a - clear e

aaa-server active, fail

To reactivate a AAA server that is marked failed, use the **aaa-server active** command. To fail an active AAA server, use the **aaa-server fail** command.

aaa-server *groupname* {**active** | **fail**} **host** *hostname*

Syntax Description

active	Sets the server to an active state.
fail	Sets the server to a failed state.
groupname	AAA server group or realm name.
host *hostname*	FQDN or IP address of the server being acted upon.

Command History

Release	Modification
6.2.1	This command was introduced.

Usage Guidelines

Without this command, servers in a group that failed remain in a failed state until all servers in the group fail, after which all are reactivated. You can find the server group or realm name, as well as all the AAA server information in the output of the **show aaa-server** command.

Examples

The following example shows the state for server 192.168.125.60 in group1, and manually reactivates it:

```
> show aaa-server group1 host 192.68.125.60
Server Group: group1
Server Protocol: RADIUS
Server Address: 192.68.125.60
Server port: 1645
Server status: FAILED. Server disabled at 11:10:08 UTC Fri Aug...
>
> aaa-server group1 active host 192.168.125.60
>
> show aaa-server group1 host 192.68.125.60
Server Group: group1
Server Protocol: RADIUS
Server Address: 192.68.125.60
Server port: 1645
Server status: ACTIVE (admin initiated). Last Transaction at 11:40:09 UTC Fri Aug...
```

Related Commands

Commands	Description
clear aaa-server statistics	Clears AAA server statistics.
show aaa-server	Displays AAA server statistics
test aaa-server	Verify the configuration for a AAA server.

app-agent heartbeat

To configure the heartbeat message interval for the app-agent (application agent) running on the threat defense device, use the **app-agent heartbeat** command.

app-agent heartbeat [**interval** *milliseconds*] [**retry-count** *integer*]

Syntax Description

interval *milliseconds*	Specifies the time interval in milliseconds between heartbeat messages. You can adjust the interval in increments of 100 milliseconds. The default is 1000. The allowed range is 100 to 6000 for release 6.2.2 and following, but 300 to 6000 for older releases. A loss of consecutive heartbeat messages up to the retry count triggers a failure notification to the rest of the system. The default of 1000 milliseconds provides an aggressive failure detection setting with the risk of false failure detections.
retry-count *integer*	Specifies the number of times the app-agent should retry the heartbeat message if there is no response, or the app-agent receives an error response for the heartbeat message, from 3 to 10. The default is 3.

Command Default

For the Firepower 2100, the default interval is 6000 milliseconds and the retry count is 10. You cannot use this command to change these values.

For other device models, the default interval value is 1000 milliseconds, and the retry count is 3.

Command History

Release	Modification
6.1	This command was introduced.
6.2.2	The allowed interval range was changed to 100 to 6000.

Usage Guidelines

The primary responsibility of the app-agent running on the threat defense device is to interface and communicate between the threat defense modules and Firepower 2100, 4100, and 9300 FXOS chassis.

The heartbeat communication channel serves the purpose of monitoring the health of the link between the FXOS chassis and the threat defense application agent. The threat defense application sends request messages to the FXOS chassis supervisor at a certain interval, with retries at a set number of times until a proper response is received from the FXOS chassis supervisor.

The heartbeat mechanism between the threat defense app-agent and FXOS chassis supervisor also monitors the Hardware Bypass feature for failure. For certain interface modules on the Firepower 2100, 4100, and 9300, you can enable the Hardware Bypass feature. Hardware Bypass ensures that traffic continues to flow between an inline interface pair during a power outage. This feature can be used to maintain network connectivity in the case of software or hardware failures.

Examples

The following example sets the app-agent heartbeat interval to 600 milliseconds and the retry count to 6 times:

```
> app-agent heartbeat interval 600 retry-count 6
```

Related Commands

Command	Description
show app-agent	Shows the app-agent status.
show inline-set	Shows inline set information.
show interface	Shows interface status information.

asp inspect-dp egress-optimization

To enable egress optimization, use the **asp inspect-dp egress-optimization** command. To disable egress optimization, use the **no** form of this command.

Egress optimization is a performance feature targeted for selected IPS traffic. The feature is enabled by default on all threat defense platforms.

Note We strongly recommend you leave this feature enabled. Disable it only if advised to do so by Cisco TAC.

asp inspect-dp egress-optimization
no asp inspect-dp egress-optimization

Command Default

Egress optimization is enabled by default.

Command History

Release	Modification
6.4	This command was introduced.

Usage Guidelines

Egress optimization is intended to be enabled at all times to improve performance. Disable egress optmization only on the advice of Cisco TAC for troubleshooting purposes.

Examples

The following example shows how to enable egress optimization:

```
> asp inspect-dp egress-optimization
```

Related Commands

Command	Description
show conn state egress_optimization	Displays information about flows eligible for egress optimization. Use this command on the advice of Cisco TAC.
show asp inspect-dp egress-optimization	Show statistics related to egress optimization.
clear asp inspect-dp egress-optimization	Clear statistics related to egress optimization.

asp load-balance per-packet

To change the load balancing behavior on multiple cores to be per packet, use the **asp load-balance per-packet** command. To restore the default load-balancing mechanism, use the **no** form of this command.

asp load-balance per-packet
no asp load-balance per-packet

Command Default

Per-packet load-balancing is disabled by default.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The job of the load balancer is to distribute packets to CPU cores and to maintain packet order. By default, a connection can only be processed by one core at a time. Due to this behavior, the cores will be under-utilized if there are a small number of interfaces/RX rings in use when compared to the number of cores. For example if there are only two Gigabit Ethernet interfaces in use on the threat defense device, then only two cores will be used. (A Ten Gigabit Ethernet interface has 4 RX rings and a Gigabit Ethernet interface as 1 RX ring.) You may want to optimize the load balancer by enabling per-packet load balancing so you can use more cores.

The default load-balancing behavior optimizes overall system performance when you have many interfaces in use, while the per-packet load balancer optimizes the overall system performance when you have a smaller number of interfaces that are active.

If you enable per-packet load balancing, when one core processes packets from an interface, another core can receive and process the next packet from the same interface. Therefore, it is possible for all cores to process packets from the same interface simultaneously.

Per-packet load balancing will improve performance if:

- The system drops packets
- The **show cpu** command shows CPU usage far less than 100%—The CPU usage is a good indicator of how many cores are being used. For example, on an 8-core system, if two cores are used, **show cpu** shows 25%; four cores: 50%; six cores: 75%.
- There are a small number of interfaces that are in use

Note Typically if there are less than 64 concurrent flows on the threat defense, then enabling per-packet load balancing will incur more overhead than its benefit.

Examples

The following example shows how to change the default load-balancing behavior:

```
> asp load-balance per-packet
```

Related Commands

Command	Description
clear asp load-balance history	Clears and resets the ASP load balancing per packet history statistics. OK
show asp load-balance	Displays a histogram of the load balancer queue sizes. OK

asp packet-profile

To obtain statistics on how a threat defense device handles network traffic, use the **asp packet-profile** command. To disable packet profiling, use the **no** form of this command.

The Accelerated Security Path or ASP process determines how many packets were fastpathed by a prefilter policy, offloaded as a large flow, fully evaluated by access control (Snort), and so on.

asp packet-profile
no asp packet-profile

Command Default

Packet profiling is enabled by default.

Command History

Release	Modification
6.5	This command was introduced.

Usage Guidelines

Packet profiling is intended to be enabled at all times. If the CPU usage is high due to statistics collection and further computation, then the feature can be disabled.

Examples

The following example shows how to enable packet profiling:

```
> asp packet-profile
```

Related Commands

Command	Description
show asp packet-profile	Displays statistics for the packets that traversed through dataplane only, the dataplane and Snort, and offloaded to hardware.
clear asp packet-profile	Clear statistics related to packet profiling.

asp rule-engine transactional-commit

Use the **asp rule-engine transactional-commit** command to enable or disable the transactional commit model for the rule engine.

asp rule-engine transactional-commit *option*
asp rule-engine transactional-commit *option*

Syntax Description

option	Enables the transactional commit model for the rule engine for the selected policies. Options include: • **access-group**—Access rules applied globally or to interfaces. • **nat**—Network address translation rules.

Command Default

By default, the transactional commit model is disabled.

Command History

Release	Modification
6.6	This command was introduced.

Usage Guidelines

By default, when you change a rule-based policy (such as access rules), the changes become effective immediately. However, this immediacy comes at a slight cost in performance. The performance cost is more noticeable for very large rule lists in a high connections-per-second environment, for example, when you change a policy with 25,000 rules while the device is handling 18,000 connections per second.

