CHAPTER 8

Configuring Event Action Rules

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

Understanding Policies

You can create multiple security policies and apply them to individual virtual sensors. A security policy is made up of a signature definition policy, an event action rules policy, and an anomaly detection policy. IPS 6.0 contains a default signature definition policy called sig0, a default event action rules policy called rules0, and a default anomaly detection policy called ad0. You can assign the default policies to a virtual sensor or you can create new policies.

The use of multiple security policies lets you create security policies based on different requirements and then apply these customized policies per VLAN or physical interface.

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

This section contains the following topics:

Signature Event Action Processor

The Signature Event Action Processor coordinates the data flow from the signature event in the alarm channel to processing through the Signature Event Action Override, the Signature Event Action Filter, and the Signature Event Action Handler. 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

  Adds actions based on the risk rating value. Signature Event Action Override applies to all signatures that fall in the range of the configured risk rating threshold. Each Signature Event Action Override is independent and has a separate configuration value for each action type.

- Signature Event Action Filter

  Subtracts actions based on the signature ID, addresses, and risk rating of the signature event. The input to the Signature Event Action Filter is the signature event with actions possibly added by the Signature Event Action Override.

  Note The Signature Event Action Filter can only subtract actions, it cannot add new actions.

  The following parameters apply to the Signature Event Action Filter:

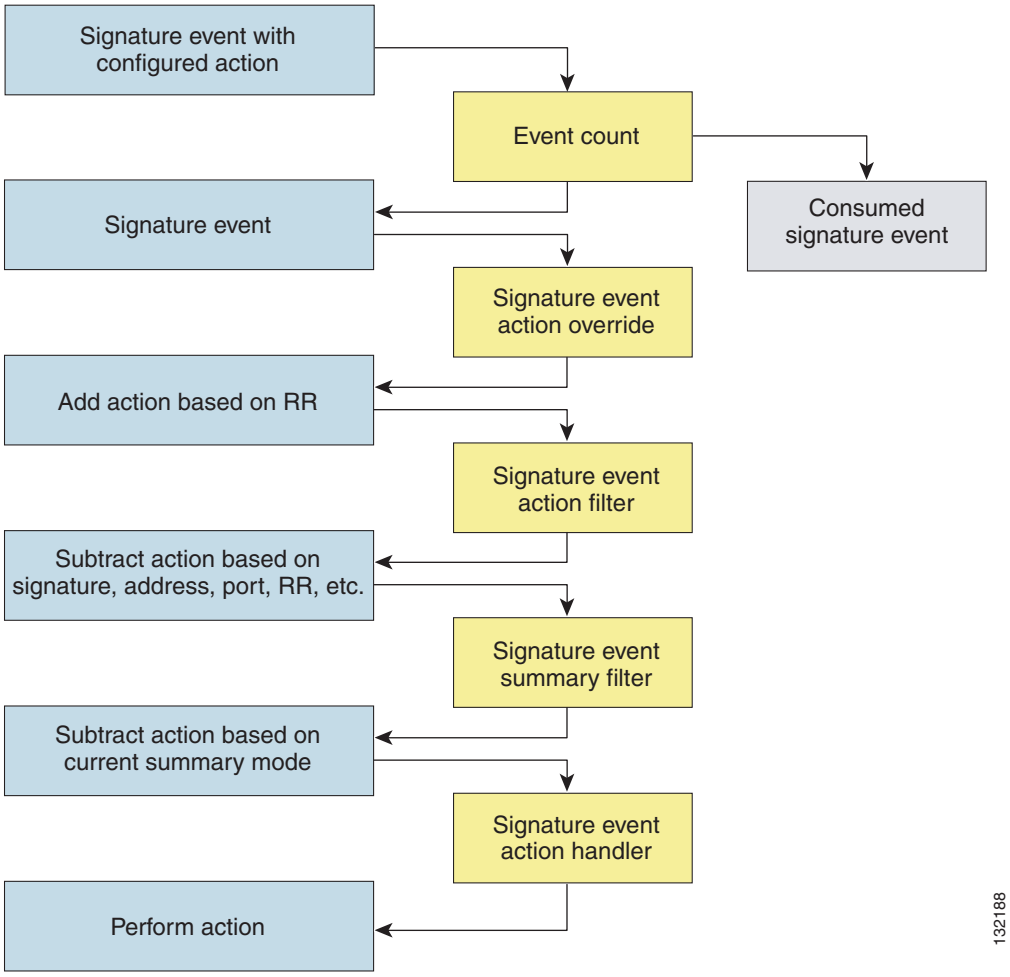
  - Signature ID
  - Subsignature ID
  - Attacker address
  - Attacker port
  - Victim address
  - Victim port
  - Risk rating threshold range
  - Actions to subtract

