CHAPTER 6

Configuring Event Action Rules

This chapter explains how to configure event action rules. It contains the following sections:

Understanding Event Action Rules

Event action rules are a group of settings you configure for the event action processing component of the sensor. These rules dictate the actions the sensor performs when an event occurs.

The event action processing component is responsible for the following functions:

- Calculating the risk rating
- Adding event action overrides
- Filtering event action
- Executing the resulting event action
- Summarizing and aggregating events
- Maintaining a list of denied attackers

Signature Event Action Processor

SEAP coordinates the data flow from the signature event in the alarm channel to processing through the SEAO, the SEAF, and the SEAH. It consists of the following components:

- Alarm channel

  The unit that represents the area to communicate signature events from the Sensor App inspection path to signature event handling.

- Signature event action override (SEAO)

  Adds actions based on the RR value. SEAO applies to all signatures that fall into the range of the configured RR threshold. Each SEAO is independent and has a separate configuration value for each action type. For more information, see Calculating the Risk Rating, page 6-8.

- Signature event action filter (SEAF)

  Subtracts actions based on the signature event's signature ID, addresses, and RR. The input to the SEAF is the signature event with actions possibly added by the SEAO.

  Note The SEAF can only subtract actions, it cannot add new actions.

  The following parameters apply to the SEAF:

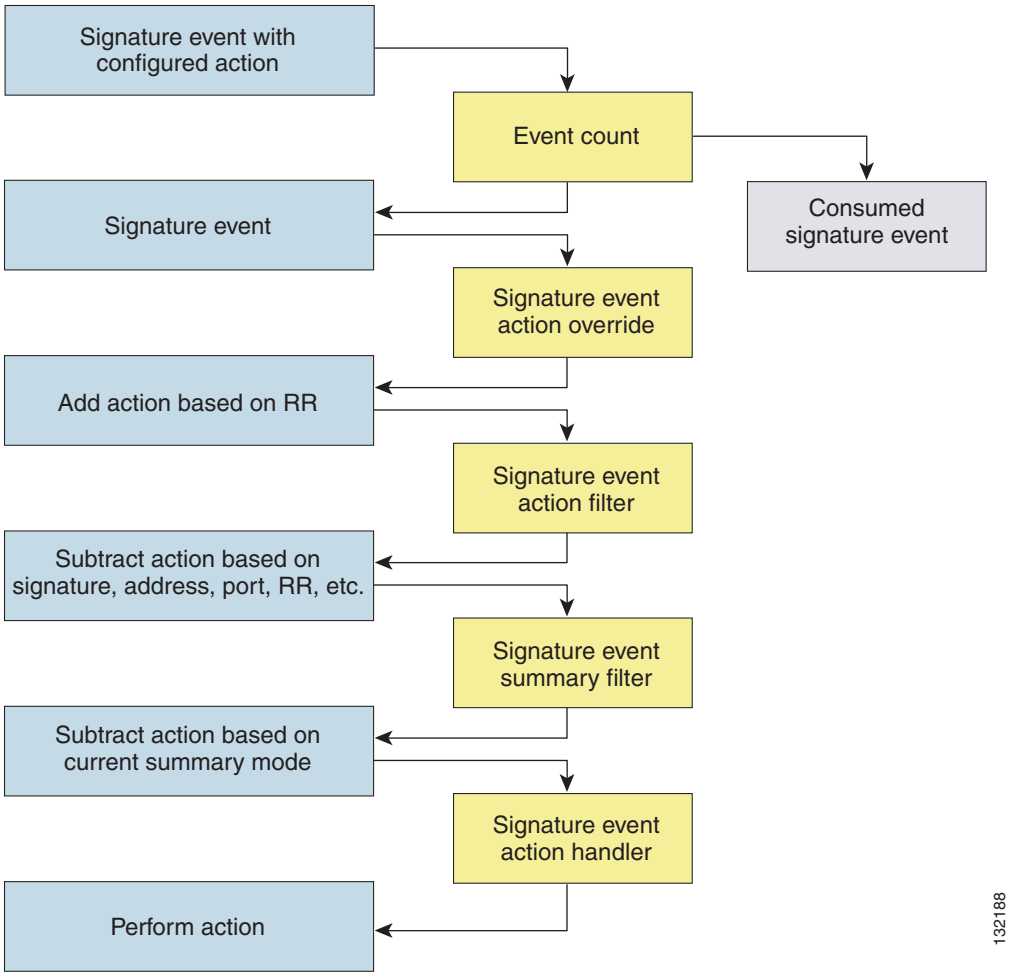
  - Signature ID
  - Subsignature ID
  - Attacker address
  - Attacker port
  - Victim address
  - Victim port
  - RR threshold range
  - Actions to subtract
  - Sequence identifier (optional)
  - Stop-or-continue bit
  - Enable action filter line bit

- Signature event action handler (SEAH)

  Performs the requested actions. The output from the SEAH is the actions being performed and possibly an evIdsAlert written to the Event Store.

Figure 6-1 on page 6-3 illustrates the logical flow of the signature event through the SEAP and the operations performed on the action for this event. It starts with the signature event with configured action received in the alarm channel and flows top to bottom as the signature event passes through the functional components of the SEAP.

Figure 6-1 Signature Event Through SEAP



Event Actions

Table 6-1 describes the event actions.

