program addresses of applications that allocated blocks in this class, program addresses of applications that released blocks in this class, and the queues to which valid blocks in this class belong.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
3.1(1)	The pool summary option was added.

Usage Guidelines

The **show blocks** command helps you determine if the FWSM is overloaded. This command lists preallocated system buffer utilization. A full memory condition is not a problem as long as traffic is moving through the FWSM. You can use the **show conn** command to see if traffic is moving. If traffic is not moving and the memory is full, there may be a problem.

You can also view this information using SNMP.

The information shown in a security context includes the system-wide information as well as context-specific information about the blocks in use and the high water mark for block usage.

See the “[Examples](#)” section for a description of the display output.

Examples

The following is sample output from the **show blocks** command in single mode:

```
hostname# show blocks
SIZE      MAX      LOW      CNT
    4      1600    1598    1599
   80      400     398     399
  256     3600    3540    3542
 1550     4716    3177    3184
16384      10       10       10
 2048     1000    1000    1000
```

[Table 3](#) shows each field description.

Table 25-1 show blocks Fields

Field	Description
SIZE	Size, in bytes, of the block pool. Each size represents a particular type. Examples are shown below.
4	Duplicates existing blocks in applications such as DNS, ISAKMP, URL filtering, uauth, TFTP, and TCP modules.
80	Used in TCP intercept to generate acknowledgment packets and for failover hello messages.
256	Used for Stateful Failover updates, syslogging, and other TCP functions. These blocks are mainly used for Stateful Failover messages. The active FWSM generates and sends packets to the standby FWSM to update the translation and connection table. In bursty traffic, where high rates of connections are created or torn down, the number of available blocks might drop to 0. This situation indicates that one or more connections were not updated to the standby FWSM. The Stateful Failover protocol catches the missing translation or connection the next time. If the CNT column for 256-byte blocks stays at or near 0 for extended periods of time, then the FWSM is having trouble keeping the translation and connection tables synchronized because of the number of connections per second that the FWSM is processing. Syslog messages sent out from the FWSM also use the 256-byte blocks, but they are generally not released in such quantity to cause a depletion of the 256-byte block pool. If the CNT column shows that the number of 256-byte blocks is near 0, ensure that you are not logging at Debugging (level 7) to the syslog server. This is indicated by the logging trap line in the FWSM configuration. We recommend that you set logging at Notification (level 5) or lower, unless you require additional information for debugging purposes.

Table 25-1 show blocks Fields (continued)

Field	Description
1550	Used to store Ethernet packets for processing through the FWSM. When a packet enters a FWSM interface, it is placed on the input interface queue, passed up to the operating system, and placed in a block. The FWSM determines whether the packet should be permitted or denied based on the security policy and processes the packet through to the output queue on the outbound interface. If the FWSM is having trouble keeping up with the traffic load, the number of available blocks will hover close to 0 (as shown in the CNT column of the command output). When the CNT column is zero, the FWSM attempts to allocate more blocks, up to a maximum of 8192. If no more blocks are available, the FWSM drops the packet.
16384	Only used for the 64-bit, 66-MHz Gigabit Ethernet cards (i82543). See the description for 1550 for more information about Ethernet packets.
2048	Control or guided frames used for control updates.
MAX	Maximum number of blocks available for the specified byte block pool. The maximum number of blocks are carved out of memory at bootup. Typically, the maximum number of blocks does not change. The exception is for the 256- and 1550-byte blocks, where the FWSM can dynamically create more when needed, up to a maximum of 8192.
LOW	Low-water mark. This number indicates the lowest number of this size blocks available since the FWSM was powered up, or since the last clearing of the blocks (with the clear blocks command). A zero in the LOW column indicates a previous event where memory was full.
CNT	Current number of blocks available for that specific size block pool. A zero in the CNT column means memory is full now.

The following is sample output from the **show blocks all** command:

```
hostname# show blocks all
Class 0, size 4
  Block   allocd_by   freed_by   data size   alloccnt   dup_cnt   oper location
0x01799940 0x00000000 0x00101603     0         0         0 alloc not_specified
0x01798e80 0x00000000 0x00101603     0         0         0 alloc not_specified
0x017983c0 0x00000000 0x00101603     0         0         0 alloc not_specified
...
Found 1000 of 1000 blocks
Displaying 1000 of 1000 blocks
```

Table 4 shows each field description.

Table 25-2 show blocks all Fields

Field	Description
Block	The block address.
allocd_by	The program address of the application that last used the block (0 if not used).
freed_by	The program address of the application that last released the block.
data size	The size of the application buffer/packet data that is inside the block.
alloccnt	The number of times this block has been used since the block came into existence.

Table 25-2 show blocks all Fields

Field	Description
dup_cnt	The current number of references to this block if used: 0 means 1 reference, 1 means 2 references.
oper	One of the four operations that was last performed on the block: alloc, get, put, or free.
location	The application that uses the block, or the program address of the application that last allocated the block (same as the allocd_by field).

The following is sample output from the **show blocks** command in a context:

```
hostname/contexta# show blocks
  SIZE  MAX  LOW  CNT  INUSE  HIGH
    4   1600 1599 1599    0     0
   80   400  400  400    0     0
  256  3600 3538 3540    0     1
 1550  4616 3077 3085    0     0
```

The following is sample output from the **show blocks queue history** command:

```
hostname# show blocks queue history
Each Summary for User and Queue_type is followed its top 5 individual queues
Block Size: 4
Summary for User "http", Queue "tcp_unp_c_in", Blocks 1595, Queues 1396
Blk_cnt Q_cnt Last_Op Queue_Type      User      Context
    186    1 put      contexta
    15    1 put      contexta
     1    1 put      contexta
     1    1 put      contextb
     1    1 put      contextc

Summary for User "aaa", Queue "tcp_unp_c_in", Blocks 220, Queues 200
Blk_cnt Q_cnt Last_Op Queue_Type      User      Context
    21    1 put      contexta
     1    1 put      contexta
     1    1 put      contexta
     1    1 put      contextb
     1    1 put      contextc

Blk_cnt Q_cnt Last_Op Queue_Type      User      Context
    200    1 alloc   ip_rx      tcp       contexta
    108    1 get     ip_rx      udp       contexta
     85    1 free    fixup     h323_ras contextb
     42    1 put     fixup     skinny    contextb

Block Size: 1550
Summary for User "http", Queue "tcp_unp_c_in", Blocks 1595, Queues 1000
Blk_cnt Q_cnt Last_Op Queue_Type      User      Context
    186    1 put      contexta
    15    1 put      contexta
     1    1 put      contexta
     1    1 put      contextb
     1    1 put      contextc
...
```