The performance is affected because the rule engine compiles rules to enable faster rule lookup. By default, the system will also search uncompiled rules when evaluating a connection attempt so that new rules can be applied; since the rules are not compiled, the search takes longer.

You can change this behavior so that the rule engine uses a transactional model when implementing rule changes, continuing to use the old rules until the new rules are compiled and ready for use. Using the transactional model, performance should not drop during the rule compilation. The following table clarifies the behavioral difference.

Model	Before Compilation	During Compilation	After Compilation
Default	Match old rules.	Match new rules. (Connections per second rate will decrease.)	Match new rules.
Transactional	Match old rules.	Match old rules. (Connections per second rate will be unaffected.)	Match new rules.

An additional benefit of the transactional model is that, when replacing an ACL on an used in an access group, there is no gap between deleting the old ACL and applying the new one. This reduces the chances that acceptable connections will be dropped during the operation.

Tip If you enable the transactional model for a rule type, there are syslog messages to mark the beginning and the end of the compilation. These messages are numbered 780001 and following.

Example

The following example enables the transactional commit model for access groups:

```
> asp rule-engine transactional-commit access-group
```

base-url

(Optional) Configures the base URL of the Clientless VPN. This URL is used in SAML metadata, which is provided to third-party IdPs, so that IdPs can redirect endpoint users back to the FTD.

(Optional) From Version 7.1, this command configures the base URL of the SAML service provider for VPN authentication. This URL is used in SAML metadata, which is provided to third-party IdPs, so that IdPs can redirect endpoint users back to the FTD.

To disable this feature, use the **no** form of this command

base-url { **value _ string** }
no base-url

Syntax Description

base-url	URL of the Clientless VPN

Command Default

None.

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
Parameters configuration	• Yes	• Yes	• Yes	• Yes	—

Command History

Release	Modification
6.3	This command was added.

Usage Guidelines

- When base-url is configured, it is the base URL of AssertionConsumerService and SingleLogoutService, and is displayed in **show saml metadata**.
- When base-url is not configured, the base URL is created from the FTD's hostname and domain-name. For example, **https://ssl-vpn.cisco.com** is the base URL in **show saml metadata** when hostname is "ssl-vpn" and domain-name is "cisco.com".
- When neither base-url or hostname and domain-name are configured, **show saml metadata** displays an error.

Examples

The following example sets up a base-url:

```
ciscoftd(config)# webvpn
ciscoftd(config-webvpn)# saml idp myIdp
ciscoftd(config-webvpn-saml-idp)# base url https://ClientlessVPN.com
```

Related Commands

Command	Description
signature	Enable or disable signature in SAML request. By default, the signature is disabled.
timeout	Configures the SAML IdP timeout.
trustpoint	Configures the trustpoint in saml-idp sub-mode.
url	Configures the SAML IdP URL.
local-base-url	Configures the local base URL of the SAML service provider for VPN authentication

blocks

To allocate additional memory to block diagnostics (displayed by the **show blocks** command), use the **blocks** command. To set the value back to the default, use the **no** form of this command.

blocks queue history enable [*memory_size*]
no blocks queue history enable [*memory_size*]

Syntax Description

memory_size	(Optional) Sets the memory size for block diagnostics in bytes, instead of applying the dynamic value. If this value is greater than free memory, an error message appears and the value is not accepted. If this value is greater than 50% of free memory, a warning message appears, but the value is accepted.

Command Default

The default memory assigned to track block diagnostics is 2136 bytes.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

To view the currently allocated memory, enter the **show blocks queue history** command.

If you reload the threat defense device, the memory allocation returns to the default.

The amount of memory allocated will be at most 150 KB, but never more than 50% of free memory. Optionally, you can specify the memory size manually.

Examples

The following example increases the memory size for block diagnostics:

```
> blocks queue history enable
```

The following example increases the memory size to 3000 bytes:

```
> blocks queue history enable 3000
```

The following example attempts to increase the memory size to 3000 bytes, but the value is more than the available free memory:

```
> blocks queue history enable 3000
ERROR: memory size exceeds current free memory
```

The following example increases the memory size to 3000 bytes, but the value is more than 50% of the free memory:

```
> blocks queue history enable 3000
WARNING: memory size exceeds 50% of current free memory
```

Related Commands

Command	Description
clear blocks	Clears the system buffer statistics.
show blocks	Shows the system buffer usage.

capture

To enable packet capture capabilities for packet sniffing and network fault isolation, use the **capture** command. To disable packet capture capabilities, use the no form of this command.

Capture network traffic:

capture *capture_name* [**type** { **asp-drop** [**all** | *drop-code*] | **raw-data** | **isakmp** [**ikev1** | **ikev2**] | **inline-tag** [*tag*] }] { **interface** { *interface_name* | **data-plane | management-plane | cplane** } } [**buffer** *buf_size*] [**file-size** *file_size*] [**ethernet-type** *type*] [**headers-only**] [**packet-length** *bytes*] [**circular-buffer**] [**trace** [**trace-count** *number*] [**match** *protocol* { **host** *source_ip* | *source_ip mask* | **any4** | **any6** } [*operator src_port*] { **host** *dest_ip* | *dest_ip mask* | **any4** | **any6** } [*operator dest_port*]]

Capture cluster control-link traffic:

capture *capture_name* **type lacp interface** *interface_id* [**buffer** *buf_size*] [**packet-length** *bytes*] [**circular-buffer**] [**real-time** [**dump**] [**detail**]]
capture *capture_name* **interface cluster** [**buffer** *buf_size*] [**ethernet-type** *type*] [**packet-length** *bytes*] [**circular-buffer**] [**trace** [**trace-count** *number*]] [**real-time** [**dump**] [**detail**]] [**trace**] [**match** *protocol* { **host** *source_ip* | *source_ip mask* | **any4** | **any6** } [*operator src_port*] { **host** *dest_ip* | *dest_ip mask* | **any4** | **any6** } [*operator dest_port*]]

Ingress switch capture packets for Secure Firewall 3100 model devices:

capture *capture_name* **switch interface** *interface_name*

Switch capture packets for Secure Firewall 4200 model devices:

capture *capture_name* **switch interface** *interface_name* [**direction** { **both** | **egress** | **ingress** }]

Capture packets cluster-wide:

cluster exec capture *capture_name* [**persist**] [**include-decrypted**]

Remove the packet capture or a parameter from the capture. Omit parameters if the intention is to remove the capture entirely.

no capture *capture_name* [*arguments*]

Stop the packet capture without removing it:

capture *capture_name* **stop**

Syntax Description

any4	Specifies any IPv4 address instead of a single IP address and mask.
any6	Specifies any IPv6 address instead of a single IP address and mask.
all	Captures all packets dropped by the accelerated security path.

asp-drop *drop-code*	(Optional) Captures packets dropped by the accelerated security path. The drop-code specifies the type of traffic that is dropped by the accelerated security path. See the CLI help for a list of drop codes. You can enter this keyword with the **packet-length**, **circular-buffer**, and **buffer** keywords, but not with the **interface** or **ethernet-type** keyword. In a cluster, dropped forwarded data packets from one unit to another are also captured.
buffer *buf_size*	(Optional) Defines the buffer size used to store the packet in bytes. Once the byte buffer is full, packet capture stops. When used in a cluster, this is the per-unit size, not the sum of all units. The maximum buffer size supported is 32 MB. The buffer size and file size options are mutually exclusive.
capture_name	Specifies the name of the packet capture. Use the same name on multiple **capture** statements to capture multiple types of traffic. When you view the capture configuration using the **show capture** command, all options are combined on one line.
data-plane	Specifies the captured packets on the dataplane interface.
direction	(Optional. Supported only on Secure Firewall 4200 model devices.) Specifies the direction of the switch traffic to be captured. It can be one of the following: • **both**—To capture switch bi-directional traffic • **egress**—To capture switch egressing traffic • **ingress**—To capture switch ingressing traffic
management-plane	Specifies the captured packets on the management interface.
circular-buffer	(Optional) Overwrites the buffer, starting from the beginning, when the buffer is full.
ethernet-type *type*	(Optional) Selects an Ethernet type to capture. Supported Ethernet types include 8021Q, ARP, IP, IP6, IPX, LACP, PPPOED, PPPOES, RARP, and VLAN. An exception occurs with the 802.1Q or VLAN type. The 802.1Q tag is automatically skipped and the inner Ethernet type is used for matching.
file-size *file-size*	(Optional) **file-size** specifies capturing packets to a file on disk . *file-size* specifies the size of capture file from 32 MB to 10 GB. The capture file will be created in flash memory **(disk0:/)** with the name **capture_name.pcap** . When the **file-size** is configured, the hard disk memory (file) is used to write the captured data in the capture buffer. The captured data gets stored in the disk based on the filename. The buffer size and file size options are mutually exclusive.
headers-only	(Optional) Selects Layer 2 and Layer 3/4 headers of packet to capture without data.
host *source_ip*, *dest_ip*	Specifies the single IP address of the host to or from which the packet is being sent.