Table 6-1 Event Actions

Event Action Name	Description
Deny Attacker Inline	(Inline mode only) Does not transmit this packet and future packets originating from the attacker address for a specified period of time.[1] **Note** This is the most severe of the deny actions. It denies current and future packets from a single attacker address. You can clear all denied attacker entries, which permits the addresses back on the network. For the procedure, see Monitoring and Clearing the Denied Attackers List, page 6-21.
Deny Attacker Service Pair Inline	(Inline mode only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
Deny Attacker Victim Pair Inline	(Inline mode only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time. **Note** For deny actions, you can set the specified period of time and maximum number of denied attackers. For the procedure, see Configuring the General Settings, page 6-20.
Deny Connection Inline	(Inline mode only) Does not transmit this packet and future packets on the TCP flow.
Deny Packet Inline	(Inline mode only) Does not transmit this packet. **Note** You cannot delete the event action override for Deny Packet Inline because it is protected. If you do not want to use that override, disable it.
Log Attacker Packets	Starts IP logging packets containing the attacker address. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Log Pair Packets	Starts IP logging packets containing the attacker-victim address pair. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Log Victim Packets	Starts IP logging packets containing the victim address.
Modify Packet Inline	Modifies packet data to remove ambiguity about what the end point might do with the packet. **Note** Modify Packet Inline is not an option for Add Event Action Filter or Add Event Action Override.

Table 6-1 Event Actions (continued)

Event Action Name	Description
Produce Alert	Writes the event to the Event Store as an alert. **Note** The Produce Alert action is not automatic when you enable alerts for a signature. To have an alert created in the Event Store, you must select Produce Alert. If you add a second action, you must include Produce Alert if you want an alert sent to the Event Store. Also, every time you configure the event actions, a new list is created and it replaces the old list. Make sure you include all the event actions you need for each signature.
Produce Verbose Alert	Includes an encoded dump of the offending packet in the alert. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected.
Request Block Connection	Sends a request to ARC to block this connection. **Note** You must have blocking devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
Request Block Host	Sends a request to ARC to block this attacker host. **Note** You must have blocking devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting." **Note** For block actions, you can set the duration of the block. For the procedure, see Configuring the General Settings, page 6-20.
Request Rate Limit	Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action. For more information, see Chapter 10, "Configuring Attack Response Controller for Blocking and Rate Limiting."
Request SNMP Trap	Sends a request to NotificationApp to perform SNMP notification. **Note** This action causes an alert to be written to the Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 11, "Configuring SNMP."
Reset TCP Connection	Sends TCP resets to hijack and terminate the TCP flow. **Note** Reset TCP Connection only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

1. The sensor maintains a list of attackers being denied by the system. To remove an entry from the denied attacker list, you can view the list of attackers and clear the entire list, or you can wait for the timer to expire. The timer is a sliding timer for each entry. Therefore, if attacker A is being denied, but issues another attack, the timer for attacker A is reset and attacker A remains in the denied attacker list until the timer expires. If the denied attacker list is at capacity and cannot add a new entry, the packet will still be denied.

Understanding Deny Packet Inlin

For signatures that have deny-packet-inline configured as an action or for an event action override that adds deny-packet-inline as an action, the following actions may be taken:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

The deny packet inline action is represented as a dropped packet action in the alert. When a deny packet inline occurs for a TCP connection, it is automatically upgraded to a deny connection inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a deny connection inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

Task List for Configuring Event Action Rules

Follow these steps when configuring the event action rules component of the IPS:

1. Create any variables that you want to use in event action filters.
2. Create TVRs.

   Assign TVRs to your network assets so that you can calculate the RR.
3. Create overrides to add actions based on the RR value.

   Assign an RR to each event action type.
4. Create filters.

   Assign filters to subtract actions based on the signature's ID, IP addresses, and RR.
5. Configure the general settings.

   Specify whether you want to use the summarizer, the meta event generator, or configure denied attacker parameters.

Event Action Variables

This section describes event action variables, and contains the following topics:

Understanding Event Action Variables

You can create event action variables and then use those variables in event action filters. When you want to use the same value within multiple filters, use a variable. When you change the value of the variable, any filter that uses that variable is updated with the new value.

Note You must preface the variable with a dollar ($) sign to indicate that you are using a variable rather than a string.

Some variables cannot be deleted because they are necessary to the signature system. If a variable is protected, you cannot edit it. You receive an error message if you try to delete protected variables. You can edit only one variable at a time.

When configuring IP addresses, specify the full IP address or ranges or set of ranges. For example:

- 10.90.1.1
- 10.89.10.10-10.89.10.23
- 10.1.1.1-10.2.255.255, 10.89.10.10-10.89.10.23

Timesaver For example, if you have an IP address space that applies to your engineering group and there are no Windows systems in that group, and you are not worried about any Windows-based attacks to that group, you could set up a variable to be the engineering group's IP address space. You could then use this variable to configure a filter that would ignore all Windows-based attacks for this group.

Configuring Event Action Variables

Use the **variables** *variable_name* **address** *ip_address* command in service event action rules submode to set up event action variables. The IP address can be one address, a range, or ranges separated by a comma.

To configure event action variables, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
```

Step 3 Create a variable:

```
sensor(config-rul)# variables variable1 address 10.89.130.108
```

The valid values for **address** are A.B.C.D-A.B.C.D [,A.B.C.D-A.B.C.D].

Step 4 Check the variable you just made:

```
sensor(config-rul)# show settings
   variables (min: 0, max: 256, current: 2)
   -----------------------------------------------
      variableName: variable1
      -----------------------------------------------
         address: 10.89.130.108 default: 0.0.0.0-255.255.255.255
      -----------------------------------------------
```

```
-----------------------------------------------
```

Step 5 Exit event action rules submode:

```
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply your changes or enter **no** to discard them.