The following is sample output from the **show blocks queue history detail** command:

```
hostname# show blocks queue history detail
History buffer memory usage: 2136 bytes (default)
Each Summary for User and Queue type is followed its top 5 individual queues
Block Size: 4
Summary for User "http", Queue_Type "tcp_unp_c_in", Blocks 1595, Queues 1396
Blk_cnt Q_cnt Last_Op Queue_Type      User      Context
```

```

186      1 put                                contexta
15       1 put                                contexta
1        1 put                                contexta
1        1 put                                contextb
1        1 put                                contextc

First Block information for Block at 0x....
dup_count 0, flags 0x8000000, alloc_pc 0x43ea2a,
start_addr 0xefb1074, read_addr 0xefb118c, write_addr 0xefb1193
urgent_addr 0xefb118c, end_addr 0xefb17b2
0efb1150: 00 00 00 03 47 c5 61 c5 00 05 9a 38 76 80 a3 00 | ....G.a....8v...
0efb1160: 00 0a 08 00 45 00 05 dc 9b c9 00 00 ff 06 f8 f3 | ....E.....
0efb1170: 0a 07 0d 01 0a 07 00 50 00 17 cb 3d c7 e5 60 62 | .....P...=`b
0efb1180: 7e 73 55 82 50 18 10 00 45 ca 00 00 2d 2d 20 49 | ~sU.P...E.-- I
0efb1190: 50 20 2d 2d 0d 0a 31 30 2e 37 2e 31 33 2e 31 09 | P --.10.7.13.1.
0efb11a0: 3d 3d 3e 09 31 30 2e 37 2e 30 2e 38 30 0d 0a 0d | ==>.10.7.0.80...

Summary for User "aaa", Queue "tcp_unp_c_in", Blocks 220, Queues 200
Blk_cnt Q_cnt Last_Op Queue_Type      User      Context
21      1 put                                contexta
1       1 put                                contexta
1       1 put                                contexta
1       1 put                                contextb
1       1 put                                contextc

First Block information for Block at 0x....
dup_count 0, flags 0x8000000, alloc_pc 0x43ea2a,
start_addr 0xefb1074, read_addr 0xefb118c, write_addr 0xefb1193
urgent_addr 0xefb118c, end_addr 0xefb17b2
0efb1150: 00 00 00 03 47 c5 61 c5 00 05 9a 38 76 80 a3 00 | ....G.a....8v...
0efb1160: 00 0a 08 00 45 00 05 dc 9b c9 00 00 ff 06 f8 f3 | ....E.....
0efb1170: 0a 07 0d 01 0a 07 00 50 00 17 cb 3d c7 e5 60 62 | .....P...=`b
0efb1180: 7e 73 55 82 50 18 10 00 45 ca 00 00 2d 2d 20 49 | ~sU.P...E.-- I
0efb1190: 50 20 2d 2d 0d 0a 31 30 2e 37 2e 31 33 2e 31 09 | P --.10.7.13.1.
0efb11a0: 3d 3d 3e 09 31 30 2e 37 2e 30 2e 38 30 0d 0a 0d | ==>.10.7.0.80...

...

```

total_count: total buffers in this class

The following is sample output from the **show blocks pool summary** command:

```

hostname# show blocks pool 1550 summary
Class 3, size 1550

=====
total_count=1531    miss_count=0
Alloc_pc          valid_cnt          invalid_cnt
0x3b0a18          00000256          00000000
0x01ad0760 0x01acfe00 0x01acf4a0 0x01aceb40 00000000 0x00000000
0x3a8f6b          00001275          00000012
0x05006aa0 0x05006140 0x050057e0 0x05004520 00000000
0x00000000

=====
total_count=9716    miss_count=0
Freed_pc          valid_cnt          invalid_cnt
0x9a81f3          00000104          00000007
0x05006140 0x05000380 0x04fffa20 0x04ffde00 00000000 0x00000000
0x9a0326          00000053          00000033
0x05006aa0 0x050057e0 0x05004e80 0x05003260 00000000 0x00000000
0x4605a2          00000005          00000000
0x04ff5ac0 0x01e8e2e0 0x01e2eac0 0x01e17d20 00000000 0x00000000
...

=====
total_count=1531    miss_count=0
Queue  valid_cnt          invalid_cnt

```

show blocks

```

0x3b0a18      00000256      00000000  Invalid Bad qtype
                0x01ad0760 0x01acfe00 0x01acf4a0 0x01aceb40 00000000 0x00000000
0x3a8f6b      00001275      00000000  Invalid Bad qtype
                0x05006aa0 0x05006140 0x050057e0 0x05004520 00000000
0x00000000

=====
free_cnt=8185  fails=0  actual_free=8185  hash_miss=0
                03a8d3e0 03a8b7c0 03a7fc40 03a6ff20 03a6f5c0 03a6ec60 kao-f1#

```

Table 5 shows each field description.

Table 25-3 show blocks pool summary Fields

Field	Description
total_count	The number of blocks for a given class.
miss_count	The number of blocks not reported in the specified category due to technical reasons.
Freed_pc	The program addresses of applications that released blocks in this class.
Alloc_pc	The program addresses of applications that allocated blocks in this class.
Queue	The queues to which valid blocks in this class belong.
valid_cnt	The number of blocks that are currently allocated.
invalid_cnt	The number of blocks that are not currently allocated.
Invalid Bad qtype	Either this queue has been freed and the contents are invalid or this queue was never initialized.
Valid tcp_usr_conn_inp	The queue is valid.

Related Commands

Command	Description
blocks	Increases the memory assigned to block diagnostics
clear blocks	Clears the system buffer statistics.
show conn	Shows active connections.

show boot device (IOS)

To view the default boot partition, use the **show boot device** command.

```
show boot device [mod_num]
```

Syntax Description	<i>mod_num</i>	(Optional) Specifies the module number. Use the show module command to view installed modules and their numbers.
---------------------------	----------------	---

Defaults	The default boot partition is cf:4.
-----------------	-------------------------------------

Command Modes	Privileged EXEC.
----------------------	------------------

Command History	Release	Modification
	Preexisting	This command was preexisting.

Examples The following is sample output from the **show boot device** command that shows the boot partitions for each installed FWSM on Cisco IOS software:

```
Router# show boot device
[mod:1 ]:
[mod:2 ]:
[mod:3 ]:
[mod:4 ]: cf:4
[mod:5 ]: cf:4
[mod:6 ]:
[mod:7 ]: cf:4
[mod:8 ]:
[mod:9 ]:
```

Related Commands	Command	Description
	boot device (IOS)	Sets the default boot partition.
	show module (IOS)	Shows all installed modules.

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]
```

Syntax Description

<i>capture_name</i>	(Optional) Name of the packet capture.
access-list <i>access_list_name</i>	(Optional) Displays information for packets that are based on IP or higher fields for the specific access list identification.
count <i>number</i>	(Optional) Displays the number of packets specified data.
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 transport.
packet-number <i>number</i>	Starts the display at the specified packet number.

Defaults

This command has no default settings.

Command Modes

Security Context Mode: single context mode and multiple context mode

Access Location: system and context command line

Command Mode: privileged mode

Firewall Mode: routed firewall mode and transparent firewall mode

Command History

Release	Modification
3.1(1)	Support for this command was introduced.

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 [Table 25-4](#), the bracketed output is displayed when you specify the **detail** keyword.

Table 25-4 Packet Capture Output Formats

Packet Type	Capture Output Format
802.1Q	<i>HH:MM:SS.ms</i> [ether-hdr] <i>VLAN-info encap-ether-packet</i>
ARP	<i>HH:MM:SS.ms</i> [ether-hdr] <i>arp-type arp-info</i>

Table 25-4 Packet Capture Output Formats (continued)

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

Examples

This example shows how to display the capture configuration:

```
hostname(config)# 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:

```
hostname(config)# 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
```

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 checkheaps

To show the checkheaps statistics, use the **show checkheaps** command in privileged EXEC mode. 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

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	—	•

Command History

Release	Modification
3.1(1)	Support for this command was introduced.

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

```
hostname# 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
```

Related Commands

Command	Description
checkheaps	Sets the checkheap verification intervals.

show checksum

To display the configuration checksum, use the **show checksum** command in privileged EXEC mode.

show checksum

Syntax Description This command has no arguments or keywords.

Defaults This command has no default settings.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	

Command History	Release	Modification
	3.1(1)	Support for 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 config** or **show checksum** command output, the output indicates a normal configuration load or write mode indicator (when loading from or writing to the FWSM Flash partition). The “.” shows that the FWSM 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:

```
hostname(config)# show checksum
Cryptochecksum: 1a2833c0 129ac70b 1a88df85 650d8b81
```

show chunkstat

To display the chunk statistics, use the **show chunkstat** command in privileged EXEC mode.

show chunkstat

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	—	•

Command History	Release	Modification
	1.1(1)	This command was introduced.