  - Sequence identifier (optional)
  - Stop-or-continue bit
  - Enable action filter line bit
  - Victim OS relevance or OS relevance
- Signature Event Action Handler

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

Figure 8-1 illustrates the logical flow of the signature event through the Signature Event Action Processor 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 Signature Event Action Processor.

Figure 8-1 Signature Event Through the Signature Event Action Processor



For More Information

For more information on risk rating, see Calculating the Risk Rating, page 8-11.

Event Actions

The Cisco IPS has the following event actions.

Alert and Log Actions

- 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.

Note There are other event actions that force a produce-alert. These actions use produce-alert as the vehicle for performing the action. Even if produce-alert is not selected or is filtered, the alert is still produced. The actions are the following: produce-verbose-alert, request-snmp-trap, log-attacker-packets, log-victim-packets, and log-pair-packets.

Note A produce-alert event action is added for an event when global correlation has increased the risk rating of an event, and has added either the deny-packet-inline or deny-attacker-inline event action.

- produce-verbose-alert—Includes an encoded dump 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.
- log-attacker-packets—Starts IP logging on packets that contain the attacker address and sends an alert. 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 on packets that contain the victim address and sends an alert. 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 on packets that contain the attacker/victim address pair. This action causes an alert to be written to the Event Store, even if produce-alert is not selected.
- 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.

Deny Actions

- deny-packet-inline (inline only)—Terminates the 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, set the override-item-status to disabled for that entry.

- deny-connection-inline (inline only)—Terminates the current packet and future packets on this TCP flow.

- 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-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-inline (inline only)—Terminates the current packet and future packets from this attacker address for a specified period of time.
- 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 is still denied.
- modify-packet-inline (inline only)—Modifies packet data to remove ambiguity about what the end point might do with the packet.

Note You cannot use modify-packet-inline as an action when adding event action filters or overrides.

Other Actions

- request-block-connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.

Note Connection blocks and network blocks are not supported on adaptive security appliances. Adaptive security appliances only support host blocks with additional connection information.

Note IPv6 does not support request-block-connection.

- request-block-host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.

Note IPv6 does not support request-block-host.

- 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.

Note The request-rate-limit action applies to a select set of signatures.

Note IPv6 does not support request-rate-limit.

- reset-tcp-connection—Sends TCP resets to hijack and terminate the TCP flow. The reset-tcp-connection action only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

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:

- dropped-packet
- denied-flow
- tcp-one-way-reset-sent

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.

TCP Reset Differences Between IPS Appliances and AIP SSM

The IPS appliance sends TCP reset packets to both the attacker and victim when reset-tcp-connection is selected. The IPS appliance sends a TCP reset packet only to the victim under the following circumstances:

- When a deny-packet-inline or deny-connection-inline is selected
- When TCP-based signatures and reset-tcp-connection have NOT been selected

In the case of the AIP SSM, the TCP reset request is sent to the ASA, and then the ASA sends the TCP reset packets. The ASA sends TCP reset packets to both the attacker and victim when the reset-tcp-connection is selected. When deny-packet-inline or deny-connection-inline is selected, the ASA sends the TCP reset packet to either the attacker or victim depending on the configuration of the signature. Signatures configured to swap the attacker and victim when reporting the alert can cause the ASA to send the TCP reset packet to the attacker.