include-decrypted	(Optional) Captures decrypted IPsec packets which contain both normal and decrypted traffic once they enter the firewall device. It also captures packets of SSL decrypted traffic. However, this option is not applicable for VTI tunnel as packets are seen in decrypted format only on the VTI interface; not on the outside like for crypto map VPN.
inline-tag *tag*	Specifies a tag for a particular SGT value or leaves it unspecified to capture a tagged packet with any SGT value.
interface *interface_name*	Sets the name of the interface on which to use packet capture. You must configure an interface for any packets to be captured except for **type asp-drop**. You can configure multiple interfaces using multiple **capture** commands with the same name. To capture packets on the management plane, you can use the **interface** keyword with **asa_mgmt_plane** as the interface name.You can specify **cluster** as the interface name to capture the traffic on the cluster control link interface. To capture packets on the internal backplane interface when you enable the management center access on a data interface, specify **nlp_int_tap**. If the type **lacp** capture is configured, the interface name is the physical name.
ikev1, **ikev2**	Captures only IKEv1 or IKEv2 protocol information.
isakmp	(Optional) Captures ISAKMP traffic for VPN connections. The ISAKMP subsystem does not have access to the upper layer protocols. The capture is a pseudo capture, with the physical, IP, and UDP layers combined together to satisfy a PCAP parser. The peer addresses are obtained from the SA exchange and are stored in the IP layer.
lacp	(Optional) Captures LACP traffic. If configured, the interface name is the physical interface name.
mask	The subnet mask for the IP address, for example, 255.255.255.0 for a Class C mask.
match *protocol*	Specifies the packets that match the five-tuple to allow filtering of those packets to be captured. You can use this keyword up to three times on one line.
operator src_port, *dest_port*	(Optional) Matches the port numbers used by the source or destination. The permitted operators are as follows: • **lt**—less than • **gt**—greater than • **eq**—equal to • **neq**—not equal to • **range**—range
packet-length *bytes*	(Optional) Sets the maximum number of bytes of each packet to store in the capture buffer.
persit	(Optional) Captures persistent packets on cluter units.
raw-data	(Optional) Captures inbound and outbound packets on one or more interfaces.

stop	Stop the packet capture without removing it. Use the **no** form of the command with this option to restart the capture.
trace *trace_count*	(Optional) Captures packet trace information, and the number of packets to capture. This option is used with an access list to insert trace packets into the data path to determine whether or not the packet has been processed as expected.
type	(Optional) Specifies the type of data captured.

Command Default

The defaults are as follows:

- The default **type** is **raw-data**.
- The default **buffer** *size* is 512 KB.
- The default Ethernet type is IP packets.
- The default **packet-length** is 1518 bytes.
- The default **direction** is ingress.

Command History

Release	Modification
6.1	This command was introduced.
6.2.1	This command was updated to store the contents of all the active captures to files on flash or disks at the time of box crash.
6.2.3	The options asa_mgmt_plane and asa_dataplane were renamed to **management-plane** and **data-plane** respectively.
6.2.3.x	The options **any4** and **any6** were introduced to capture IPv4 and IPv6 network traffic respectively.
6.3	The option [**file-size** *file-size*] allows you to capture file size in MB (32-10000).
6.7	The **interface nlp_int_tap** interface name was added to capture packets on the internal backplane interface when you enable the management center access on a data interface.
7.4	The **direction** keyword was added to capture switch traffic that flows in **egress**, **ingress**, or **both** directions. This keyword is applicable only for Secure Firewall 4200 model devices.

Usage Guidelines

Capturing packets is useful when troubleshooting connectivity problems or monitoring suspicious activity. You can create multiple captures. The **capture** command is not saved to the running configuration, and is not copied to the standby unit during high availability.

The threat defense device is capable of tracking all IP traffic that flows across it and of capturing all the IP traffic that is destined to it, including all the management traffic (such as SSH and Telnet traffic).

The threat defense architecture consists of three different sets of processors for packet processing; this architecture poses certain restrictions on the capability of the capture feature. Typically most of the packet forwarding functionality in the threat defense device is handled by the two front-end network processors, and packets are sent to the control-plane general-purpose processor only if they need application inspection. The

packets are sent to the session management path network processor only if there is a session miss in the accelerated path processor.

Because all the packets that are forwarded or dropped by the threat defense device hits the two front-end network processors, the packet capture feature is implemented in these network processors. So all the packets that hit the threat defense device can be captured by these front end processors, if an appropriate capture is configured for those traffic interfaces. On the ingress side, the packets are captured the moment the packet hits the interfaces, and on the egress side the packets are captured just before they are sent out on the wire.

To save the captured data, packet capture automatically writes captured data to the physical storage on the fly, without having to use the **copy** command. The capture size is supported up to 10 GB. Captures larger than 100 MB are automatically compressed.

Save the Capture

The contents of any active capture on threat defense device are saved when the box crashes. When you activate captures as part of the troubleshooting process, you must note the following points:

- The size of capture buffer to use and if there is enough space on flash/disk.
- The capture buffer should be marked as circular for all the use cases, so that captured packets are the most recent before crash.

The name of the file for saving contents of an active capture is in the format of:

[<context_name>.]<capture_name>.pcap

The *context_name* indicates the name of the user context in which capture is activated in the multi-context mode. For the single context mode, the context_name is not applicable.

The *capture_name* indicates the name of the capture that is activated.

The capture save happens before the console or crash dump. This increases the crash downtime by about 5 seconds for a 33 MB capture buffer. The risk of a nested crash is minimal because copying the captured contents to a file is a simple process.

View the Capture

To view the packet capture, use the **show capture** *name* command. To save the capture to a file, use the **copy capture** command. Use the **https://***FTP-ip-address*/*admin*/**capture/***capture_name*[/**pcap**] command to see the packet capture information with a web browser. If you specify the **pcap** keyword, then a libpcap-format file is downloaded to the web browser and can be saved using the web browser. (A libcap file can be viewed with TCPDUMP or Ethereal.)

If you copy the buffer contents to a TFTP server in ASCII format, you will see only the headers, not the details and hexadecimal dump of the packets. To see the details and hexadecimal dump, you need to transfer the buffer in PCAP format and read it with TCPDUMP or Ethereal.

Delete the Capture

Entering **no capture** without any keywords deletes the capture. To preserve the capture, specify the **interface** keyword; the capture is detached from the specified interface, and the capture is preserved.

Clustering

You can precede the **capture** command with **cluster exec** to issue the **capture** command on one unit and run the command in all the other units at the same time. After you have performed cluster-wide capture, to copy

the same capture file from all units in the cluster at the same time to a TFTP server, enter the **cluster exec copy** command on the master unit.

cluster exec capture *capture_name arguments*

cluster exec copy /pcap capture: *cap_name* **tftp://***location/path/filename***.pcap**

Multiple PCAP files, one from each unit, are copied to the TFTP server. The destination capture file name is automatically attached with the unit name, such as filename_A.pcap, filename_B.pcap, and so on. In this example, A and B are cluster unit names.

Note A different destination name is generated if you add the unit name at the end of the filename.

Limitations

The following are some of the limitations of the capture feature. Most of the limitations are caused by the distributed nature of the threat defense architecture and by the hardware accelerators that are being used in the threat defense device.

- For inline SGT tagged packets, captured packets contain an additional CMD header that your PCAP viewer might not understand.
- If the 802.1Q tag in the packets is different than that of the configured sub-interfaces, such packets are not captured. The packets are ignored because they are not associated with any named interface.
- If there is no ingress interface and therefore no global interface, packets sent on the backplane are treated as control packets. These packets bypass the access list check and are always captured.
- The show capture command shows the correct reason when capturing a specific asp-drop. However, the show capture command does not show the correct reason when capturing all asp-drops.

The packet capture feature with the file-size option has the following limitations:

- Applicable only for Firepower 4100/9300 series.
- For existing capture, you cannot add the file size option.
- The **copy** command is not supported.
- Real-time, trace, linear, and circular buffer are not supported.
- If the number of captures with the file size option is increased, the performance of the system will be reduced.
- If the system load is high, it leads to packet capture data loss.