Examples The following example shows how to display the chunk statistics:

```
hostname# 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 class

To show the contexts assigned to a class, use the **show class** command in privileged EXEC mode.

show class *name*

Syntax Description

name Specifies the name as a string up to 20 characters long. To show the default class, enter **default** for the name.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	N/A	N/A	—	—	•

Command History

Release	Modification
2.2(1)	This command was introduced.

Examples

The following is sample output from the **show class default** command:

```
hostname# show class default

Class Name      Members   ID   Flags
default        All      1    0001
```

Related Commands

Command	Description
class	Configures a resource class.
clear configure class	Clears the class configuration.
context	Configures a security context.
limit-resource	Sets the resource limit for a class.
member	Assigns a context to a resource class.

show conn

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

```
show conn [all | count] [state state_type] | [{{foreign | local} ip [-ip2] netmask mask}] | [long | detail]
| [{{lport | fport} port1 [-port2]] | [protocol {tcp | udp}]
```

Syntax Description

all	Display 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	Displays connections in detail, including translation type and interface information.
foreign	Displays connections with the specified foreign IP address.
fport	Displays connections with the specified foreign port.
<i>ip</i>	IP address in dotted-decimal format or beginning address in a range of IP addresses.
<i>-ip2</i>	(Optional) Ending IP address in a range of IP addresses.
local	Displays connections with the specified local IP address.
long	(Optional) Displays connections in long format.
lport	Displays connections with the specified local port.
netmask	Specifies a subnet mask for use with the given IP address.
<i>mask</i>	Subnet mask in dotted-decimal format.
<i>port1</i>	Port number or beginning port number in a range of port numbers.
<i>-port2</i>	(Optional) Ending port number in a range of port numbers.
protocol	(Optional) Specifies the connection protocol.
state	(Optional) Displays the state of specified connections.
<i>state_type</i>	Specifies the connection state type. See Table 7 for a list of the keywords available for connection state types.
tcp	Displays TCP protocol connections.
udp	Displays UDP protocol connections.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
1.1(1)	This command was introduced.
3.2(1)	The b state for TCP state bypass and X state for xlate bypass were added.

Usage Guidelines

The **show conn** command displays the number of active TCP connections, and provides information about connections of various types. Use the **show conn all** command to see the entire table of connections.

**Note**

When the FWSM 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 local** command.

The connection types that you can specify using the **show conn state** command are defined in [Table 7](#). When specifying multiple connection types, use commas without spaces to separate the keywords.

Table 25-5 Connection State Types

Keyword	Connection Type Displayed
up	Connections in the up state.
conn_inbound	Inbound connections.
ctiqbe	CTIQBE connections
data_in	Inbound data connections.
data_out	Outbound data connections.
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.
sip	SIP connections.
skinny	SCCP connections.
smtp_data	SMTP mail data connections.
sqlnet_fixup_data	SQL*Net data inspection engine connections.

When you use the **detail** option, the system displays information about the translation type and interface information using the connection flags defined in [Table 8](#).

Table 25-6 Connection Flags

Flag	Description
a	awaiting outside ACK to SYN
A	awaiting inside ACK to SYN
b	State bypass
B	initial SYN from outside
C	Computer Telephony Interface Quick Buffer Encoding (CTIQBE) media connection
d	dump
D	UDP DNS
E	outside back connection
f	inside FIN
F	outside FIN
g	Media Gateway Control Protocol (MGCP) connection
G	connection is part of a group ¹
h	H.225
H	H.323
i	incomplete TCP or UDP connection
I	inbound data
j	GTP data
J	GTP control
k	Skinny Client Control Protocol (SCCP) media connection
K	GTP t3-response
m	SIP media connection
M	SMTP data
n	GUP
N	Supervisor-based acceleration connection
O	outbound data
p	PISA connection
P	inside back connection
q	SQL*Net data
r	inside acknowledged FIN
R	outside acknowledged FIN for TCP connection
R	UDP SunRPC ²
s	awaiting outside SYN
S	awaiting inside SYN
t	SIP transient connection ³
T	SIP connection ⁴
U	up

Table 25-6 Connection Flags (continued)

Flag	Description
X	xlate creation bypassed
W	WAAS session

1. 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.
2. Because each row of **show conn** command output represents one connection (TCP or UDP), there will be only one R flag per row.
3. For UDP connections, the value t indicates that it will timeout after one minute.
4. For UDP connections, the value T indicates that the connection will timeout according to the value specified using the **timeout sip** command.

**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 FWSM 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 inactivity defined by the **conn timeout** command (by default, 1:00:00), the connection is closed and the corresponding conn flag entries are no longer displayed.

Examples

When specifying multiple connection types, use commas without spaces to separate the keywords. The following is sample output including RPC, H.323, and SIP connection information in the Up state from the **show conn** command:

```
hostname# show conn state up,rpc,h323,sip
```

The following is sample output that shows a TCP session connection from inside host 10.1.1.15 to the outside Telnet server at 192.168.49.10. 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.

```
hostname# show conn
2 in use, 2 most used
TCP out 192.168.49.10:23 in 10.1.1.15:1026 idle 0:00:22
Bytes 1774 flags UIO
UDP out 192.168.49.10:31649 in 10.1.1.15:1028 idle 0:00:14
flags D-
```

The following sample output that shows a UDP connection from outside host 192.168.49.10 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.

```
hostname(config)# show conn detail
2 in use, 2 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
      B - initial SYN from outside, b - State bypass, C - CTIQBE media,
      D - DNS, d - dump, E - outside back connection, 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 - Skinny media,
      M - SMTP data, m - SIP media, n - GUP, O - outbound data,
      P - inside back connection, 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
      X - xlate creation bypassed
TCP outside:192.168.49.10/23 inside:10.1.1.15/1026 flags UIO
UDP outside:192.168.49.10/31649 inside:10.1.1.15/1028 flags dD
```

The following is sample output from a GRE session connection (PROT:47) from host 172.16.2.1 to host 172.16.112.2. Because it is a non TCP connection, it is unidirectional and there are no flags.

```
hostname# show conn
2 in use, 2 most used
Network Processor 1 connections
PROT:47 out 172.16.112.2 in 172.16.2.1 idle 0:00:08
Bytes 18
```

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

```
hostname# show conn all
6 in use, 6 most used
TCP out 209.165.201.1:80 in 10.3.3.4:1404 idle 0:00:00 Bytes 11391
TCP out 209.165.201.1:80 in 10.3.3.4:1405 idle 0:00:00 Bytes 3709
TCP out 209.165.201.1:80 in 10.3.3.4:1406 idle 0:00:01 Bytes 2685
TCP out 209.165.201.1:80 in 10.3.3.4:1407 idle 0:00:01 Bytes 2683
TCP out 209.165.201.1:80 in 10.3.3.4:1403 idle 0:00:00 Bytes 15199
TCP out 209.165.201.1:80 in 10.3.3.4:1408 idle 0:00:00 Bytes 2688
UDP out 209.165.201.7:24 in 10.3.3.4:1402 idle 0:01:30
UDP out 209.165.201.7:23 in 10.3.3.4:1397 idle 0:01:30
UDP out 209.165.201.7:22 in 10.3.3.4:1395 idle 0:01:30
```

In this example, host 10.3.3.4 on the inside has accessed a website at 209.165.201.1. The global address on the outside interface is 209.165.201.7.

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

```
hostname# show conn detail
0 in use, 26152 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
      B - initial SYN from outside, b - State bypass, C - CTIQBE media,
      D - DNS, d - dump, E - outside back connection, 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 - Skinny media,
      M - SMTP data, m - SIP media, n - GUP, O - outbound data,
      P - inside back connection, 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
      X - xlate creation bypassed
Network Processor 1 connections
```

Related Commands	Commands	Description
	inspect ctiqbe	Enables CTIQBE application inspection.
	inspect h323	Enables H.323 application inspection.
	inspect mgcp	Enables MGCP application inspection.
	inspect sip	Removes java applets from HTTP traffic.
	inspect skinny	Enables SCCP application inspection.

show console-output

To display the currently captured console output, use the **show console-output** command in privileged EXEC mode. The FWSM automatically captures output destined for the internal console port. Do not use the internal console port unless you are advised to do so by Cisco TAC. This command allows you to view console output on your Telnet or SSH session.

show console-output

Syntax Description This command has no arguments or keywords.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	•

Release	Modification
1.1(1)	This command was introduced.

Usage Guidelines Information that displays only on a console port includes output from the **perfmom** command, startup messages, and some debug messages. The console buffer is a maximum of 1 K, and is not user configurable.

Examples The following example shows the message that displays when there is no console output:

```
hostname# show console-output
Sorry, there are no messages to display
```

Command	Description
clear configure console	Restores the default console connection settings.

show context

To show context information including allocated interfaces and the configuration file URL, the number of contexts configured, or from the system execution space, a list of all contexts, use the **show context** command in privileged EXEC mode.

show context [*name* | **detail** | **count**]

Syntax Description		
count	(Optional) Shows the number of contexts configured.	
detail	(Optional) Shows additional detail about the context(s) including the running state and information for internal use.	
<i>name</i>	(Optional) Sets the context name. If you do not specify a name, the FWSM displays all contexts. Within a context, you can only enter the current context name.	

Defaults

In the system execution space, the FWSM displays all contexts if you do not specify a name.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	—	•	•

Command History

Release	Modification
2.2(1)	This command was introduced.

Usage Guidelines

See the “[Examples](#)” section for a description of the display output.

Examples

The following is sample output from the **show context** command. The following sample display shows three contexts:

```
Context Name      Class      Interfaces      Mode      URL
*admin           default   Vlan100,101    Routed    disk:/admin.cfg
contexta         Gold      Vlan200,201    Transparent  disk:/contexta.cfg
contextb         Silver    Vlan300,301    Routed    disk:/contextb.cfg
Total active Security Contexts: 3
```

[Table 25-7](#) shows each field description.

Table 25-7 show context Fields

Field	Description
Context Name	Lists all context names. The context name with the asterisk (*) is the admin context.
Class	Shows the resource class to which the context belongs.
Interfaces	Shows the interfaces assigned to the context.
Mode	Shows the firewall mode for each context, either Routed or Transparent.
URL	Shows the URL from which the FWSM loads the context configuration.

The following is sample output from the **show context detail** command:

```
hostname# show context detail

Context "admin", has been created, but initial ACL rules not complete
  Config URL: disk:/admin.cfg
  Real Interfaces: Vlan100
  Mapped Interfaces: Vlan100
  Class: default, Flags: 0x00000013, ID: 1

Context "ctx", has been created, but initial ACL rules not complete
  Config URL: disk:/ctx.cfg
  Real Interfaces: Vlan10,20,30
  Mapped Interfaces: int1, int2, int3
  Class: default, Flags: 0x00000011, ID: 2

Context "system", is a system resource
  Config URL: startup-config
  Real Interfaces:
  Mapped Interfaces: Vlan100,10,20,30
  Class: default, Flags: 0x00000019, ID: 257

Context "null", is a system resource
  Config URL: ... null ...
  Real Interfaces:
  Mapped Interfaces:
  Class: default, Flags: 0x00000009, ID: 258
```

Table 25-8 shows each field description.

Table 25-8 Context States

Field	Description
Context	The context name. The null context information is for internal use only. The system context represents the system execution space.
State Message:	The context state. See the possible messages below.
Has been created, but initial ACL rules not complete	The FWSM parsed the configuration but has not yet downloaded the default ACLs to establish the default security policy. The default security policy applies to all contexts initially, and includes disallowing traffic from lower security levels to higher security levels, enabling application inspection, and other parameters. This security policy ensures that no traffic can pass through the FWSM after the configuration is parsed but before the configuration ACLs are compiled. You are unlikely to see this state because the configuration ACLs are compiled very quickly.

Table 25-8 Context States

Field	Description
Has been created, but not initialized	You entered the context name command, but have not yet entered the config-url command.
Has been created, but the config hasn't been parsed	The default ACLs were downloaded, but the FWSM has not parsed the configuration. This state might exist because the configuration download might have failed because of network connectivity issues, or you have not yet entered the config-url command. To reload the configuration, from within the context, enter copy startup-config running-config . From the system, reenter the config-url command. Alternatively, you can start configuring the blank running configuration.
Is a system resource	This state applies only to the system execution space and to the null context. The null context is used by the system, and the information is for internal use only.
Is a zombie	You deleted the context using the no context or clear context command, but the context information persists in memory until the FWSM reuses the context ID for a new context, or you restart.
Is active	This context is currently running and can pass traffic according to the context configuration security policy.
Is ADMIN and active	This context is the admin context and is currently running.
Was a former ADMIN, but is now a zombie	You deleted the admin context using the clear configure context command, but the context information persists in memory until the FWSM reuses the context ID for a new context, or you restart.
Real Interfaces	The interfaces assigned to the context. If you mapped the interface IDs in the allocate-interface command, this display shows the real name of the interface. The system execution space includes all interfaces.
Mapped Interfaces	If you mapped the interface IDs in the allocate-interface command, this display shows the mapped names. If you did not map the interfaces, the display lists the real names again.
Class	The resource class to which the context belongs.
Flag	For internal use only.
ID	An internal ID for this context.

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

```
hostname# show context count
Total active contexts: 2
```

Related Commands

Command	Description
admin-context	Sets the admin context.
allocate-interface	Assigns interfaces to a context.
changeto	Changes between contexts or the system execution space.

Command	Description
config-url	Specifies the location of the context configuration.
context	Creates a security context in the system configuration and enters context configuration mode.

show counters

To display the protocol stack counters, use the **show counters** command in privileged EXEC mode.

```
show counters [all | context context-name | summary | top n ] [detail]
               [protocol protocol_name[:counter_name]] [threshold n]
```

Syntax Description

all	(Multiple mode only) Displays counters for all contexts.
context <i>context-name</i>	(Multiple mode only) Specifies the context name for which to show counters.
:counter_name	Specifies a counter by name.
detail	Displays additional counter information.
protocol <i>protocol_name</i>	Displays the counters for the specified protocol.
summary	(Multiple mode only) Shows all context counters combined.
threshold <i>n</i>	Displays only those counters at or above the specified threshold. The range is 1 through 4294967295.
top <i>n</i>	(Multiple mode only) Shows the contexts that are the top <i>n</i> users of the specified counter. You must specify a counter name with this option. The range is 1 through 4294967295.

Defaults

For multiple context mode, the default context is **summary**, which shows counters for every context. For single mode, the context name is ignored and the output shows the “context” as “single_vf.”

The default count threshold is **1**.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	•

Command History

Release	Modification
2.2(1)	This command was introduced.

Examples

The following example shows how to display all counters:

```
hostname# show counters all
Protocol      Counter      Value  Context
IOS_IPC      IN_PKTS      2      admin
IOS_IPC      OUT_PKTS     2      admin
IOS_IPC      IN_PKTS     15     customera
IOS_IPC      OUT_PKTS     6      customera
```

The following example shows how to display a summary of counters:

```
hostname# show counters
Protocol      Counter      Value  Context
NPCP          IN_PKTS      7195   Summary
NPCP          OUT_PKTS     7603   Summary
IOS_IPC       IN_PKTS      869    Summary
IOS_IPC       OUT_PKTS     865    Summary
IP            IN_PKTS      380    Summary
IP            OUT_PKTS     411    Summary
IP            TO_ARP       105    Summary
IP            TO_UDP       9       Summary
UDP           IN_PKTS      9       Summary
UDP           DROP_NO_APP  9       Summary
FIXUP         IN_PKTS      202    Summary
```

The following example shows how to display counters for a context:

```
hostname# show counters context admin
Protocol      Counter      Value  Context
IOS_IPC       IN_PKTS      4      admin
IOS_IPC       OUT_PKTS     4      admin
```

Related Commands

Command	Description
clear counters	Clears the protocol stack counters.
show counters description	Shows a list of protocol counters.

show counters description

To display the protocol stack counter descriptions, use the **show counters description** command in privileged EXEC mode.

show counters description

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	—	•

Command History

Release	Modification
2.2(1)	This command was introduced.

Examples

The following is sample output from the **show counters description** command:

```
hostname# show counters description
Protocol      Counter      Description
NPCP          IN_PKTS      Packets from network processors
NPCP          OUT_PKTS     Packets to network processors
NPCP          DROP_LIMIT1  Gigamac packets dropped due to IP protocol que
ue limiter
NPCP          DROP_LIMIT2  Gigamac packets dropped due to ARP protocol qu
eue limiter
NPCP          DROP_LIMIT3  Gigamac packets dropped due to Fixup queue lim
iter
...
```

Related Commands

Command	Description
clear counters	Clears the protocol stack counters.
show counters	Shows the protocol stack counters.

show cpu

To display the CPU utilization information, use the **show cpu usage** command in privileged EXEC mode.

show cpu [usage]

From the system configuration in multiple context mode:

show cpu [usage] [context {all | context_name}]

Syntax Description

all	Specifies that the display show all contexts.
context	Specifies that the display show a context.
context_name	Specifies the name of the context to display.
usage	(Optional) Displays the CPU usage.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
1.1(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** command to find process related loads (that is, activity on behalf of items listed by the output of the **show process** command in both single mode and from the system configuration in multiple context mode).

Further, you can request, when in multiple context mode, a breakdown of the process related load to CPU consumed by any configured contexts by changing to each context and entering the **show cpu** command or by entering the **show cpu context** variant of this command.

While process related load is rounded to the nearest whole number, context related loads include one additional decimal digit of precision. For example, entering **show cpu** from the system context produces a different number than from entering the **show cpu context system** command. The former is an approximate summary of everything in **show cpu context all**, and the latter is only a portion of that summary.

Examples

The following example shows how to display the CPU utilization:

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

This example shows how to display the CPU utilization for the system context in multiple mode:

```
hostname# show cpu context system
CPU utilization for 5 seconds = 9.1%; 1 minute: 9.2%; 5 minutes: 9.1%
```

The following shows how to display the CPU utilization for all contexts:

```
hostname# show cpu usage context all
5 sec 1 min 5 min Context Name
9.1% 9.2% 9.1% system
0.0% 0.0% 0.0% admin
5.0% 5.0% 5.0% one
4.2% 4.3% 4.2% two
```

This example shows how to display the CPU utilization for a context named “one”:

```
hostname/one# show cpu usage
CPU utilization for 5 seconds = 5.0%; 1 minute: 5.0%; 5 minutes: 5.0%
```

Related Commands

Command	Description
show counters	Displays the protocol stack counters.

show cpu threshold

To display the CPU usage information when the configured rising threshold is reached and remains for the configured monitoring interval period, use the **show cpu threshold** command in privileged EXEC mode.

show cpu threshold

Syntax Description This command has no keywords and no arguments.

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Release	Modification
3.2(1)	This command was introduced.

Usage Guidelines The CPU usage threshold is computed using an approximation of the load for the configured monitoring period, and then by feeding this approximation into two moving averages.

Examples The following example shows how to display the CPU usage threshold:

```
hostname# show cpu threshold
CPU utilization RisingThresholdValue = 60%; RisingThresholdPeriod = 300secs
```

Command	Description
show cpu usage	Displays the CPU usage information.

show crashinfo

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

show crashinfo [save]

Syntax Description

save	(Optional) Displays if the FWSM is configured to save crash information to Flash memory or not.
-------------	---

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	—	•

Command History

Release	Modification
3.1	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).

If there is no crash data saved in flash, or if the crash data has been cleared by entering the **clear crashinfo** command, the **show crashinfo** command displays an error message.

Examples

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

```
hostname# show crashinfo save
crashinfo save enable
```

The following example shows the output for a crash file test. (However, this test does not actually crash the FWSM. It provides a simulated example file.)

```
hostname(config)# crashinfo test
hostname(config)# exit
hostname# show crashinfo
: Saved_Test_Crash
```

```
Thread Name: ci/console (Old pc 0x001a6ff5 ebp 0x00e88920)
```

```

Traceback:
0: 00323143
1: 0032321b
2: 0010885c
3: 0010763c
4: 001078db
5: 00103585
6: 00000000
   vector 0x000000ff (user defined)
       edi 0x004f20c4
       esi 0x00000000
       ebp 0x00e88c20
       esp 0x00e88bd8
       ebx 0x00000001
       edx 0x00000074
       ecx 0x00322f8b
       eax 0x00322f8b
error code n/a
   eip 0x0010318c
   cs 0x00000008
   eflags 0x00000000
   CR2 0x00000000
Stack dump: base:0x00e8511c size:16384, active:1476
0x00e89118: 0x004f1bb4
0x00e89114: 0x001078b4
0x00e89110-0x00e8910c: 0x00000000
0x00e89108-0x00e890ec: 0x12345678
0x00e890e8: 0x004f1bb4
0x00e890e4: 0x00103585
0x00e890e0: 0x00e8910c
0x00e890dc-0x00e890cc: 0x12345678
0x00e890c8: 0x00000000
0x00e890c4-0x00e890bc: 0x12345678
0x00e890b8: 0x004f1bb4
0x00e890b4: 0x001078db
0x00e890b0: 0x00e890e0
0x00e890ac-0x00e890a8: 0x12345678
0x00e890a4: 0x001179b3
0x00e890a0: 0x00e890b0
0x00e8909c-0x00e89064: 0x12345678
0x00e89060: 0x12345600
0x00e8905c: 0x20232970
0x00e89058: 0x616d2d65
0x00e89054: 0x74002023
0x00e89050: 0x29676966
0x00e8904c: 0x6e6f6328
0x00e89048: 0x31636573
0x00e89044: 0x7069636f
0x00e89040: 0x64786970
0x00e8903c-0x00e88e50: 0x00000000
0x00e88e4c: 0x000a7473
0x00e88e48: 0x6574206f
0x00e88e44: 0x666e6968
0x00e88e40: 0x73617263
0x00e88e3c-0x00e88e38: 0x00000000
0x00e88e34: 0x12345600
0x00e88e30-0x00e88dfc: 0x00000000
0x00e88df8: 0x00316761
0x00e88df4: 0x74706100
0x00e88df0: 0x12345600
0x00e88dec-0x00e88ddc: 0x00000000
0x00e88dd8: 0x00000070
0x00e88dd4: 0x616d2d65

```

```

0x00e88dd0: 0x74756f00
0x00e88dcc: 0x00000000
0x00e88dc8: 0x00e88e40
0x00e88dc4: 0x004f20c4
0x00e88dc0: 0x12345600
0x00e88dbc: 0x00000000
0x00e88db8: 0x00000035
0x00e88db4: 0x315f656c
0x00e88db0: 0x62616e65
0x00e88dac: 0x0030fcf0
0x00e88da8: 0x3011111f
0x00e88da4: 0x004df43c
0x00e88da0: 0x0053fef0
0x00e88d9c: 0x004f1bb4
0x00e88d98: 0x12345600
0x00e88d94: 0x00000000
0x00e88d90: 0x00000035
0x00e88d8c: 0x315f656c
0x00e88d88: 0x62616e65
0x00e88d84: 0x00000000
0x00e88d80: 0x004f20c4
0x00e88d7c: 0x00000001
0x00e88d78: 0x01345678
0x00e88d74: 0x00f53854
0x00e88d70: 0x00f7f754
0x00e88d6c: 0x00e88db0
0x00e88d68: 0x00e88d7b
0x00e88d64: 0x00f53874
0x00e88d60: 0x00e89040
0x00e88d5c-0x00e88d54: 0x12345678
0x00e88d50-0x00e88d4c: 0x00000000
0x00e88d48: 0x004f1bb4
0x00e88d44: 0x00e88d7c
0x00e88d40: 0x00e88e40
0x00e88d3c: 0x00f53874
0x00e88d38: 0x004f1bb4
0x00e88d34: 0x0010763c
0x00e88d30: 0x00e890b0
0x00e88d2c: 0x00e88db0
0x00e88d28: 0x00e88d88
0x00e88d24: 0x0010761a
0x00e88d20: 0x00e890b0
0x00e88d1c: 0x00e88e40
0x00e88d18: 0x00f53874
0x00e88d14: 0x0010166d
0x00e88d10: 0x0000000e
0x00e88d0c: 0x00f53874
0x00e88d08: 0x00f53854
0x00e88d04: 0x0048b301
0x00e88d00: 0x00e88d30
0x00e88cfc: 0x0000000e
0x00e88cf8: 0x00f53854
0x00e88cf4: 0x0048a401
0x00e88cf0: 0x00f53854
0x00e88cec: 0x00f53874
0x00e88ce8: 0x0000000e
0x00e88ce4: 0x0048a64b
0x00e88ce0: 0x0000000e
0x00e88cdc: 0x00f53874
0x00e88cd8: 0x00f7f96c
0x00e88cd4: 0x0048b4f8
0x00e88cd0: 0x00e88d00
0x00e88ccc: 0x0000000f
0x00e88cc8: 0x00f7f96c

```