Target Value Ratings

This section describes what RR is and how to use it to configure the TVR. This section contains the following topics:

Calculating the Risk Rating

An RR is a value between 0 and 100 that represents a numerical quantification of the risk associated with a particular event on the network. The calculation takes into account the value of the network asset being attacked (for example, a particular server), so it is configured on a per-signature basis (ASR and SFR) and on a per-server basis (TVR).

RRs let you prioritize alerts that need your attention. These RR factors take into consideration the severity of the attack if it succeeds, the fidelity of the signature, and the overall value of the target host to you. The RR is reported in the evIdsAlert.

The following values are used to calculate the RR for a particular event:

- **Attack Severity Rating (ASR)**—A weight associated with the severity of a successful exploit of the vulnerability.

  The ASR is derived from the alert severity parameter of the signature.

- **Signature Fidelity Rating (SFR)**—A weight associated with how well this signature might perform in the absence of specific knowledge of the target.

  SFR is calculated by the signature author on a per-signature basis. The signature author defines a baseline confidence ranking for the accuracy of the signature in the absence of qualifying intelligence on the target. It represents the confidence that the detected behavior would produce the intended effect on the target platform if the packet under analysis were allowed to be delivered. For example, a signature that is written with very specific rules (specific regular expression) has a higher SFR than a signature that is written with generic rules.

- **Target Value Rating (TVR)**—A weight associated with the perceived value of the target.

  TVR is a user-configurable value that identifies the importance of a network asset (through its IP address). You can develop a security policy that is more stringent for valuable corporate resources and looser for less important resources. For example, you could assign a TVR to the company web server that is higher than the TVR you assign to a desktop node. In this example, attacks against the company web server have a higher RR than attacks against the desktop node.

Note RR is a product of ASR, SFR, and TVR with an optional PD (promiscuous delta) subtracted in promiscuous mode only.

Configuring the Target Value Rating

You can assign a TVR to your network assets. T he TVR is one of the factors used to calculate the RR value for each alert. You can assign different TVRs to different targets. Events with a higher RR trigger more severe signature event actions.

Use the **target-value target-value-setting** {**zerovalue | low | medium | high | mission-critical**} **target-address** *ip_address* command in service event action rules submode to set TVRs for your network assets. The default is medium.

The following options apply:

- **target-address** *ip_address*—Range set of IP address(es).
- **target-value-setting**—Choose one of the following:
  - **zerovalue**—No value of this target.
  - **low**—Lower value of this target.
  - **medium**—Normal value of this target.
  - **high**—Elevated value of this target.
  - **mission-critical**—Extreme value of this target.

To configure TVRs for your network assets, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
```

Step 3 Assign the TVR to the network asset:

```
sensor(config-rul)# target-value target-value-setting mission-critical target-address
10.89.130.108
```

Step 4 Check the TVR setting you just configured:

```
sensor(config-rul)# show settings
-----------------------------------------------
   target-value (min: 0, max: 5, current: 1)
   -----------------------------------------------
      target-value-setting: mission-critical
      target-address: 10.89.130.108 default: 0.0.0.0-255.255.255.255
   -----------------------------------------------
sensor(config-rul)#
```

Step 5 Exit event action rules submode:

```
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 6 Press **Enter** to apply your changes or enter `no` to discard them.

Event Action Overrides

This section describes event action overrides, and contains the following topics:

Understanding Event Action Overrides

You can add an event action override to change the actions associated with an event based on the RR of that event. Event action overrides are a way to add event actions globally without having to configure each signature individually. Each event action has an associated RR range. If a signature event occurs and the RR for that event falls within the range for an event action, that action is added to the event. For example, if you want any event with an RR of 85 or more to generate an SNMP trap, you can set the RR range for Request SNMP Trap to 85-100. If you do not want to use action overrides, you can disable the entire event action override component.

Understanding Deny Packet Inline

For signatures that have deny-packet-inline configured as an action or for an event action override that adds deny-packet-inline as an action, the following actions may be taken:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

The deny packet inline action is represented as a dropped packet action in the alert. When a deny packet inline occurs for a TCP connection, it is automatically upgraded to a deny connection inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a deny connection inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

Configuring Event Action Overrides

Use the **overrides** {**request-block-connection** | **request-block-host** | **deny-attacker-inline** | **deny-packet-inline** | **deny-attacker-service-pair-inline** | **deny-attacker-victim-pair-inline** | **deny-connection-inline** | **log-attacker-packets** | **log-victim-packets** | **log-pair-packets** | **reset-tcp-connection** | **produce-alert** | **produce-verbose-alert** | **request-rate-limit** | **request-snmp-trap**} command in service event action rules submode to configure the parameters of event action overrides.

For a description of all the event actions, see Event Actions, page 6-4.

Note You cannot delete the event action override for **deny-packet-inline** because it is protected. If you do not want to use that override, disable it.

The following options apply:

- **no**—Removes an entry or selection setting.
- **override-item-status** {**enabled** | **disabled**}—Enables or disables the use of this override item. The default is enabled.
- **risk-rating-range**—Range of RR values for this override item. The default is 1 to 100.
- **show**—Displays system settings and/or history information.