Examples

To capture a packet, enter the following command:

```
> capture captest interface inside
> capture captest interface outside
```

On a web browser, you can view the content of the **capture** command that was issued, named "captest," at the following location:

```
https://171.69.38.95/admin/capture/captest
```

To download a libpcap file (that web browsers use) to a local machine, enter the following command:

```
https://171.69.38.95/capture/http/pcap
```

The following example shows how to capture a packet in the single-mode when the threat defense Device crashes:

```
> capture 789 interface inside
```

The contents of capture '789' is saved as *789.pcap* file.

The following example shows how to capture a packet in the multi-mode when the threat defense crashes:

```
>capture 624 interface inside
```

The contents of capture '624' in admin context is saved as *admin.624.pcap* file.

The following example shows how to capture ARP packets:

```
> capture arp ethernet-type arp interface outside
```

Capture for Clustering

To enable capture on all units in the cluster, you can add the cluster exec keywords in front of each of these commands.

The following example shows how to create an LACP capture for the clustering environment:

```
> capture lacp type lacp interface gigabitEthernet0/0
```

The following example shows how to create a capture for control path packets in the clustering link:

```
> capture cp interface cluster match udp any eq 49495 any
> capture cp interface cluster match udp any any eq 49495
```

The following example shows how to capture data path traffic through the cluster:

```
> capture abc interface inside match tcp host 1.1.1.1 host 2.2.2.2 eq www
> capture abc interface inside match dup host 1.1.1.1 any
```

Capture for Switch

The following example shows how to create and start an egress traffic capture for a switch:

```
> capture switchegress_cap switch interface gigabitEthernet0/0 direction egress
> no capture switchegress_cap switch stop
```

Related Commands

Command	Description
clear capture	Clears the capture buffer.
copy capture	Copies a capture file to a server.
show capture	Displays the capture configuration when no options are specified.

capture-traffic

To intercept and capture packets passing through the threat defense interface, use the **capture-traffic** command. You can capture traffic on a specified threat defense domain that matches the integer expression from the list of options presented, either the management interface (br1) or traffic interfaces.

capture-traffic

You are prompted for the domain and TCP dump options.

Syntax Description

domain	Specifies the domain where traffic is captured: • 0—br1, captures traffic from the management interface • 1—Router, captures traffic from the configured data interfaces
-A	Prints each packet (minus its link level header) in ASCII. Handy for capturing web pages.
-B	Sets the operating system capture buffer size to buffer_size.
-c	Exits after receiving count packets.
-C	Before writing a raw packet to a savefile, checks whether the file is currently larger than file_size and, if so, close the current savefile and open a new one. Savefiles after the first savefile will have the name specified with the -w flag, with a number after it, starting at 1 and continuing upward. The units of file_size are millions of bytes (1,000,000 bytes, not 1,048,576 bytes).
-d	Dumps the compiled packet-matching code in a human readable form to standard output and stop.
-dd	Dumps packet-matching code as a C program fragment.
-ddd	Dumps packet-matching code as decimal numbers (preceded with a count).
-D	Prints the list of the network interfaces available on the system and on which tcpdump can capture packets. For each network interface, a number and an interface name, possibly followed by a text description of the interface, is printed. The interface name or the number can be supplied to the -i flag to specify an interface on which to capture. This can be useful on systems that do not have a command to list them (Windows systems, or UNIX systems lacking ifconfig -a); the number can be useful on Windows 2000 and later systems, where the interface name is a somewhat complex string. The -D flag will not be supported if tcpdump was built with an older version of libpcap that lacks the pcap_findaclldevs() function.
-e	Prints the link-level header on each dump line.
-E	Uses spi@ipaddr algo:secret for decrypting IPsec ESP packets that are addressed to addr and contain Security Parameter Index value spi. This combination may be repeated with comma or newline separation.

-f	Prints ‘foreign’ IPv4 addresses numerically rather than symbolically (this option is intended to get around serious brain damage in Sun's NIS server usually it hangs forever translating non-local internet numbers). The test for ‘foreign’ IPv4 addresses is done using the IPv4 address and netmask of the interface on which capture is being done. If that address or netmask are not available, available, either because the interface on which capture is being done has no address or netmask or because the capture is being done on the Linux 'any' interface, which can capture on more than one interface, this option will not work correctly.
-F	Uses file as input for the filter expression. An additional expression given on the command line is ignored.
-G	If specified, rotates the dump file specified with the -w option every rotate_seconds seconds. Savefiles will have the name specified by -w which should include a time format as defined by strftime(3). If no time format is specified, each new file will overwrite the previous. If used in conjunction with the -C option, filenames will take the form of ‘file<count>’.
-I	Puts the interface in ‘monitor mode’; this is supported only on IEEE 802.11 Wi-Fi interfaces, and supported only on some operating systems.
-K	Does not attempt to verify TCP checksums. This is useful for interfaces that perform the TCP checksum calculation in hardware; otherwise, all outgoing TCP checksums will be flagged as bad.
-l	Makes stdout line buffered. Useful if you want to see the data while capturing it. Example, “tcpdump -l | tee dat” or “tcpdump -l > dat & tail -f dat”.
-L	Lists the known data link types for the interface and exit.
-m	Loads SMI MIB module definitions from file module. This option can be used several times to load several MIB modules into tcpdump.
-M	Uses secret as a shared secret for validating the digests found in TCP segments with the TCP-MD5 option (RFC 2385), if present.
-n	Does not convert addresses (i.e., host addresses, port numbers, etc.) to names.
-N	Does not print domain name qualification of host names. Example, if you give this flag then tcpdump will print “nic” instead of “nic.ddn.mil”.
-O	Does not run the packet-matching code optimizer. This is useful only if you suspect a bug in the optimizer.
-p	Does not put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, `-p' cannot be used as an abbreviation for `ether host {local-hw-addr} or ether broadcast'.

-q	Quick output. Prints less protocol information so output lines are shorter.
-R	Assumes ESP/AH packets to be based on old specification (RFC1825 to RFC1829). If specified, tcpdump will not print replay prevention field. Because there is no protocol version field in ESP/AH specification, tcpdump cannot deduce the version of ESP/AH protocol.
-r	Reads packets from file (which was created with the -w option). Standard input is used if file is "-".
-S	Prints absolute, rather than relative, TCP sequence numbers.
-s	Snarfs snaplen bytes of data from each packet rather than the default of 68 (with SunOS's NIT, the minimum is actually 96). 68 bytes is adequate for IP, ICMP, TCP, and UDP but may truncate protocol information from name server and NFS packets (see below). Packets truncated because of a limited snapshot are indicated in the output with "[|proto]", where proto is the name of the protocol level at which the truncation has occurred. Note that taking larger snapshots both increases the amount of time it takes to process packets and, effectively, decreases the amount of packet buffering. This may cause packets to be lost. You should limit snaplen to the smallest number that will capture the protocol information you're interested in. Setting snaplen to 0 means use the required length to catch whole packets.
-T	Forces packets selected by 'expression' to be interpreted the specified type. Currently known types are aodv (Ad-hoc On-demand Distance Vector protocol), cnfp (Cisco NetFlow protocol), rpc (Remote Procedure Call), rtp (Real-Time Applications protocol), rtcp (Real-Time Applications control protocol), snmp (Simple Network Management Protocol), tftp (Trivial File Transfer Protocol), vat (Visual Audio Tool), and wb (distributed White Board).
-t	Does not print a timestamp on each dump line.
-tt	Prints an unformatted timestamp on each dump line.
-ttt	Prints a delta (micro-second resolution) between current and previous line on each dump line.
-tttt	Prints a timestamp in default format proceeded by date on each dump line.
-ttttt	Prints a delta (micro-second resolution) between current and first line on each dump line.
-u	Prints undecoded NFS handles.
-U	Makes output saved via the -w option "packet-buffered"; i.e., as each packet is saved, it will be written to the output file, rather than being written only when the output buffer fills. The -U flag will not be supported if tcpdump was built with an older version of libpcap that lacks the pcap_dump_flush() function.