```

0x00e88cc4-0x00e88cc0: 0x0000000e
0x00e88cbc: 0x00e89040
0x00e88cb8: 0x00000000
0x00e88cb4: 0x00f5387e
0x00e88cb0: 0x00f53874
0x00e88cac: 0x00000002
0x00e88ca8: 0x00000001
0x00e88ca4: 0x00000009
0x00e88ca0-0x00e88c9c: 0x00000001
0x00e88c98: 0x00e88cb0
0x00e88c94: 0x004f20c4
0x00e88c90: 0x0000003a
0x00e88c8c: 0x00000000
0x00e88c88: 0x0000000a
0x00e88c84: 0x00489f3a
0x00e88c80: 0x00e88d88
0x00e88c7c: 0x00e88e40
0x00e88c78: 0x00e88d7c
0x00e88c74: 0x001087ed
0x00e88c70: 0x00000001
0x00e88c6c: 0x00e88cb0
0x00e88c68: 0x00000002
0x00e88c64: 0x0010885c
0x00e88c60: 0x00e88d30
0x00e88c5c: 0x00727334
0x00e88c58: 0xa0ffffff
0x00e88c54: 0x00e88cb0
0x00e88c50: 0x00000001
0x00e88c4c: 0x00e88cb0
0x00e88c48: 0x00000002
0x00e88c44: 0x0032321b
0x00e88c40: 0x00e88c60
0x00e88c3c: 0x00e88c7f
0x00e88c38: 0x00e88c5c
0x00e88c34: 0x004b1ad5
0x00e88c30: 0x00e88c60
0x00e88c2c: 0x00e88e40
0x00e88c28: 0xa0ffffff
0x00e88c24: 0x00323143
0x00e88c20: 0x00e88c40
0x00e88c1c: 0x00000000
0x00e88c18: 0x00000008
0x00e88c14: 0x0010318c
0x00e88c10-0x00e88c0c: 0x00322f8b
0x00e88c08: 0x00000074
0x00e88c04: 0x00000001
0x00e88c00: 0x00e88bd8
0x00e88bfc: 0x00e88c20
0x00e88bf8: 0x00000000
0x00e88bf4: 0x004f20c4
0x00e88bf0: 0x000000ff
0x00e88bec: 0x00322f87
0x00e88be8: 0x00f5387e
0x00e88be4: 0x00323021
0x00e88be0: 0x00e88c10
0x00e88bdc: 0x004f20c4
0x00e88bd8: 0x00000000 *
0x00e88bd4: 0x004eabb0
0x00e88bd0: 0x00000001
0x00e88bcc: 0x00f5387e
0x00e88bc8-0x00e88bc4: 0x00000000
0x00e88bc0: 0x00000008
0x00e88bbc: 0x0010318c
0x00e88bb8-0x00e88bb4: 0x00322f8b

```

```

0x00e88bb0: 0x00000074
0x00e88bac: 0x00000001
0x00e88ba8: 0x00e88bd8
0x00e88ba4: 0x00e88c20
0x00e88ba0: 0x00000000
0x00e88b9c: 0x004f20c4
0x00e88b98: 0x000000ff
0x00e88b94: 0x001031f2
0x00e88b90: 0x00e88c20
0x00e88b8c: 0xffffffff
0x00e88b88: 0x00e88cb0
0x00e88b84: 0x00320032
0x00e88b80: 0x37303133
0x00e88b7c: 0x312f6574
0x00e88b78: 0x6972772f
0x00e88b74: 0x342f7665
0x00e88b70: 0x64736666
0x00e88b6c: 0x00020000
0x00e88b68: 0x00000010
0x00e88b64: 0x00000001
0x00e88b60: 0x123456cd
0x00e88b5c: 0x00000000
0x00e88b58: 0x00000008

```

```

Cisco XXX Firewall Version X.X
Cisco XXX Device Manager Version X.X

```

```

Compiled on Fri 15-Nov-04 14:35 by root

```

```

hostname up 10 days 0 hours

```

```

Hardware: XXX-XXX, 64 MB RAM, CPU Pentium 200 MHz
Flash i28F640J5 @ 0x300, 16MB
BIOS Flash AT29C257 @ 0xffffd8000, 32KB

```

```

0: ethernet0: address is 0003.e300.73fd, irq 10
1: ethernet1: address is 0003.e300.73fe, irq 7
2: ethernet2: address is 00d0.b7c8.139e, irq 9

```

```

Licensed Features:

```

```

Failover:           Disabled
VPN-DES:            Enabled
VPN-3DES-AES:      Disabled
Maximum Interfaces: 3
Cut-through Proxy: Enabled
Guards:             Enabled
URL-filtering:     Enabled
Inside Hosts:      Unlimited
Throughput:        Unlimited
IKE peers:         Unlimited

```

```

This XXX has a Restricted (R) license.

```

```

Serial Number: 480430455 (0x1ca2c977)
Running Activation Key: 0xc2e94182 0xc21d8206 0x15353200 0x633f6734
Configuration last modified by enable_15 at 13:49:42.148 UTC Wed Nov 20 2004

```

```

----- show clock -----

```

```

15:34:28.129 UTC Sun Nov 24 2004

```

```

----- show memory -----

```

```

Free memory:      50444824 bytes
Used memory:      16664040 bytes

```

```

-----
Total memory:          67108864 bytes

----- show conn count -----

0 in use, 0 most used

----- show xlate count -----

0 in use, 0 most used

----- show blocks -----

  SIZE   MAX   LOW   CNT
    4   1600  1600  1600
   80    400   400   400
  256    500   499   500
 1550   1188   795   927

----- show interface -----

interface ethernet0 "outside" is up, line protocol is up
  Hardware is i82559 ethernet, address is 0003.e300.73fd
  IP address 172.23.59.232, subnet mask 255.255.0.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
    6139 packets input, 830375 bytes, 0 no buffer
    Received 5990 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    90 packets output, 6160 bytes, 0 underruns
    0 output errors, 13 collisions, 0 interface resets
    0 babbles, 0 late collisions, 47 deferred
    0 lost carrier, 0 no carrier
    input queue (curr/max blocks): hardware (5/128) software (0/2)
    output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet1 "inside" is up, line protocol is down
  Hardware is i82559 ethernet, address is 0003.e300.73fe
  IP address 10.1.1.1, subnet mask 255.255.255.0
  MTU 1500 bytes, BW 10000 Kbit half duplex
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    1 packets output, 60 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collisions, 0 deferred
    1 lost carrier, 0 no carrier
    input queue (curr/max blocks): hardware (128/128) software (0/0)
    output queue (curr/max blocks): hardware (0/1) software (0/1)
interface ethernet2 "intf2" is administratively down, line protocol is down
  Hardware is i82559 ethernet, address is 00d0.b7c8.139e
  IP address 127.0.0.1, subnet mask 255.255.255.255
  MTU 1500 bytes, BW 10000 Kbit half duplex
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collisions, 0 deferred
    0 lost carrier, 0 no carrier
    input queue (curr/max blocks): hardware (128/128) software (0/0)
    output queue (curr/max blocks): hardware (0/0) software (0/0)