To add event action overrides, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)#
```

Step 3 To configure how packets are treated for overrides:

a. To deny packets from the source IP address of the attacker:

```
sensor(config-rul)# overrides deny-attacker-inline
sensor(config-rul-ove)#
```

b. To not transmit the single packet causing the alert:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides deny-packet-inline
sensor(config-rul-ove)#
```

c. To not transmit packets on the specified TCP connection:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides deny-connection-inline
sensor(config-rul-ove)#
```

d. To send TCP RST packets to terminate the connection:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides reset-tcp-connection
sensor(config-rul-ove)#
```

Step 4 To configure overrides to request blocks:

a. To request a block of the connection:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides request-block-connection
sensor(config-rul-ove)#
```

b. To request a block of the attacker host:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides request-block-host
sensor(config-rul-ove)#
```

Step 5 To log packets for overrides:

a. To log the packets from the attacker IP address:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides log-attacker-packets
sensor(config-rul-ove)#
```

b. To log the packets from the victim IP address:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides log-victim-packets
sensor(config-rul-ove)#
```

c. To log packets from both the attacker and victim IP addresses:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides log-pair-packets
sensor(config-rul-ove)#
```

Step 6 To write alerts to the Event Store:

a. To write an alert to the Event Store:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides produce-alert
sensor(config-rul-ove)#
```

b. To write verbose alerts to the Event Store:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides produce-verbose-alert
sensor(config-rul-ove)#
```

c. To write events that request an SNMP trap to the Event Store:

```
sensor(config-rul-ove)# exit
sensor(config-rul)# overrides request-snmp-trap
sensor(config-rul-ove)#
```

Step 7 To configure the RR for this override item:

```
sensor(config-rul-ove)# risk-rating-range 85-100
```

Note The default RR range is 0 to 100. Set it to a different value, such as 85 to 100.

Step 8 To enable or disable the use of this override item:

```
sensor(config-rul-ove)# override-item-status {enabled | disabled}
```

The default is enabled.

Step 9 Verify the settings:

```
sensor(config-rul-ove)# show settings
   action-to-add: deny-attacker-inline default: produce-alert
   -----------------------------------------------
      override-item-status: Enabled default: Enabled
      risk-rating-range: 85-100 default: 0-100
   -----------------------------------------------
```

Step 10 Exit event action rules submode:

```
sensor(config-rul-ove)# exit
sensor(config-rul)#
Apply Changes:?[yes]:
```

Step 11 Press **Enter** to apply your changes or enter **no** to discard them.

Event Action Filters

This section describes event action filters. and contains the following topics:

Understanding Event Action Filters

Event action filters are processed as an ordered list and you can move filters up or down in the list. Filters let the sensor perform certain actions in response to the event without requiring the sensor to perform all actions or remove the entire event. Filters work by removing actions from an event. A filter that removes all actions from an event effectively consumes the event.

Note When filtering sweep signatures, we recommend that you do not filter the destination addresses. If there are multiple destination addresses, only the last address is used for matching the filter.

Caution Event action filters based on source and destination IP addresses do not function for the Sweep engine, because they do not filter as regular signatures. To filter source and destination IP addresses in sweep alerts, use the source and destination IP address filter parameters in the Sweep engine signatures.

Configuring Event Action Filters

You can configure event action filters to remove specific actions from an event or to discard an entire event and prevent further processing by the sensor. You can use event action variables that you defined to group addresses for your filters. For the procedure for configuring event action variables, see Configuring Event Action Variables, page 6-7.

Note You must preface the variable with a dollar sign ($) to indicate that you are using a variable rather than a string. Otherwise, you receive the `Bad source and destination` error.

Use the **filters {dit | insert | move] name1 [begin | end | inactive | before | after}** command in service event action rules submode to set up event action filters.

The following options apply:

- **actions-to-remove**—Event actions to remove for this filter item.
  - **deny-attacker-inline** —(Inline mode only) does not transmit this packet and future packets from the attacker address for a specified period of time.
  - **deny-attacker-service-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
  - **deny-attacker-victim-pair-inline**—(Inline only) Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
  - **deny-connection-inline**—(Inline mode only) Does not transmit this packet and future packets on the TCP Flow.
  - **deny-packet-inline**—(Inline mode only) Does not transmit this packet.
  - **log-attacker-packets**—Starts IP logging of packets containing the attacker address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-pair-packets**—Starts IP logging of packets containing the attacker-victim address pair. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **log-victim-packets**—Starts IP logging of packets containing the victim address. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **produce-alert** —Writes the event to the Event Store as an alert.
  - **produce-verbose-alert**—Includes an encoded dump (possibly truncated) of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected.
  - **request-block-connection**—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
  - **request-block-host**—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
  - **request-rate-limit**—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
  - **request-snmp-trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if **produce-alert** is not selected. You must have SNMP configured on the sensor to implement this action.
  - **reset-tcp-connection**—Sends TCP resets to hijack and terminate the TCP flow. **Reset TCP Connection** only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.
  - **modify-packet-inline**— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- **attacker-address-range**—Range set of attacker address(es) for this item (for example, 10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255).
- **attacker-port-range**—Range set of attacker port(s) for this item (for example, 147-147,8000-10000).
- **default**—Sets the value back to the system default setting.

- **deny-attacker-percentage**—Percentage of packets to deny for deny attacker features. The valid range is 0 to 100. The default is 100.
- **filter-item-status** {**enabled** | **disabled**}—Enables or disables the use of this filter item.
- **no**—Removes an entry or selection setting.
- **risk-rating-range**—Range of RR values for this filter item.
- **signature-id-range**—Range set of signature ID(s) for this item (for example, 1000-2000,3000-3000).
- **stop-on-match**—Continues evaluating filters or stops when this filter item is matched.
- **subsignature-id-range**—Range set of subsignature ID(s) for this item (for example, 0-2,5-5).
- **user-comment** —Lets you add your comments about this filter item.
- **victim-address-range**—Range set of victim address(es) for this item (for example, 10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255).
- **victim-port-range**—Range set of victim port(s) for this item (for example, 147-147,8000-10000).

To configure event action filters, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
sensor(config-rul)#
```