For More Information

- For the procedure for configuring denied attackers, see Monitoring and Clearing the Denied Attackers List, page 8-33.
- For the procedure for configuring time and number of denied attackers, see Configuring the General Settings, page 8-31.
- For more information on configuring blocking devices, see Chapter 13, “Configuring Attack Response Controller for Blocking and Rate Limiting.”
- For the procedure for configuring SNMP, see Chapter 14, “Configuring SNMP.”

Configuration Sequence

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 target value ratings.

   Assign target value ratings to your network assets so that you can calculate the risk rating.
3. Create overrides to add actions based on the risk rating value.

   Assign a risk rating to each event action type.
4. Create filters.

   Assign filters to subtract actions based on the ID, IP addresses, and risk rating of the signature.
5. Create OS mappings.

   OS mappings are used for attack relevance rating in the calculation of risk rating for an alert.
6. Configure the general settings.

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

Working With Event Action Rules Policies

Use the **service event-action-rules** *name* command in service event action rules submode to create an event action rules policy. The values of this event action rules policy are the same as the default event action rules policy, rules0, until you edit them.

Or you can use the **copy event-action-rules** *source_destination* command in privileged EXEC mode to make a copy of an existing policy and then edit the values of the new policy as needed.

Use the **list event-action-rules-configurations** command in privileged EXEC mode to list the event action rules policies.

Use the **no service event-action-rules** *name* command in global configuration mode to delete an event action rules policy. Use the **default service event-action-rules** *name* command in global configuration mode to reset the event action rules policy to factory settings.

To create, copy, display, edit, and delete event action rules policies, follow these steps:

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

Step 2 Create an event action rules policy.

```
sensor# configure terminal
sensor(config)# service event-action-rules MyRules
sensor(config-eve)# exit
Apply Changes?[yes]: yes
sensor(config)# exit
sensor#
```

Step 3 Copy an existing event action rules policy to a new event action rules policy.

```
sensor# copy event-action-rules rules0 rules1
sensor#
```

Note You receive an error if the policy already exists or if there is not enough space available for the new policy.

Step 4 Accept the default event action rules policy values or edit the following parameters:

a. Add event action rules variables.

b. Configure event action rules overrides.

c. Configure event action rules filters.

d. Configure the event action rules general settings.

e. Configure the event action rules target value rating.

f. Configure the event action rules OS identification settings.

Step 5 To display a list of event action rules policies on the sensor.

```
sensor# list event-action-rules-configurations
Event Action Rules
   Instance   Size   Virtual Sensor
   rules0     255    vs0
   temp       707    N/A
   MyRules    255    N/A
   rules1     141    vs1
sensor#
```

Step 6 To delete an event action rules policy.

```
sensor(config)# no service event-action-rules MyRules
sensor(config)#
```

Note You cannot delete the default event action rules policy, rules0.

Step 7 Confirm the event action rules instance has been deleted.

```
sensor# list event-action-rules-configurations
Event Action Rules
   Instance   Size   Virtual Sensor
   rules0     112    vs0
   rules1     142    N/A
sensor#
```

Step 8 To reset an event action rules policy to factory settings.

```
sensor# configure terminal
sensor(config)# default service event-action-rules rules1
sensor(config)#
```

For More Information

- For the procedure for configuring event action rules variables, see Event Action Variables, page 8-9.
- For the procedure for configuring event action rules overrides, see Event Action Overrides, page 8-14.
- For the procedure for configuring event action rules filters, see Event Action Filters, page 8-17.