----- show cpu usage -----

CPU utilization for 5 seconds = 0%; 1 minute: 0%; 5 minutes: 0%

```

```
----- show process -----
```

	PC	SP	STATE	Runtime	SBASE	Stack	Process
Hsi	001e3329	00763e7c	0053e5c8	0	00762ef4	3784/4096	arp_timer
Lsi	001e80e9	00807074	0053e5c8	0	008060fc	3792/4096	FragDBGC
Lwe	00117e3a	009dc2e4	00541d18	0	009db46c	3704/4096	dbgtrace
Lwe	003cee95	009de464	00537718	0	009dc51c	8008/8192	Logger
Hwe	003d2d18	009e155c	005379c8	0	009df5e4	8008/8192	tcp_fast
Hwe	003d2c91	009e360c	005379c8	0	009e1694	8008/8192	tcp_slow
Lsi	002ec97d	00b1a464	0053e5c8	0	00b194dc	3928/4096	xlate clean
Lsi	002ec88b	00b1b504	0053e5c8	0	00b1a58c	3888/4096	uxlate clean
Mrd	002e3a17	00c8f8d4	0053e600	0	00c8d93c	7908/8192	tcp_intercept_times
Lsi	00423dd5	00d3a22c	0053e5c8	0	00d392a4	3900/4096	route_process
Hsi	002d59fc	00d3b2bc	0053e5c8	0	00d3a354	3780/4096	PIX Garbage Collec
Hwe	0020e301	00d5957c	0053e5c8	0	00d55614	16048/16384	isakmp_time_keepr
Lsi	002d377c	00d7292c	0053e5c8	0	00d719a4	3928/4096	perfmon
Hwe	0020bd07	00d9c12c	0050bb90	0	00d9b1c4	3944/4096	IPSec
Mwe	00205e25	00d9e1ec	0053e5c8	0	00d9c274	7860/8192	IPsec timer handler
Hwe	003864e3	00db26bc	00557920	0	00db0764	6904/8192	qos_metric_daemon
Mwe	00255a65	00dc9244	0053e5c8	0	00dc8adc	1436/2048	IP Background
Lwe	002e450e	00e7bb94	00552c30	0	00e7ad1c	3704/4096	pix/trace
Lwe	002e471e	00e7cc44	00553368	0	00e7bdcc	3704/4096	pix/tconsole
Hwe	001e5368	00e7ed44	00730674	0	00e7ce9c	7228/8192	pix/intf0
Hwe	001e5368	00e80e14	007305d4	0	00e7ef6c	7228/8192	pix/intf1
Hwe	001e5368	00e82ee4	00730534	2470	00e8103c	4892/8192	pix/intf2
H*	001a6ff5	0009ff2c	0053e5b0	4820	00e8511c	12860/16384	ci/console
Csi	002dd8ab	00e8a124	0053e5c8	0	00e891cc	3396/4096	update_cpu_usage
Hwe	002cb4d1	00f2bfbcb	0051e360	0	00f2a134	7692/8192	uauth_in
Hwe	003d17d1	00f2e0bc	00828cf0	0	00f2c1e4	7896/8192	uauth_thread
Hwe	003e71d4	00f2f20c	00537d20	0	00f2e294	3960/4096	udp_timer
Hsi	001db3ca	00f30fc4	0053e5c8	0	00f3004c	3784/4096	557mcfix
Crđ	001db37f	00f32084	0053ea40	508286220	00f310fc	3688/4096	557poll
Lsi	001db435	00f33124	0053e5c8	0	00f321ac	3700/4096	557timer
Hwe	001e5398	00f441dc	008121e0	0	00f43294	3912/4096	fover_ip0
Cwe	001dcdad	00f4523c	00872b48	120	00f44344	3528/4096	ip/0:0
Hwe	001e5398	00f4633c	008121bc	10	00f453f4	3532/4096	icmp0
Hwe	001e5398	00f47404	00812198	0	00f464cc	3896/4096	udp_thread/0
Hwe	001e5398	00f4849c	00812174	0	00f475a4	3456/4096	tcp_thread/0
Hwe	001e5398	00f495bc	00812150	0	00f48674	3912/4096	fover_ip1
Cwe	001dcdad	00f4a61c	008ea850	0	00f49724	3832/4096	ip/1:1
Hwe	001e5398	00f4b71c	0081212c	0	00f4a7d4	3912/4096	icmp1
Hwe	001e5398	00f4c7e4	00812108	0	00f4b8ac	3896/4096	udp_thread/1
Hwe	001e5398	00f4d87c	008120e4	0	00f4c984	3832/4096	tcp_thread/1
Hwe	001e5398	00f4e99c	008120c0	0	00f4da54	3912/4096	fover_ip2
Cwe	001e542d	00f4fa6c	00730534	0	00f4eb04	3944/4096	ip/2:2
Hwe	001e5398	00f50afc	0081209c	0	00f4fbb4	3912/4096	icmp2
Hwe	001e5398	00f51bc4	00812078	0	00f50c8c	3896/4096	udp_thread/2
Hwe	001e5398	00f52c5c	00812054	0	00f51d64	3832/4096	tcp_thread/2
Hwe	003d1a65	00f78284	008140f8	0	00f77fdc	300/1024	listen/http1
Mwe	0035cafa	00f7a63c	0053e5c8	0	00f786c4	7640/8192	Crypto CA

```
----- show failover -----
```

```
No license for Failover
```

```
----- show traffic -----
```

```
outside:
```

```
received (in 865565.090 secs):
    6139 packets    830375 bytes
     0 pkts/sec     0 bytes/sec
transmitted (in 865565.090 secs):
```

show crashinfo

```

          90 packets      6160 bytes
          0 pkts/sec     0 bytes/sec
inside:
  received (in 865565.090 secs):
    0 packets      0 bytes
    0 pkts/sec     0 bytes/sec
  transmitted (in 865565.090 secs):
    1 packets      60 bytes
    0 pkts/sec     0 bytes/sec
intf2:
  received (in 865565.090 secs):
    0 packets      0 bytes
    0 pkts/sec     0 bytes/sec
  transmitted (in 865565.090 secs):
    0 packets      0 bytes
    0 pkts/sec     0 bytes/sec

```

```
----- show perfmon -----
```

```

PERFMON STATS:   Current   Average
Xlates           0/s      0/s
Connections      0/s      0/s
TCP Conns        0/s      0/s
UDP Conns        0/s      0/s
URL Access       0/s      0/s
URL Server Req   0/s      0/s
TCP Fixup        0/s      0/s
TCPIntercept     0/s      0/s
HTTP Fixup       0/s      0/s
FTP Fixup        0/s      0/s
AAA Authen       0/s      0/s
AAA Author       0/s      0/s
AAA Account      0/s      0/s
: End_Test_Crash

```

Related Commands

Command	Description
clear crashinfo	Deletes the contents of the crash file.
crashinfo force	Forces a crash of the FWSM.
crashinfo save disable	Disables crash information from writing to Flash memory.
crashinfo test	Tests the ability of the FWSM to save crash information to a file in Flash memory.

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 in global configuration or privileged EXEC mode.

show crypto accelerator statistics

Syntax Description

This command has no keywords or variables.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	•	•	•	—	—
Privileged EXEC	•	•	•	—	—

Command History

Release	Modification
3.1(1)	This command was introduced.

Examples

The following example entered in global configuration mode, displays global crypto accelerator statistics:

```
hostname # 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
hostname #

```

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 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 in privileged EXEC mode.

show crypto ca certificates [*trustpointname*]

Syntax Description	<i>trustpointname</i>	(Optional) Specifies the name of a trustpoint. If you do not specify a name, this command displays all certificates installed on the system.
---------------------------	-----------------------	--

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History	Release	Modification
	3.1(1)	This command was introduced.

Examples The following example entered in global configuration mode, displays a CA certificate for a trustpoint named tp1:

```
hostname# 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 = a@b.con
  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
hostname#

```

Related Commands

Command	Description
crypto ca authenticate	Obtains a CA certificate for a specified trustpoint.
crypto ca crl request	Requests a CRL based on the configuration parameters of a specified trustpoint.
crypto ca enroll	Initiates the enrollment process with a CA.
crypto ca import	Imports a certificate to a specified trustpoint.
crypto ca trustpoint	Enters trustpoint mode for a specified trustpoint.

show crypto ca crls

To display all cached CRLs or to display all CRLs cached for a specified trustpoint, use the **show crypto ca crls** command in privileged EXEC mode.