-v	When parsing and printing, produces (slightly more) verbose output. For example, the time to live, identification, total length and options in an IP packet are printed. Also enables additional packet integrity checks such as verifying the IP and ICMP header checksum. When writing to a file with the -w option, report, every 10 seconds, the number of packets captured.
-vv	Even more verbose output. For example, additional fields are printed from NFS reply packets, and SMB packets are fully decoded.
-vvv	Even more verbose output. For example, telnet SB... SE options are printed in full. With -X Telnet options are printed in hex as well.
-w	Write the raw packets to file rather than parsing and printing them out. They can later be printed with the -r option. Standard output is used if file is "-".
-W	Used in conjunction with the -C option, this will limit the number of files created to the specified number, and begin over writing files from the beginning, thus creating a 'rotating' buffer. In addition, it will name the files with enough leading 0s to support the maximum number of files, allowing them to sort correctly.
-x	When parsing and printing, in addition to printing the headers of each packet, prints the data of each packet (minus its link level header) in hex. The smaller of the entire packet or snaplen bytes will be printed. Note that this is the entire link-layer packet, so for link layers that pad (e.g. Ethernet), the padding bytes will also be printed when the higher layer packet is shorter than the required padding.
-xx	When parsing and printing, in addition to printing the headers of each packet, prints the data of each packet, including its link level header, in hex.
-X	When parsing and printing, in addition to printing the headers of each packet, print the data of each packet (minus its link level header) in hex and ASCII. This is very handy for analyzing new protocols.
-XX	When parsing and printing, in addition to printing the headers of each packet, prints the data of each packet, including its link level header, in hex and ASCII.
-y	Sets the data link type to use while capturing packets to datalinktype.
-Z	Drops privileges (if root) and changes user ID to user and the group ID to the primary group of user.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

By default the **capture-traffic** command produces one line of text per every packet it intercepts. Each line includes: a time stamp; the protocol name; the source and destination addresses (for IP packets, these are IP addresses; for other protocols, **capture-traffic** does not print any identifiers unless explicitly asked to do so (see the **-e** command line description)); and information including TCP sequence numbers, flags, ARP/ICMP commands, and so on.

Note The **pcap** file (output of the **capture-traffic** or **debug daq**command) displays untranslated details of the packet that was received; the **Connection Events** list (management center) displays translated packet details that are actually applied with the policies.

To stop the capture, type Control + C. If you use **-w** *outputfile* option, the packet capture will be saved with that file name in /var/common/. Otherwise it is written to the display.

Examples

The following example shows how to capture traffic from the management interface:

```
> capture-traffic
Please choose domain to capture traffic from:
  0 - br1
  1 - Router
Selection? 0
Please specify tcpdump options desired.
(or enter '?' for a list of supported options)
-v
```

Related Commands

Command	Description
show traffic	Displays traffic statistics.
show interface	Displays interface status information.

clear aaa-server statistics

To reset statistics for AAA servers, use the **clear aaa-server statistics** command.

clear aaa-server statistics [**LOCAL** | *groupname* [**host** *hostname*] | **protocol** *protocol*]

Syntax Description

groupname	(Optional) Clears statistics for servers in a group.
host *hostname*	(Optional) Clears statistics for a particular server in the group.
LOCAL	(Optional) Clears statistics for the LOCAL user database.
protocol *protocol*	(Optional) Clears statistics for servers of the specified protocol. Enter ? to see the available protocols.

Command Default

Removes all AAA server statistics across all groups.

Command History

Release	Modification
6.2.1	This command was introduced.

Examples

The following example shows how to reset the AAA statistics for all server groups:

```
> clear aaa-server statistics
```

The following example shows how to reset the AAA statistics for an entire server group:

```
> clear aaa-server statistics svrgrp1
```

The following example shows how to reset the AAA statistics for a specific server in a group:

```
> clear aaa-server statistics svrgrp1 host 10.2.3.4
```

Related Commands

Commands	Description
show aaa-server	Displays AAA server statistics

clear access-list

To clear an access-list counter, use the clear access-list command.

clear access-list *id*

Syntax Description

id	Name of an access list.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

When you enter the **clear access-list** command, you must specify the *id* of an access list to clear the counters. Use the **show access-list** command for a list of ACLs.

Examples

The following example shows how to clear a specific access list counter:

```
> clear access-list inbound
```

Related Commands

Command	Description
show access-list	Displays the access list entries by number.
show running-config access-list	Displays the access list configuration that is running on the adaptive security appliance.

clear arp

To clear dynamic ARP entries or ARP statistics, use the **clear arp** command.

clear arp [**statistics** | *interface_name*]

Syntax Description

statistics	Clears ARP statistics.
interface_name	Clears statistics for the specified interface.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example clears all ARP statistics:

```
> clear arp statistics
```

Related Commands

Command	Description
show arp statistics	Shows ARP statistics.
show running-config arp	Shows the current configuration of the ARP timeout.

clear asp

To clear accelerated security path (ASP) statistics, use the **clear asp** command.

clear asp { **cluster counter** | **dispatch** | **drop** [**flow** | **frame**] | **event dp-cp** | **inspect-dp ack-passthrough** | **inspect-dp egress-optimization** | **inspect-dp snort** { **counters** [**instance** *number* [**queue** *number*]] | **queue-exhaustion** [**snapshot** *number*] } | **load-balance history** | **overhead** | **packet-profile** | **table** [**arp** | **classify** | **filter** [**access-list** *acl_name*]] }

Syntax Description

access-list *acl_name*	Clears the hit counters only for a specified access list.
arp	Clears the hits counters in ASP ARP tables only.
classify	Clears the hits counters in ASP classify tables only
cluster counter	Clears cluster counters.
counters	Clears the data-path inspection Snort counters.
dispatch	Clears dispatch statistics.
event	Clears data-path to control-plane event statistics.
filter	Clears the hits counters in ASP filter tables only
flow	Clears the dropped flow statistics.
frame	Clears the dropped frame/packet statistics.
inspect-dp ack-passthrough	Clears counters for empty TCP ACK packets that bypassed Snort inspection.
inspect-dp egress-optimization	Clears egress optimization statistics.
inspect-dp snort	Clears data-path inspection Snort statistics.
instance *number*	Clears the counters by instance ID.
load-balance history	Clears the history of ASP load balancing per packet and reset the number of times an automatic switch occurred
overhead	Clears all ASP multiprocessor overhead statistics.
packet-profile	Clears packet profile statistics.
queue *number*	Clears the counters by instance ID and queue ID.
queue-exhaustion	Clears the data-path inspection Snort queue snapshot.
snapshot *number*	Clears the queue exhaustion by snapshot ID.

table	Clears the hit counters in ASP ARP tables and ASP classify tables.

Command History

Release	Modification
6.1	This command was introduced.
6.4	The **clear asp inspect-dp egress-optimization** command was introduced.
6.5	The **packet-profile** keyword was added.
7.0	The **inspect-dp ack-passthrough** keyword was added.

Examples

The following example clears all dispatch statistics:

```
> clear asp dispatch
```

Related Commands

Command	Description
show asp	Shows ASP statistics.

clear bfd

To clear all bi-directional forwarding detection (BFD) counters, use the **clear bfd counters** command.

clear bfd counters [**ld** *local_discr* | *interface_name* | **ipv4** *ip_address* | **ipv6** *ip_address*]

Syntax Description

ld *local_discr*	(Optional) Clears BFD counters for the specified local discriminator, 1 - 4294967295.
interface_name	(Optional) Clears BFD counters for the specified interface.
ipv4 *ip_address*	(Optional) Clears BFD counters for the specified neighbor IPv4 address.
ipv6 *ip_address*	(Optional) Clears BFD counters for the specified neighbor IPv6 address.

Command History

Release	Modification
6.3	This command was introduced.

Examples

The following example clears all BFD counters:

```
> clear bfd counters
```

Related Commands

Command	Description
show bfd	Shows BFD protocol information, including packets dropped, neighbors, and map entries.

clear bgp

To reset Border Gateway Protocol (BGP) connections using hard or soft reconfiguration, use the **clear bgp** command.

clear bgp { [***** | **external**] [**ipv4 unicast** [*as_number* | *neighbor_address* | **table-map**] | **ipv6 unicast** [*as_number* | *neighbor_address*]] [**soft**] [**in** | **out**] | *as_number* [**soft**] [**in** | **out**] | *neighbor_address* [**soft**] [**in** | **out**] | **table-map**}

Syntax Description

*****	Specifies that all current BGP sessions will be reset.
as_number	(Optional) Number of the autonomous system in which all BGP peer sessions will be reset.
external	Specifies that all external BGP sessions will be reset.
in	(Optional) Initiates inbound reconfiguration. If neither the **in** nor **out** keywords are specified, both inbound and outbound sessions are reset.
ipv4 unicast	Resets BGP connections using hard or soft econfiguration for IPv4 address family sessions.
ipv6 unicast	Resets BGP connections using hard or soft econfiguration for IPv6 address family sessions.
neighbor_address	(Optional) Specifies that only the identified BGP neighbor will be reset. The value for this argument can be an IPv4 or IPv6 address.
out	(Optional) Initiates inbound or outbound reconfiguration. If neither the **in** nor **out** keywords are specified, both inbound and outbound sessions are reset.
soft	(Optional) Clears slow-peer status forcefully, and moves it to original update group.
table-map	Clears table-map configuration information in BGP routing tables. This command can be used to clear traffic-index information configured with the BGP Policy Accounting feature.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **clear bgp** command can be used to initiate a hard reset or soft reconfiguration. A hard reset tears down and rebuilds the specified peering sessions and rebuilds the BGP routing tables. A soft reconfiguration uses stored prefix information to reconfigure and activate BGP routing tables without tearing down existing peering sessions. Soft reconfiguration uses stored update information, at the cost of additional memory for storing the updates, to allow you to apply a new BGP policy without disrupting the network. Soft reconfiguration can be configured for inbound or outbound sessions.