Step 3 Create the filter name:

```
sensor(config-rul)# filters insert name1 begin
```

Use **name1**, **name2**, and so forth to name your event action filters. Use the **begin | end | inactive | before | after** keywords to specify where you want to insert the filter.

Step 4 Configure the values for this filter:

a. Set the signature ID range:

```
sensor(config-rul-fil)# signature-id-range 1000-1005
```

The default is 900 to 65535.

b. Set the subsignature ID range:

```
sensor(config-rul-fil)# subsignature-id-range 1-5
```

The default is 0 to 255.

c. Set the attacker address range:

```
sensor(config-rul-fil)# attacker-address-range 10.89.10.10-10.89.10.23
```

The default is 0.0.0.0 to 255.255.255.255.

d. Set the victim address range:

```
sensor(config-rul-fil)# victim-address-range 192.56.10.1-192.56.10.255
```

The default is 0.0.0.0 to 255.255.255.255.

e. Set the victim port range:

```
sensor(config-rul-fil)# victim-port-range 0-434
```

The default is 0 to 65535.

f. Set the risk rating range:

```
sensor(config-rul-fil)# risk-rating-range 85-100
```

The default is 0 to 100.

g. Set the actions to remove:

```
sensor(config-rul-fil)# actions-to-remove reset-tcp-connection
```

h. If you are filtering a deny action, set the percentage of deny actions you want:

```
sensor(config-rul-fil)# deny-attacker-percentage 90
```

The default is 100.

i. Set the status of the filter to either disabled or enabled.

```
sensor(config-rul-fil)# filter-item-status {enabled | disabled}
```

The default is enabled.

j. Set the stop on match parameter.

```
sensor(config-rul-fil)# stop-on-match {true | false}
```

True tells the sensor to stop processing filters if this item matches. **False** tells the sensor to continue processing filters even if this item matches.

k. Add any comments you want to explain this filter:

```
sensor(config-rul-fil)# user-comments
```

Step 5 Verify the settings for the filter:

```
sensor(config-rul-fil)# show settings
   NAME: name1
   ----------------------------------------------
      signature-id-range: 1000-10005 default: 900-65535
      subsignature-id-range: 1-5 default: 0-255
      attacker-address-range: 10.89.10.10-10.89.10.23 default: 0.0.0.0-255.255.255.255
      victim-address-range: 192.56.10.1-192.56.10.255 default: 0.0.0.0-255.255.255.255
      attacker-port-range: 0-65535 <defaulted>
      victim-port-range: 1-343 default: 0-65535
      risk-rating-range: 85-100 default: 0-100
      actions-to-remove: reset-tcp-connection default:
      deny-attacker-percentage: 90 default: 100
      filter-item-status: Enabled default: Enabled
      stop-on-match: True default: False
      user-comment: This is a new filter. default:
   ----------------------------------------------
ssenor(config-rul-fil)#
```

Step 6 To edit an existing filter:

```
sensor(config-rul)# filters edit name1
```

Step 7 Edit the parameters (see Steps 4a through 4k).

Step 8 To move a filter up or down in the filter list:

```
sensor(config-rul-fil)# exit
sensor(config-rul)# filters move name5 before name1
```

Step 9 Verify that you have moved the filters:

```
sensor(config-rul-fil)# exit
sensor(config-rul)# show settings
   -----------------------------------------------
   filters (min: 0, max: 4096, current: 5 - 4 active, 1 inactive)
   -----------------------------------------------
   ACTIVE list-contents
   -----------------------------------------------
      NAME: name5
      -----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      -----------------------------------------------
      -----------------------------------------------
      NAME: name1
      -----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      -----------------------------------------------
      -----------------------------------------------
      NAME: name2
      -----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      -----------------------------------------------
      -----------------------------------------------
   -----------------------------------------------
   INACTIVE list-contents
   -----------------------------------------------
-----------------------------------------------
sensor(config-rul)#
```

Step 10 To move a filter to the inactive list:

```
sensor(config-rul)# filters move name1 inactive
```

Step 11 Verify that the filter has been moved to the inactive list:

```
sensor(config-rul-fil)# exit
sensor(config-rul)# show settings
   -----------------------------------------------
   INACTIVE list-contents
   -----------------------------------------------
      -----------------------------------------------
      NAME: name1
      -----------------------------------------------
         signature-id-range: 900-65535 <defaulted>
         subsignature-id-range: 0-255 <defaulted>
         attacker-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         victim-address-range: 0.0.0.0-255.255.255.255 <defaulted>
         attacker-port-range: 0-65535 <defaulted>
         victim-port-range: 0-65535 <defaulted>
         risk-rating-range: 0-100 <defaulted>
         actions-to-remove: <defaulted>
         filter-item-status: Enabled <defaulted>
         stop-on-match: False <defaulted>
         user-comment: <defaulted>
      -----------------------------------------------
      -----------------------------------------------
sensor(config-rul)#
```