- For the procedure for configuring event action rules general settings, see General Settings, page 8-29.
- For the procedure for configuring event action rules target value ratings, see Target Value Ratings, page 8-11.
- For the procedure for configuring event action rules OS identifications, see OS Identifications, page 8-23.

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 IP address space of the engineering group. You could then use this variable to configure a filter that would ignore all Windows-based attacks for this group.

Adding, Editing, and Deleting Event Action Variables

Use the **variables** *variable_name* **address** *ip_address* command in service event action rules submode to create an event action variable. The IP address can be one address, a range, or ranges separated by a comma. Use the **no variables** *variable_name* command in service event action rules submode to delete an event action variable.

To add, delete, and edit 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 Add an event action rules variable.

```
sensor(config-eve)# 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 Verify the event action rules variable you added.

```
sensor(config-eve)# 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 To edit an event action rules variable, change the IP address.

```
sensor(config-eve)# variables variable1 address 10.89.130.191
```

Step 6 Verify the event action rules variable you edited.

```
sensor(config-eve)# show settings
   variables (min: 0, max: 256, current: 2)
   ----------------------------------------------
      variableName: variable1
      ----------------------------------------------
         address: 10.89.130.191 default: 0.0.0.0-255.255.255.255
      ----------------------------------------------
   ----------------------------------------------
```

Step 7 To delete an event action rules variable.

```
sensor(config-eve)# no variables variable1
```

Step 8 Verify the event action rules variable you deleted:

```
sensor(config-eve)# show settings
   variables (min: 0, max: 256, current: 0)
   ----------------------------------------------
   ----------------------------------------------
```

Step 9 Exit event action rules submode.

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

Step 10 Press **Enter** to apply your changes or enter **no** to discard them.

Target Value Ratings

This section describes what risk rating is and how to use it to configure the target value rating. This section contains the following topics:

Calculating the Risk Rating

A risk rating 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 in to account the value of the network asset being attacked (for example, a particular server), so it is configured on a per-signature basis (attack severity rating and signature fidelity rating) and on a per-server basis (target value rating). The risk rating is calculated from several components, some of which are configured, some collected, and some derived.

Note The risk rating is associated with alerts not signatures.

Risk ratings let you prioritize alerts that need your attention. These risk rating factors take in to 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 risk rating is reported in the evIdsAlert.

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

- Signature fidelity rating (SFR)—A weight associated with how well this signature might perform in the absence of specific knowledge of the target. The signature fidelity rating is configured per signature and indicates how accurately the signature detects the event or condition it describes.

  Signature fidelity rating 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 signature fidelity rating than a signature that is written with generic rules.

  Note The signature fidelity rating does not indicate how bad the detected event may be.

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

  The attack severity rating is derived from the alert severity parameter (informational, low, medium, or high) of the signature. The attack severity rating is configured per signature and indicates how dangerous the event detected is.

  Note The attack severity rating does not indicate how accurately the event is detected.

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

  Target value rating is a user-configurable value (zero, low, medium, high, or mission critical) 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 target value rating to the company web server that is higher than the target value rating you assign to a desktop node. In this example, attacks against the company web server have a higher risk rating than attacks against the desktop node. Target value rating is configured in the Event Action Rules policy.

- Attack relevance rating (ARR)—A weight associated with the relevancy of the targeted OS.

  Attack relevance rating is a derived value (relevant, unknown, or not relevant), which is determined at alert time. The relevant OSes are configured per signature.

- Promiscuous delta (PD)—A weight associated with the promiscuous delta.

  Promiscuous delta is in the range of 0 to 30 and is configured per signature.

  Note If the trigger packet is not inline, the promiscuous delta is subtracted from the rating.

- Watch list rating (WLR)—A weight associated with the CSA MC watch list in the range of 0 to 100 (CSA MC only uses the range 0 to 35).

  If the attacker for the alert is found on the watch list, the watch list rating for that attacker is added to the rating.