Examples

In the following example, all the BGP sessions are reset:

```
> clear bgp *
```

In the following example, a soft reconfiguration is initiated for the inbound session with the neighbor 10.100.0.1, and the outbound session is unaffected:

```
> clear bgp 10.100.0.1 soft in
```

In the following example, the route refresh capability is enabled on the BGP neighbor routers, a soft reconfiguration is initiated for the inbound session with the neighbor 172.16.10.2, and the outbound session is unaffected:

```
> clear bgp 172.16.10.2 in
```

In the following example, a hard reset is initiated for sessions with all routers in the autonomous system numbered 35700:

```
> clear bgp 35700
```

In the following example, a soft reconfiguration is configured for all inbound eBGP peering sessions:

```
> clear bgp external soft in
```

In the following example, all outbound address family IPv4 multicast eBGP peering sessions are cleared:

```
> clear bgp external ipv4 multicast out
```

In the following example, a soft reconfiguration is initiated for the inbound sessions for BGP neighbors in IPv4 unicast address family sessions in autonomous system 65400, and the outbound session is unaffected:

```
> clear bgp ipv4 unicast 65400 soft in
```

In the following example, a hard reset is initiated for BGP neighbors in IPv4 unicast address family sessions in the 4-byte autonomous system numbered 65538 in asplain notation:

```
> clear bgp ipv4 unicast 65538
```

In the following example, a hard reset is initiated for BGP neighbors in IPv4 unicast address family sessions in the 4-byte autonomous system numbered 1.2 in asdot notation:

```
> clear bgp ipv4 unicast 1.2
```

The following example clears the table map for IPv4 unicast peering sessions:

```
> clear bgp ipv4 unicast table-map
```

clear blocks

To reset the packet buffer counters such as the exhaustion condition and history information, use the **clear blocks** command.

clear blocks [**exhaustion** {**history** | **snapshot**} | **export-failed** | **queue** [**history** [**core-local** [*number*]]]]

Syntax Description

core-local [*number*]	(Optional) Clears system buffers queued by application for all cores, or if you specify the core number, a specific core.
exhaustion	(Optional) Clears the exhaustion condition.
export-failed	(Optional) Clears the export failed counters.
history	(Optional) Clears the history.
queue	(Optional) Clears queued blocks.
snapshot	(Optional) Clears the snapshot information.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

Resets the low watermark counters to the current available blocks in each pool. Additionally, this command clears the history information stored during the last buffer allocation failure.

Examples

The following example clears the blocks:

```
> clear blocks
```

Related Commands

Command	Description
blocks	Increases the memory assigned to block diagnostics.
show blocks	Shows the system buffer utilization.

clear capture

To clear the capture buffer, use the **clear capture** command.

clear capture {**/all** | *capture_name*}

Syntax Description

/all	Clears packets on all interfaces.
capture_name	Specifies the name of the packet capture.

Command History

Release	Modification
6.1	This command was introduced.

Examples

This example shows how to clear the capture buffer for the capture buffer “example.”

```
> clear capture example
```

Related Commands

Command	Description
capture	Enables packet capture capabilities for packet sniffing and network fault isolation.
show capture	Displays the capture configuration when no options are specified.

clear clns

To clear Connectionless-mode Network Protocol (CLNP) information, use the **clear clns** command.

clear clns {**is-neighbors** | **neighbors** | **traffic**}

Syntax Description

is-neighbors	Clears intermediate-system neighbor routes.
neighbors	Clears all CLNS neighbor routes.
traffic	Clears CLNS protocol statistics.

Command History

Release	Modification
6.3	This command was introduced.

Examples

This example shows how to clear all CLNS neighbor routes:

```
> clear clns neighbors
```

Related Commands

Command	Description
show clns	Displays Connectionless-mode Network Protocol (CLNP) network information.

clear cluster info

To clear cluster statistics, use the **clear cluster info** command.

clear cluster info {**flow-mobility counters** | **health details** | **trace** | **transport**}

Syntax Description

flow-mobility counters	Clears the cluster flow-mobility counters.
health details	Clears cluster health information.
trace	Clears cluster event trace information.
transport	Clears cluster transport statistics.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

To view cluster statistics, use the show cluster info command.

Examples

The following example clears cluster event trace information:

```
> clear cluster info trace
```

Related Commands

Command	Description
show cluster info	Shows cluster statistics.

clear configure key chain

To remove the key chains that are configured., use the **clear configure key chain** command.

clear configure key chain*key-chain-name*

Command History

Release	Modification
6.4	This command was introduced.

Usage Guidelines

Use the **clear configure key chain** command to remove the configured key chain.

Examples

The following example shows how to remove the configured key chain.

```
> clear configure key chain CHAIN1
>
```

Related Commands

Command	Description
key chain	Configure the key chains for ospfv2 authentication.
show key chain	Displays the configured key chains.
show running key chain	Displays the key chain details that is currently active.

clear conn

To clear a specific connection or multiple connections, use the **clear conn** command.

clear conn [**vrf** { *name* | **global** }] { **all** | **protocol** { **tcp** | **udp** | **sctp** } | **address** *ip* [**-** *ip*] [*netmask mask*] | **port** *port* [**-** *port*] | **inline-set** *name* | **security-group** { **name** | **tag** } *attribute* } | **user** [*domain_nickname* \ ] *user_name* | **user-group** [*domain_nickname* \\ ] *user_group_name*] | **zone** [*zone_name*] [**data-rate**] }

Syntax Description

address *ip*[**-***ip*]	Clears connections with the specified source or destination IP address (IPv4 or IPv6). To specify a range, separate the IP addresses with a dash (-). For example: 10.1.1.1-10.1.1.5
all	Clears all connections, including to-the-box connections. Without the all keyword, only through-the-box connections are cleared.
inline-set *name*	Clears connections that match the specified inline set.
netmask *mask*	(Optional) Specifies a subnet mask for use with the given IP address.
port *port*[**-***port*]	Clears connections with the specified source or destination port. To specify a range, separate the port numbers with a dash (-). For example: 1000-2000
protocol {**tcp** | **udp** | **sctp**}	Clears connections with the specified protocol.
security-group {**name** | **tag**} *attribute*	Clears connections with the specified security group attribute.
user [*domain_nickname*\] *user_name*	Clears connections that belong to the specified user. When you do not include the *domain_nickname* argument, the system clears connections for the user in the default domain.
user-group [*domain_nickname*\\] *user_group_name*]	Clears connections that belong to the specified user group. When you do not include the *domain_nickname* argument, the system clears connections for the user group in the default domain.
zone [*zone_name*]	Clears connections that belong to a security zone.
[**vrf** {*name* | **global**}]	If you enable virtual routing and forwarding (VRF), also known as virtual routers, you can limit the command to a specific virtual router using the **vrf** *name* keyword. Specify **vrf global** to limit the command to the global virtual router. If you omit this keyword, the command applies to all virtual routers.
data-rate	(Optional) Clears the current maximum data-rate stored.

Command History

Release	Modification
6.1	This command was introduced.
6.6	The **vrf** and **data-rate** keywords was added.

Usage Guidelines

When you make security policy changes to the configuration, all new connections use the new security policy. Existing connections continue to use the policy that was configured at the time of the connection establishment. To ensure that all connections use the new policy, you need to disconnect the current connections so they can reconnect using the new policy using the **clear conn** command. You can alternatively use the **clear local-host** command to clear connections per host, or the **clear xlate** command for connections that use dynamic NAT.

When the device creates a pinhole to allow secondary connections, this is shown as an incomplete connection in the **show conn** command output. To clear this incomplete connection, use the **clear conn** command.