Step 12 Exit event action rules submode:

```
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 13 Press **Enter** to apply your changes or enter **no** to discard them.

General Settings

This section describes the general settings, and contains the following topics:

Understanding the General Settings

You can configure the general settings that apply to event action rules, such as whether you want to use the summarizer and the meta event generator. The summarizer groups events into a single alert, thus decreasing the number of alerts the sensor sends out. The meta event generator processes the component events, which lets the sensor watch for suspicious activity transpiring over a series of events.

You can configure how long you want to deny attackers, the maximum number of denied attackers, and how long you want blocks to last.

Understanding Event Action Summarization

Summarization decreases the volume of alerts sent out from the sensor by providing basic aggregation of events into a single alert. Special parameters are specified for each signature and they influence the handling of the alerts. Each signature is created with defaults that reflect a preferred normal behavior. However, you can tune each signature to change this default behavior within the constraints for each engine type.

The non-alert generating actions (deny, block, TCP reset) go through the filters for each signature event unsummarized. The alert-generating actions are not performed on these summarized alerts; instead the actions are applied to the one summary alert and then put through the filters.

If you select one of the other alert-generating actions and do not have it filtered out, the alert is created even if you do not select Produce Alert. To prevent alerts from being created, you must have all alert-generating actions filtered out.

Summarization and event actions are processed after the Meta engine has processed the component events. This lets the sensor watch for suspicious activity transpiring over a series of events.

Understanding Event Action Aggregation

Basic aggregation provides two operating modes. The simple mode involves configuring a threshold number of hits for a signature that must be met before the alert is sent. A more advanced mode is timed-interval counting. In this mode, the sensor tracks the number of hits per second and only sends alerts when that threshold is met. In this example, a *hit* is a term used to describe an event, which is basically an alert, but it is not sent out of the sensor as an alert until the threshold number of hits has been exceeded.

You can select from the following summarization options:

- Fire All—Fire All mode fires an alert each time the signature is triggered. If the threshold is set for summarization, the following happens: Alerts are fired for each execution until summarization occurs. After summarization starts only one alert every summary interval fires for each address set. Alerts for other address sets are either all seen or separately summarized. The signature reverts to Fire All mode after a period of no alerts of that signature.
- Summary—Summary mode fires an alert the first time a signature is triggered, and then additional alerts for that signature are summarized for the duration of the summary interval. Only one alert every summary interval should fire for each address set. If the global summary threshold is reached, the signature goes into Global Summarization mode.
- Global Summarization—Global Summarization mode fires an alert for every summary interval. Signatures can be preconfigured for global summarization.
- Fire Once—Fire Once mode fires an alert for each address set. You can upgrade this mode to Global Summarization mode.

Understanding the Deny Attackers Inline Event Action

You can configure certain aspects of the deny attackers inline event action. You can configure the number of seconds you want to deny attackers inline and you can limit the number of attackers you want denied in the system at any one time.

Configuring the General Settings

Use the following commands in service event action rules submode to configure general event action rules settings:

- **global-block-timeout** —Number of minutes to block a host or connection.

  The valid range is 0 to 10000000. The default is 30 minutes.

- **global-deny-timeout**—Number of seconds to deny attackers inline.

  The valid range is 0 to 518400. The default is 3600.

- **global-filters-status** {**enabled | disabled**}—Enables or disables the use of the filters.

  The default is enabled.

- **global-metaevent-status {enabled | disabled}**—Enables or disables the use of the Meta Event Generator.

  The default is enabled.

- **global-overrides-status** {**enabled | disabled**}—Enables or disables the use of the overrides.

  The default is enabled.

- **global-summarization-status {enabled | disabled}**—Enables or disables the use of the summarizer.

  The default is enabled.

- **max-denied-attackers**—Limits the number of denied attackers possible in the system at any one time.

  The valid range is 0 to 100000000. The default is 10000.

To configure event action general settings, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Enter event action rules submode:

```
sensor# configure terminal
sensor(config)# service event-action-rules rules0
```

Step 3 Enter general submode:

```
sensor(config)# general
```

Step 4 To enable or disable the meta event generator:

```
sensor(config-rul-gen)# global-metaevent-status {enabled | disabled}
```

The default is enabled.

Step 5 To enable or disable the summarizer:

```
sensor(config-rul-gen)# global-summarization-status {enabled | disabled}
```

The default is enabled.

Step 6 To configure the denied attackers inline event action:

a. To limit the number of denied attackers in the system at any given time:

```
sensor(config-rul-gen)# max-denied-attackers 100
```

The default is 1000.

b. To configure the amount of seconds to deny attackers in the system:

```
sensor(config-rul-gen)# global-deny-timeout 1000
```

The default is 3600 seconds.

Step 7 To configure the number of minutes to block a host or a connection:

```
sensor(config-rul-gen)# global-block-timeout 20
```

The default is 30 minutes.

Step 8 To enable or disable any overrides that you have set up:

```
sensor(config-rul-gen)# global-overrides-status {enabled | disabled}
```

The default is enabled.

Step 9 To enable or disable any filters that you have set up:

```
sensor(config-rul-gen)# global-filters-status {enabled | disabled}
```

The default is enabled.

Step 10 Check the settings for general submode:

```
sensor(config-rul-gen)# show settings
   general
   -----------------------------------------------
      global-overrides-status: Enabled default: Enabled
      global-filters-status: Enabled default: Enabled
      global-summarization-status: Enabled default: Enabled
      global-metaevent-status: Enabled default: Enabled
      global-deny-timeout: 1000 default: 3600
      global-block-timeout: 20 default: 30
      max-denied-attackers: 100 default: 10000
   -----------------------------------------------
sensor(config-rul-gen)#
```

Step 11 Exit event action rules submode:

```
sensor(config-rul-gen)# exit
sensor(config-rul)# exit
Apply Changes:?[yes]:
```

Step 12 Press **Enter** to apply your changes or enter **no** to discard them.

Monitoring and Clearing the Denied Attackers List

Use the **show statistics denied-attackers** command to display the list of denied attackers. Use the **clear denied-attackers** command in service event action rules submode to delete the denied attackers list and clear the virtual sensor statistics.

If your sensor is configured to operate in inline mode, the traffic is passing through the sensor. You can configure signatures to deny packets, connections, and attackers while in inline mode, which means that single packets, connections, and specific attackers will be denied, that is, not transmitted, when the sensor encounters them.

When the signature fires, the attacker is denied and placed in a list. As part of sensor administration, you may want to delete the list or clear the statistics in the list.

To display the list of denied attackers and delete the list and clear the statistics, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Display the list of denied IP addresses:

```
sensor# show statistics denied-attackers
Denied Attackers and hit count for each.
   10.20.4.2 = 9
   10.20.5.2 = 5
```

The statistics show that there are two IP addresses being denied at this time.

Step 3 Delete the denied attackers list:

```
sensor# clear denied-attackers
Warning: Executing this command will delete all addresses from the list of
attackers currently being denied by the sensor.
Continue with clear? [yes]:
```

Step 4 Enter **yes** to clear the list.

Step 5 Verify that you have cleared the list:

```
sensor# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = mypair
      Denied Address Information
         Number of Active Denied Attackers = 0
         Number of Denied Attackers Inserted = 2
         Number of Denied Attackers Total Hits = 287
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 1
      Denied Attackers and hit count for each.
```

There is no longer any information under the `Denied Attackers and hit count for each category`.

Step 6 To clear only the statistics:

```
sensor# show statistics virtual-sensor clear
```

Step 7 Verify that you have cleared the statistics:

```
JWK-4255# show statistics virtual-sensor
Virtual Sensor Statistics
   Statistics for Virtual Sensor vs0
      Name of current Signature-Definition instance = sig0
      Name of current Event-Action-Rules instance = rules0
      List of interfaces monitored by this virtual sensor = mypair
      Denied Address Information
         Number of Active Denied Attackers = 2
         Number of Denied Attackers Inserted = 0
         Number of Denied Attackers Total Hits = 0
         Number of times max-denied-attackers limited creation of new entry = 0
         Number of exec Clear commands during uptime = 1
      Denied Attackers and hit count for each.
         10.20.2.5 = 0
         10.20.5.2 = 0
```

The statistics have all been cleared except for the `Number of Active Denied Attackers` and `Number of exec Clear commands during uptime` categories. It is important to know if the list has been cleared.

Event Action Rules Example

The following example demonstrates how the individual components of your event action rules work together.

Risk Rating Ranges for Example 1

- **Produce Alert**—1-100
- **Produce Verbose Alert**—90-100
- **Request SNMP Trap**—50-100
- **Log Pair Packets**—90-100
- **Log Victim Packets**—90-100
- **Log Attacker Packets**—90-100
- **Reset TCP Connection**—90-100
- **Request Block Connection**—70-89
- **Request Block Host**—90-100
- **Deny Attacker Inline**—0-0
- **Deny Connection Inline**—90-100
- **Deny Packet Inline**—90-100

Event Action Filters for Example 1

1. SigID=2004, Attacker Address=*, Victim Address=20.1.1.1, Actions to Remove=ALL, Risk Rating Range=1-100, StopOnMatch=True
2. SigID=2004, Attacker Address=30.1.1.1, Victim Address=*, Actions to Remove=ALL, Risk Rating Range=1-100, StopOnMatch=True
3. SigID=2004, Attacker Address=*, Victim Address=*, Actions to Remove=None, Risk Rating Range=95-100, StopOnMatch=True
4. SigID=2004, Attacker Address=*, Victim Address=*, Actions to Remove=denyAttackerInline, requestBlockHost, requestBlockConnection, Risk Rating Range=56-94, StopOnMatch=True
5. SigID=2004, Attacker Address=*, Victim Address=*, Actions to Remove=denyAttackerInline, requestBlockHost, produceAlert, resetTcpConnection, logAttackerPackets, Risk Rating Range=1-55, StopOnMatch=True

Results for Example 1

When SIG 2004 is detected:

- If the attacker address is 30.1.1.1 or the victim address is 20.1.1.1, the event is consumed (ALL actions are subtracted).

If the attacker address is not 30.1.1.1 and the victim address is not 20.1.1.1:

- If the RR is 50, Produce Alert and Request SNMP Trap are added by the event action override component, but Produce Alert is subtracted by the event action filter. However, the event action policy forces the alert action because Request SNMP Trap is dependent on the <evIdsAlert>.
- If the RR is 89, Request SNMP Trap and Request Block Connection are added by the event action override component. However, Request Block Connection is subtracted by the event action filter.
- If the RR is 96, all actions except Deny Attacker Inline and Request Block Connection are added by the event action override component, and none are removed by the event action filter. The third filter line with the filter action NONE is optional, but is presented as a clearer way to define this type of filter.

Monitoring Events

This section describes how to display and clear events from the Event Store, and contains the following topics:

Displaying Events

Use the **show events** [{[**alert** [informational] [low] [medium] [high] [**include-traits** ***traits***] [**exclude-traits** ***traits***]] | **error** [warning] [error] [fatal] | **NAC** | **status**}] [*hh:mm:ss* [*month day* [*year*]] | **past** *hh:mm:ss*] command to display events from the Event Store.

Events are displayed beginning at the start time. If you do not specify a start time, events are displayed beginning at the current time. If you do not specify an event type, all events are displayed.

Note Events are displayed as a live feed until you cancel the request by pressing Ctrl-C.

The following options apply:

- **alert**—Displays alerts. Provides notification of some suspicious activity that may indicate an attack is in process or has been attempted.

  If no level is selected (informational, low, medium, or high), all alert events are displayed.
- **include-traits**—Displays alerts that have the specified traits.
- **exclude-traits**—Does not display alerts that have the specified traits.
- **traits**—Trait bit position in decimal (0 to 15).
- **error**—Displays error events. Error events are generated by services when error conditions are encountered.

- **NAC**—Displays Attack Response Controller (ARC) requests.

Note ARC is formerly known as Network Access Controller (NAC). This name change has not been completely implemented throughout the IDM and CLI for IPS 5.1.

- **status**—Displays status events.
- **past**—Displays events starting in the past for the specified hours, minutes, and seconds.
- *hh:mm:ss*—Hours, minutes, and seconds in the past to begin the display.

Note The **show events** command waits until a specified event is available. It continues to wait and display events until you exit by pressing Ctrl-C.

To display events from the Event Store, follow these steps:

Step 1 Log in to the CLI.

Step 2 Display all events starting now:

```
sensor#@ show events
evError: eventId=1041472274774840147 severity=warning vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 12075
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown

evError: eventId=1041472274774840148 severity=error vendor=Cisco
  originator:
    hostId: sensor2
    appName: cidwebserver
    appInstanceId: 351
  time: 2003/01/07 04:41:45 2003/01/07 04:41:45 UTC
  errorMessage: name=errTransport WebSession::sessionTask(6) TLS connection exce
ption: handshake incomplete.
```

The feed continues showing all events until you press **Ctrl-C**.

Step 3 Display the block requests beginning at 10:00 a.m. on February 9, 2005:

```
sensor#@ show events NAC 10:00:00 Feb 9 2005
evShunRqst: eventId=1106837332219222281 vendor=Cisco
  originator:
    deviceName: Sensor1
    appName: NetworkAccessControllerApp
    appInstance: 654
  time: 2005/02/09 10:33:31 2004/08/09 13:13:31
  shunInfo:
    host: connectionShun=false
      srcAddr: 11.0.0.1
      destAddr:
      srcPort:
      destPort:
      protocol: numericType=0 other
    timeoutMinutes: 40
  evAlertRef: hostId=esendHost 123456789012345678
sensor#
```

Step 4 Display errors with the warning level starting at 10:00 a.m. February 9 2005:

```
sensor# show events error warning 10:00:00 Feb 9 2005
evError: eventId=1041472274774840197 severity=warning vendor=Cisco
  originator:
    hostId: sensor
    appName: cidwebserver
    appInstanceId: 12160
  time: 2003/01/07 04:49:25 2003/01/07 04:49:25 UTC
  errorMessage: name=errWarning received fatal alert: certificate_unknown
```

Step 5 Display alerts from the past 45 seconds:

```
sensor# show events alert past 00:00:45

evIdsAlert: eventId=1109695939102805307 severity=medium vendor=Cisco
  originator:
    hostId: sensor
    appName: sensorApp
    appInstanceId: 367
  time: 2005/03/02 14:15:59 2005/03/02 14:15:59 UTC
  signature: description=Nachi Worm ICMP Echo Request id=2156 version=S54
    subsigId: 0
    sigDetails: Nachi ICMP
  interfaceGroup:
  vlan: 0
  participants:
    attacker:
      addr: locality=OUT 10.89.228.202
    target:
      addr: locality=OUT 10.89.150.185
  riskRatingValue: 70
  interface: fe0_1
  protocol: icmp


evIdsAlert: eventId=1109695939102805308 severity=medium vendor=Cisco
  originator:
--MORE--
```

Step 6 Display events that began 30 seconds in the past:

```
sensor# show events past 00:00:30
evStatus: eventId=1041526834774829055 vendor=Cisco
  originator:
    hostId: sensor
    appName: mainApp
    appInstanceId: 2215
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
  controlTransaction: command=getVersion successful=true
    description: Control transaction response.
    requestor:
      user: cids
      application:
        hostId: 64.101.182.101
        appName: -cidcli
        appInstanceId: 2316


evStatus: eventId=1041526834774829056 vendor=Cisco
  originator:
    hostId: sensor
    appName: login(pam_unix)
    appInstanceId: 2315
  time: 2003/01/08 02:41:00 2003/01/08 02:41:00 UTC
```

```
syslogMessage:
  description: session opened for user cisco by cisco(uid=0)
```

Clearing Events from Event Store

Use the **clear events** command to clear Event Store.

To clear events from Event Store, follow these steps:

Step 1 Log in to the CLI using an account with administrator privileges.

Step 2 Clear Event Store:

```
sensor# clear events
Warning: Executing this command will remove all events currently stored in the event
store.
Continue with clear? []:
```

Step 3 Enter **yes** to clear the events.