```
show crypto ca crls [trustpointname]
```

Syntax Description	<i>trustpointname</i>	(Optional) Specifies the name of a trustpoint. If you do not specify a name, this command displays all CRLs cached on the system.
--------------------	-----------------------	---

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Privileged EXEC	•	•	•	•	—

Command History	Release	Modification
	3.1(1)	This command was introduced.

Examples The following example entered in global configuration mode, displays a CRL for a trustpoint named tp1:

```
hostname# show crypto ca crls tp1
CRL Issuer Name:
  cn=ms-sub1-ca-5-2004,ou=Franklin DevTest,o=Cisco
Systems, l=Franklin, st=MA, c=US, ea=user@cisco.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
```

Related Commands	Command	Description
	crypto ca authenticate	Obtains a CA certificate for a specified trustpoint.
	crypto ca crl request	Requests a CRL based on the configuration parameters of a specified trustpoint.
	crypto ca enroll	Initiates the enrollment process with a CA.
	crypto ca import	Imports a certificate to a specified trustpoint.
	crypto ca trustpoint	Enters trustpoint mode for a specified trustpoint.

show crypto ipsec df-bit

To display the IPsec DF-bit policy for IPsec packets for a specified interface, use the **show crypto ipsec df-bit** command in global configuration mode and privileged EXEC mode.

show crypto ipsec df-bit *interface*

Syntax Description

<i>interface</i>	Specifies an interface name.
token	Indicates a token-based server for user authentication is used.

Defaults

No default behaviors or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	•	•	•	—	—
Privileged EXEC	•	•	•	—	—

Command History

Release	Modification
1.1(1)	This command was introduced.
3.1(1)	This command was changed from show crypto ipsec .

Examples

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

```
hostname(config)# show crypto ipsec df-bit inside
df-bit inside copy
hostname(config)#
```

Related Commands

Command	Description
crypto ipsec df-bit	Configures the IPsec DF-bit policy for IPsec packets.
crypto ipsec fragmentation	Configures the fragmentation policy for IPsec packets.
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 in global configuration or privileged EXEC modes.

show crypto ipsec fragmentation *interface*

Syntax Description

<i>interface</i>	Specifies an interface name.
token	Indicates a token-based server for user authentication is used.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Global configuration	•	•	•	—	—
Privileged EXEC	•	•	•	—	—

Command History

Release	Modification
1.1(1)	This command was introduced.
3.1(1)	This command was changed from show crypto ipsec .

Examples

The following example, entered in global configuration mode, displays the IPSec fragmentation policy for an interface named inside:

```
hostname(config)# show crypto ipsec fragmentation inside
fragmentation inside before-encryption
hostname(config)#
```

Related Commands

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

show crypto key mypubkey

To display key pairs of the indicated type, use the **show crypto key mypubkey** command in privileged EXEC mode.

```
show crypto key mypubkey {rsa | dsa}
```

Syntax Description

dsa	Displays DSA key pairs.
rsa	Displays RSA key pairs.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	—

Command History

Release	Modification
3.1(1)	This command was introduced.

Examples

The following example entered in global configuration mode, displays RSA key pairs:

```
hostname(config)# show crypto key mypubkey rsa
...
```

Related Commands

Command	Description
crypto key generate dsa	Generates DSA key pairs.
crypto key generate rsa	Generates RSA key pairs.
crypto key zeroize	Removes all key pairs of the indicated type.

show crypto protocol statistics

To display the protocol-specific statistics in the crypto accelerator MIB, use the **show crypto protocol statistics** command in global configuration or privileged EXEC mode.

show crypto protocol statistics *protocol*

Syntax Description	<i>protocol</i>	<p>Specifies the name of the protocol for which to display statistics. Protocol choices are as follows:</p> <ul style="list-style-type: none"> ikev1—Internet Key Exchange version 1. ipsec—IP Security Phase-2 protocols. ssl—Secure Socket Layer. other—Reserved for new protocols. all—All protocols currently supported.
---------------------------	-----------------	--

Defaults No default behavior or values.

Command Modes The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Global configuration	•	•	•	—	—
Privileged EXEC	•	•	•	—	—

Command History	Release	Modification
	3.1(1)	This command was introduced.

Examples The following examples entered in global configuration mode, display crypto accelerator statistics for specified protocols:

```
hostname # show crypto protocol statistics ikev1
[IKEv1 statistics]
  Encrypt packet requests: 39
  Encapsulate packet requests: 39
  Decrypt packet requests: 35
  Decapsulate packet requests: 35
  HMAC calculation requests: 84
  SA creation requests: 1
  SA rekey requests: 3
  SA deletion requests: 2
  Next phase key allocation requests: 2
  Random number generation requests: 0
  Failed requests: 0
```

```
hostname # show crypto protocol statistics ipsec
[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

hostname # show crypto protocol statistics ssl
[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

hostname # show crypto protocol statistics other
[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

hostname # 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
hostname #

```

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 ctiqbe

To display information about CTIQBE sessions established across the FWSM, use the **show ctiqbe** command in privileged EXEC mode.

show ctiqbe

Syntax Description

This command has no arguments or keywords.

Defaults

No default behavior or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple	
				Context	System
Privileged EXEC	•	•	•	•	•

Command History

Release	Modification
3.1(1)	This command was introduced.

Usage Guidelines

The **show ctiqbe** command displays information of CTIQBE sessions established across the FWSM. Along with **debug ctiqbe** and **show local-host**, this command is used for troubleshooting CTIQBE inspection engine issues.



Note

We recommend that you have the **pager** command configured before using the **show ctiqbe** command. If there are a lot of CTIQBE sessions and the **pager** command is not configured, it can take a while for the **show ctiqbe** command output to reach the end.

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 FWSM. 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 CallManager at 172.29.1.77, where TCP port 2748 is the Cisco CallManager. The heartbeat interval for the session is 120 seconds.

```
hostname# 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 FWSM 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.

The following is the xlate information for these CTIBQE connections:

```

hostname# show xlate debug
3 in use, 3 most used
Flags: D|DNS, d|dump, I|identity, i|inside, n|no random,
       |o|outside, r|portmap, s|static
TCP PAT from inside:10.0.0.99/1117 to outside:172.29.1.99/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:172.29.1.99/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:172.29.1.99/1029 flags ri idle 0:00:23
timeout 0:04:10

```

Related Commands

Commands	Description
class-map	Defines the traffic class to which to apply security actions.
inspect ctiqbe	Enables CTIQBE application inspection.
service-policy	Applies a policy map to one or more interfaces.
show conn	Displays the connection state for different connection types.
timeout	Sets the maximum idle time duration for different protocols and session types.

show curpriv

To display the current user privileges, use the **show curpriv** command:

```
show curpriv
```

Syntax Description

This command has no arguments or keywords.

Defaults

No default behaviors or values.

Command Modes

The following table shows the modes in which you can enter the command:

Command Mode	Firewall Mode		Security Context		
	Routed	Transparent	Single	Multiple Context	System
Global configuration	•	•	—	—	•
Privileged EXEC	•	•	—	—	•
Unprivileged	•	•	—	—	•

Command History

Release	Modification
1.1(1)	This command was introduced.

Usage Guidelines

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

Examples

These examples show output from the **show curpriv** command when a user named enable_15 is at different privilege levels. The username indicates the name that the user entered when the user logged in, P_PRIV indicates that the user has entered the **enable** command, and P_CONF indicates that the user has entered the **config terminal** command.

```
hostname(config)# show curpriv
Username : enable_15
Current privilege level : 15
Current Mode/s : P_PRIV P_CONF
hostname(config)# exit
```

```
hostname(config)# show curpriv
Username : enable_15
Current privilege level : 15
Current Mode/s : P_PRIV
hostname(config)# exit
```

```
hostname(config)# show curpriv
Username : enable_1
```

■ show curpriv

```
Current privilege level : 1
Current Mode/s : P_UNPR
hostname(config)#
```

Related Commands

Command	Description
clear configure privilege	Remove privilege command statements from the configuration.
show running-config privilege	Display privilege levels for commands.