Note This command does not clear connections to the Management interface; it can only clear management connections to a data interface or the Diagnostic interface.

Examples

The following example shows how to view all connections and then clear the management connection from 10.10.10.108:

```
> show conn all
TCP mgmt 10.10.10.108:4168 NP Identity Ifc 10.0.8.112:22, idle 0:00:00,
bytes 3084, flags UOB
> clear conn address 10.10.10.108
```

The following example shows how to clear connection maximum data-rate stored in the extension memory:

```
> clear conn data-rate
Released conn extension memory for 10 connection(s)
```

Related Commands

Commands	Description
clear local-host	Clears all connections by a specific local host or all local hosts.
clear xlate	Clears a dynamic NAT session, and any connections using NAT.
show conn	Shows connection information.
show local-host	Displays the network states of local hosts.
show xlate	Shows NAT sessions.

clear console-output

To remove the currently captured console output, use the **clear console-output** command.

clear console-output

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to remove the currently captured console output:

```
> clear console-output
```

Related Commands

Command	Description
show console-output	Displays the captured console output.
show running-config console timeout	Displays the idle timeout for a console connection to the device.

clear counters

To clear the protocol stack counters, use the **clear counters** command.

clear counters [**all** | **summary** | **top** *n*] [**detail**] [**protocol** *protocol_name* [*counter_name*]] [**threshold** *n*]

Syntax Description

all	(Optional) Clears all filter details.
counter_name	(Optional) Specifies a counter by name. Use the **show counters protocol** command to see available counter names.
detail	(Optional) Clears detailed counters information.
protocol *protocol_name*	(Optional) Clears the counters for the specified protocol.
summary	(Optional) Clears the counter summary.
threshold *n*	(Optional) Clears the counters at or above the specified threshold. The range is 1 through 4294967295.
top *n*	(Optional) Clears the counters at or above the specified threshold. The range is 1 through 4294967295.

Command Default

The **clear counters summary detail** command is the default.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to clear the protocol stack counters:

```
> clear counters
```

Related Commands

Command	Description
show counters	Displays the protocol stack counters.

clear cpu profile

To clear the CPU profiling statistics, use the **clear cpu** command.

clear cpu profile

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to delete the crash file:

```
> clear cpu profile
```

Related Commands

Command	Description
show cpu	Displays information about the CPU.
show cpu profile	Displays CPU profiling data.

clear crashinfo

To delete the contents of the crash file in flash memory, use the **clear crashinfo** command.

clear crashinfo [**module** {**0** | **1**}]

Syntax Description

module {**0** | **1**}	(Optional) Clears the crash file for a module in slot 0 or 1.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to delete the crash file:

```
> clear crashinfo
```

Related Commands

Command	Description
crashinfo force	Forces a crash of the system.
crashinfo test	Tests the ability of the system to save crash information to a file in flash memory.
show crashinfo	Displays the contents of the crash file stored in flash memory.

clear crypto accelerator statistics

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

clear crypto accelerator statistics

Command History

Release	Modification
6.1	This command was introduced.

Examples

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

```
> clear crypto accelerator statistics
>
```

Related Commands

Command	Description
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 in the crypto accelerator MIB.
show crypto protocol statistics	Displays the protocol-specific statistics from the crypto accelerator MIB.

clear crypto ca crls

To empty the CRL cache of all CRLs associated with a specified trustpoint, all CRLs associated with the trustpool from the cache, or the CRL cache of all CRLs, use the **clear crypto ca crls** command.

clear crypto ca crls [**trustpool** | **trustpoint** *trust_point_name*]

Syntax Description

trustpoint *trust_point_name*	The name of a trustpoint. If you do not specify a name, this command clears all CRLs cached on the system. If you give the trustpoint keyword without a trustpointname, the command fails.
trustpool	Indicates that the action should be applied only to the CRLs that are associated with certificates in the trustpool.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following independent examples clear all of the trustpool CRLs, clears all of the CRLs associated with trustpoint123, and removes all of the cached CRLs from the device:

```
> clear crypto ca crl trustpool
> clear crypto ca crl trustpoint trustpoint123
> clear crypto ca crl
```

Related Commands

Command	Description
show crypto ca crl	Displays all cached CRLs or CRLs cached for a specified trustpoint.

clear crypto ca trustpool

To remove all certificates from the trustpool, use the **clear crypto ca trustpool** command.

clear crypto ca trustpool noconfirm

Syntax Description

noconfirm	Suppresses user confirmation prompts, and the command will be processed as requested.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example clears all certificates:

```
> clear crypto ca trustpool
>
```

Related Commands

Command	Description
crypto ca trustpool export	Exports the certificates that constitute the PKI trustpool.
crypto ca trustpool import	Imports the certificates that constitute the PKI trustpool.
crypto ca trustpool remove	Removes a single specified certificate from the trustpool.

clear crypto ikev1

To remove the IPsec IKEv1 SAs or statistics, use the **clear crypto ikev1** command.

clear crypto ikev1 {**sa** [*ip_address*] | **stats**}

Syntax Description

sa *ip_address*	Clears the SA. To clear all IKEv1 SAs, use this option without specifying an IP address. Otherwise, specify the IPv4 or IPv6 address of the SA to clear.
stats	Clears the IKEv1 statistics.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example removes all of the IPsec IKEv1 statistics from the threat defense device:

```
> clear crypto ikev1 stats
>
```

The following example deletes SAs with a peer IP address of 10.86.1.1:

```
> clear crypto ikev1 sa 10.86.1.1
>
```

Related Commands

Command	Description
show ipsec sa	Displays information about IPSec SAs, including counters, entry, map name, peer IP address and hostname.
show running-config crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.

clear crypto ikev2

To remove the IPsec IKEv2 SAs or statistics, use the **clear crypto ikev2** command.

clear crypto ikev2 {**sa** [*ip_address*] | **stats**}

Syntax Description

sa *ip_address*	Clears the SA. To clear all IKEv2 SAs, use this option without specifying an IP address. Otherwise, specify the IPv4 or IPv6 address of the SA to clear.
stats	Clears the IKEv2 statistics.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example removes all of the IPsec IKEv2 statistics from the threat defense device:

```
> clear crypto ikev2 stats
>
```

The following example deletes SAs with a peer IP address of 10.86.1.1:

```
> clear crypto ikev2 sa 10.86.1.1
>
```

Related Commands

Command	Description
show ipsec sa	Displays information about IPSec SAs, including counters, entry, map name, peer IP address and hostname.
show running-config crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.

clear crypto ipsec sa

To remove the IPsec SA counters, entries, crypto maps or peer connections, use the **clear crypto ipsec sa** command.

clear crypto ipsec sa [**counters** | **entry** *ip_address* {**esp** | **ah**} *spi* | **inactive** | **map** *map_name* | **peer** *ip_address*]

Syntax Description

ah	Authentication header.
counters	Clears all IPsec per SA statistics.
entry *ip_address*	Deletes the tunnel that matches the specified IP address/hostname, and protocol, and SPI value.
esp	Encryption security protocol.
inactive	Clears all inactive IPsec SAs.
map *map_name*	Deletes all tunnels associated with the specified crypto map as identified by map name.
peer *ip_address*	Deletes all IPsec SAs to a peer as identified by the specified hostname or IP address.
spi	Identifies the Security Parameters Index (a hexidecimal number). This must be the inbound SPI. We do not support this command for the outbound SPI.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

To clear all IPsec SAs, use this command without arguments.

Examples

The following example removes all of the IPsec SAs from the threat defense:

```
> clear crypto ipsec sa
>
```

The following example deletes SAs with a peer IP address of 10.86.1.1:

```
> clear crypto ipsec sa peer 10.86.1.1
```

Related Commands

Command	Description
show ipsec sa	Displays information about IPSec SAs, including counters, entry, map name, peer IP address and hostname.
show running-config crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.

clear crypto isakmp

To clear ISAKMP SAs or statistics, use the **clear crypto isakmp** command.

clear crypto isakmp [**sa** | **stats**]

Syntax Description

sa	Clears IKEv1 and IKEv2 SAs.
stats	Clears IKEv1 and IKEv2 statistics.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

To clear all ISAKMP operational data, use this command without arguments.