Figure 8-2 illustrates the risk rating formula:

Figure 8-2 Risk Rating Formula

$$RR = \frac{ASR * TVR * SFR}{10000} + ARR - PD + WLR$$

191016

Threat Rating

Threat rating is risk rating that has been lowered by event actions that have been taken. All event actions have a threat rating adjustment. The largest threat rating from all of the event actions taken is subtracted from the risk rating.

The event actions have the following threat ratings:

- Deny attacker inline—45
- Deny attacker victim pair inline—40
- Deny attacker service pair inline—40
- Deny connection inline—35
- Deny packet inline—35
- Modify packet inline—35
- Request block host—20
- Request block connection—20

- Reset TCP connection—20
- Request rate limit—20

Adding, Editing, and Deleting Target Value Ratings

You can assign a target value rating to your network assets. The target value rating is one of the factors used to calculate the risk rating value for each alert. You can assign different target value ratings to different targets. Events with a higher risk rating trigger more severe signature event actions.

Use the **target-value** {**zerovalue** | **low** | **medium** | **high** | **mission-critical**} **target-address** *ip_address* command in service event action rules submode to add target value ratings for your network assets. The default is medium. Use the **no target-value** [**zerovalue** | **low** | **medium** | **high** | **mission-critical**] command in service event action rules submode to delete target value ratings.

The following options apply:

- Target value rating setting:
  - **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.
- **target-address** *ip_address*—Range set of IP address(es).

To add, edit, and delete target value ratings 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 rules1
```

Step 3 Assign the target value rating to the network asset.

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

Step 4 Verify the target value rating you added.

```
sensor(config-eve)# 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-eve)#
```

Step 5 To edit a target value rating, change the target value rating setting of the asset.

```
sensor(config-eve)# target-value low target-address 10.89.130.108
```

Step 6 Verify the target value rating you edited.

```
sensor(config-eve)# show settings
----------------------------------------------
 target-value (min: 0, max: 5, current: 1)
```

```
 ----------------------------------------------
    target-value-setting: low
    target-address: 10.89.130.108 default: 0.0.0.0-255.255.255.255
 ----------------------------------------------
```

Step 7 To delete the target value rating.

```
sensor(config-eve)# no target-value low
```

Step 8 Verify the target value rating you deleted.

```
sensor(config-eve)# show settings
----------------------------------------------
 target-value (min: 0, max: 5, current: 0)
 ----------------------------------------------
 ----------------------------------------------
```

Step 9 Exit event action rules submode.

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

Step 10 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, page 8-14
- Adding, Editing, Enabling, and Disabling Event Action Overrides, page 8-14

Understanding Event Action Overrides

You can add an event action override to change the actions associated with an event based on the risk rating 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 risk rating range. If a signature event occurs and the risk rating 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 a risk rating of 85 or more to generate an SNMP trap, you can set the risk rating 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.

Adding, Editing, Enabling, and Disabling 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. Use the **no overrides** command in service event action rules submode to delete the parameters of event action overrides. Configure the override event actions, then the risk rating range, then enable or disable the override.

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, set the override-item-status to disabled for that entry.

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 risk rating values for this override item. The default is 0 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-eve)#
```

Step 3 Assign the action for the override:

- To deny packets from the source IP address of the attacker.

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

- To not transmit the single packet causing the alert.

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

- To not transmit packets on the specified TCP connection.

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

- To send TCP RST packets to terminate the connection.

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

- To request a block of the connection.

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

- To request a block of the attacker host.

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

- To log the packets from the attacker IP address.

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

- To log the packets from the victim IP address.

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

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

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

- To write an alert to Event Store.

```
sensor(config-eve)# overrides produce-alert
sensor(config-eve-ove)#
```

- To write verbose alerts to Event Store.

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

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

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

Step 4 Configure the risk rating for this override item.

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

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

Step 5 To enable or disable the use of this override item.

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

The default is enabled.

Step 6 Verify the settings.