Examples

The following example removes all of the ISAKMP SAs:

```
> clear crypto isakmp sa
>
```

Related Commands

Command	Description
show isakmp	Displays information about ISAKMP operational data.
show running-config crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.

clear crypto protocol statistics

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

clear crypto protocol statistics *protocol*

Syntax Description

protocol	Specifies the name of the protocol for which you want to clear statistics. Protocol choices are as follows: • **all**—All protocols currently supported. • **ikev1**—Internet Key Exchange (IKE) version 1. • **ikev2**—Internet Key Exchange (IKE) version 2. • **ipsec**—IP Security (IPsec) Phase-2 protocols. • **other**—Reserved for new protocols. • **srtp**—Secure RTP (SRTP) protocol • **ssh**—Secure Shell (SSH) protocol • **ssl**—Secure Socket Layer (SSL) protocol.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example clears all crypto accelerator statistics:

```
> clear crypto protocol statistics all
>
```

Related Commands

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

clear crypto ssl

To clear SSL information, use the **clear crypto ssl** command.

clear crypto ssl {**cache** [**all**] | **errors** | **mib** | **objects**}

Syntax Description

cache	Clears expired sessions in the SSL session cache.
all	(Optional) Clears all sessions and statistics in the SSL session cache.
errors	Clears SSL errors.
mib	Clears SSL MIB statistics.
objects	Clears SSL object statistics.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example clears all SSL cache sessions and statistics:

```
> clear crypto ssl cache all
```

Related Commands

Command	Description
show crypto ssl	Displays the SSL information.

clear dhcpd

To clear the DHCP server bindings and statistics, use the **clear dhcpd** command.

clear dhcpd {**binding** [**all** | *ip_address*] | **statistics**}

Syntax Description

all	(Optional) Clears all dhcpd bindings.
binding	Clears all the client address bindings.
ip_address	(Optional) Clears the binding for the specified IP address.
statistics	Clears statistical information counters.

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to clear the dhcpd statistics:

```
> clear dhcpd statistics
```

Related Commands

Command	Description
show dhcpd	Displays DHCP binding, statistic, or state information.

clear dhcprelay statistics

To clear the DHCP relay statistic counters, use the **clear dhcprelay statistics** command.

clear dhcprelay statistics

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example shows how to clear the DHCP relay statistics:

```
> clear dhcprelay statistics
```

Related Commands

Command	Description
show dhcprelay statistics	Displays DHCP relay agent statistic information.
show running-config dhcprelay	Displays the current DHCP relay agent configuration.

clear dns

To clear IP addresses associated with fully qualified domain name (FQDN) hosts, as resolved through DNS requests, use the **clear dns** command.

clear dns [**host** *fqdn_name*] [**ipcache** [**counters**]]

Syntax Description

host *fqdn_name*	(Optional) Specifies the fully qualified domain name whose IP addresses you want to clear. If you do not specify a host, all DNS resolutions are cleared.
ipcache [**counters**]	Clear all the entries from the IP cache obtained through DNS snooping, which is used in direct internet access policy-based routing. Specify **counters** to simply reset all the hit counts for entries in the cache without deleting them.

Command History

Release	Modification
6.1	This command was introduced.
7.1	The **ipcache** [**counters**] keywords were added.

Examples

The following example clears the IP addresses associated with the specified FQDN host:

```
> clear dns host www.example.com
```

The following example clears the IP cache. After you remove the IP cache, the system repopulates the cache using new DNS queries of the domain names in the network-service objects and object groups. Until the DNS queries are completed, traffic destined to domain names will no longer be classified for the network-services group that contains the domain names of the cleared IP cache entries.

```
> clear dns ip-cache
```

Related Commands

Command	Description
show dns hosts	Shows the DNS resolution for a specific host.

clear dns-hosts cache

To clear the DNS cache, use the **clear dns-hosts cache** command.

clear dns-hosts cache

Command History

Release	Modification
6.1	This command was introduced.

Examples

The following example clears the DNS cache:

```
> clear dns-hosts cache
```

Related Commands

Command	Description
show dns-hosts	Shows the DNS cache.

clear efd-throttle

To clear throttle from throttled elephant flows and bypass Snort inspection, use the **clear efd-throttle** command.

clear efd-throttle { *IPv4_address* | *IPv6_address/prefix* | **all bypass** | **any** { *source_port* { *destination_IPv4_address* | *destination_IPv6_address/prefix* | **any** } | **any** { *destination_IPv4_address* | *destination_IPv6_address/prefix* | **any** { *destination_port* { **tcp bypass** | **udp bypass** } | **any** { **tcp bypass** | **udp bypass** } } } } }

Syntax Description

IPv4_address	Clears the throttled elephant flow for the specified IPv4 address (5-tuple).
IPv6_address/prefix	Clears the throttled elephant flow for the specified IPv6 address.
all	Clears throttle and inspects all elephant flows.
bypass	(Optional) Clears throttle and bypasses Snort inspection for all elephant flows.
any	• Use as an abbreviation for source address and mask of 0.0.0.0 0.0.0.0 and ::/0 • Use for any source port or destination port.
source_port	Clears throttle for connections with the specified source port.
destination_port	Clears throttle for connections with the specified destination port.
tcp	Clears throttle for TCP connections only.
udp	Clears throttle for UDP connections only.

Command History

Release	Modification
7.2	This command was introduced.

Examples

The following example shows how to clear throttling of a throttled elephant flow and continue Snort inspection on that flow:

```
> clear efd-throttle 172.16.77.0 255.255.255.0 1234 172.16.4.0 255.255.255.0 80 tcp
```

The following example shows how to clear throttling of a throttled elephant flow and bypass Snort inspection for that flow:

```
> clear efd-throttle 172.16.77.0 255.255.255.0 1234 172.16.4.0 255.255.255.0 80 tcp bypass
```

The following example shows how to clear throttling of all throttled elephant flows and continue Snort inspection on all the flows:

```
> clear efd-throttle all
```

The following example shows how to clear throttling of all throttled elephant flows and bypass Snort inspection for all the flows:

```
> clear efd-throttle all bypass
```

clear eigrp events

To clear the EIGRP event log, use the **clear eigrp events** command.

clear eigrp [*as_number*] **events**

Syntax Description

as_number	(Optional) Specifies the autonomous system number of the EIGRP process for which you are clearing the event log. Because the device only supports one EIGRP routing process, you do not need to specify the autonomous system number (process ID).

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

You can use the **show eigrp events** command to view the EIGRP event log.

Examples

The following example clears the EIGRP event log:

```
> clear eigrp events
```

Related Commands

Command	Description
show eigrp events	Displays the EIGRP event log.

clear eigrp neighbors

To delete entries from the EIGRP neighbor table, use the **clear eigrp neighbors** command.

clear eigrp [*as_number*] **neighbors** [*ip_addr* | *if_name*] [**soft**]

Syntax Description

as_number	(Optional) Specifies the autonomous system number of the EIGRP process for which you are deleting neighbor entries. Because the device only supports one EIGRP routing process, you do not need to specify the autonomous system number (AS), which is the process ID.
if_name	(Optional) The name of an interface. Specifying an interface name removes all neighbor table entries that were learned through this interface.
ip_addr	(Optional) The IP address of the neighbor you want to remove from the neighbor table.
soft	Causes the device to resynchronize with the neighbor without resetting the adjacency.

Command Default

If you do not specify a neighbor IP address or an interface name, all dynamic entries are removed from the neighbor table.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

The **clear eigrp neighbors** command does not remove neighbors that were manually defined from the neighbor table. Only dynamically discovered neighbors are removed.

You can use the **show eigrp neighbors** command to view the EIGRP neighbor table.

Examples

The following example removes all entries from the EIGRP neighbor table:

```
> clear eigrp neighbors
```

The following example removes all entries learned through the interface named “outside” from the EIGRP neighbor table:

```
> clear eigrp neighbors outside
```

Related Commands

Command	Description
show eigrp neighbors	Displays the EIGRP neighbor table.

clear eigrp topology

To delete entries from the EIGRP topology table, use the **clear eigrp topology** command.

clear eigrp [*as_number*] **topology** *ip_addr* [*mask*]

Syntax Description

as_number	(Optional) Specifies the autonomous system number of the EIGRP process. Because the device only supports one EIGRP routing process, you do not need to specify the autonomous system number (AS), which is the process ID.
ip_addr	The IP address to clear from the topology table.
mask	(Optional) The network mask to apply to the *ip-addr* argument.

Command History

Release	Modification
6.1	This command was introduced.

Usage Guidelines

This command clears existing EIGRP entries from the EIGRP topology table. You can use the **show eigrp topology** command to view the topology table entries.

Examples

The following example removes entries in the 192.168.1.0 network from EIGRP topology table:

```
> clear eigrp topology 192.168.1.0 255.255.255.0
```

Related Commands

Command	Description
show eigrp topology	Displays the EIGRP topology table.