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

Step 7 Edit the risk rating of an event action override.

```
sensor(config-eve)# overrides deny-attacker-inline
sensor(config-eve-ove)# risk-rating 95-100
```

Step 8 Verify the event action override that you edited.

```
sensor(config-eve-ove)# exit
sensor(config-eve)# show settings
-----------------------------------------------
 overrides (min: 0, max: 14, current: 1)
 -----------------------------------------------

       override-item-status: Enabled <defaulted>
       risk-rating-range: 95-100 default: 0-100
    -----------------------------------------------
```

Step 9 Delete the event action override.

```
sensor(config-eve)# no overrides deny-attacker-inline
sensor(config-eve-ove)#
```

Step 10 Verify the event action override that you deleted.

```
sensor(config-eve-ove)# exit
sensor(config-eve)# show settings
overrides (min: 0, max: 14, current: 1)
-----------------------------------------------
   action-to-add: deny-attacker-inline
   -----------------------------------------------
      override-item-status: Enabled <defaulted>
      risk-rating-range: 95 default: 0-100
   -----------------------------------------------
      override-item-status: Enabled <defaulted>
      risk-rating-range: 90-100 <defaulted>
   -----------------------------------------------
-----------------------------------------------
```

Step 11 Exit event action rules submode.

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

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

For More Information

For a detailed description of all the event actions, see Event Actions, page 8-4.

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.

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** {**edit | 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 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.
  - **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 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 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 Event Store, even if **produce-alert** is not selected.
  - **produce-alert** —Writes the event to 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.
- **os-relevance**—Event OS relevance for this filter.
  - **relevant**—The event is relevant to the target OS.
  - **not-relevant**—The event is not relevant to the target OS.
  - **unknown**—It is unknown whether the event is relevant to the target OS.
- **risk-rating-range**—Range of risk rating 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 {true | false}**—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 rules1
sensor(config-eve)#
```

Step 3 Create the filter name.

```
sensor(config-eve)# 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 Specify the values for this filter:

a. Specify the signature ID range.

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

The default is 900 to 65535.

b. Specify the subsignature ID range.

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

The default is 0 to 255.

c. Specify the attacker address range.

```
sensor(config-eve-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. Specify the victim address range.

```
sensor(config-eve-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. Specify the victim port range.

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

The default is 0 to 65535.

f. Specify the OS relevance.

```
sensor(config-eve-fil)# os-relevance relevant
```

The default is 0 to 100.

g. Specify the risk rating range.

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

The default is 0 to 100.

h. Specify the actions to remove.

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

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

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

The default is 100.

j. Specify the status of the filter to either disabled or enabled.

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

The default is enabled.

k. Specify the stop on match parameter.

```
sensor(config-eve-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.

l. Add any comments you want to use to explain this filter.

```
sensor(config-eve-fil)# user-comment NEW FILTER
```

Step 5 Verify the settings for the filter.

```
sensor(config-eve-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: NEW FILTER default:
      os-relevance: relevant default: relevant|not-relevant|unknown
   ----------------------------------------------
senor(config-eve-fil)#
```

Step 6 To edit an existing filter.

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

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

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

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

Step 9 Verify that you have moved the filters.

```
sensor(config-eve-fil)# exit
sensor(config-eve)# 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-eve)#
```

Step 10 To move a filter to the inactive list.

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

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

```
sensor(config-eve-fil)# exit
sensor(config-eve)# 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-eve)#
```

Step 12 Exit event action rules submode.

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

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

For More Information

- For a detailed description of all the event actions, see Event Actions, page 8-4.
- For the procedure for configuring event action variables, see Adding, Editing, and Deleting Event Action Variables, page 8-9.

OS Identifications

This section describes OS identifications and how to configure OS maps, and contains the following topics:

Understanding Passive OS Fingerprinting

Passive OS fingerprinting lets the sensor determine the OS that hosts are running. The sensor analyzes network traffic between hosts and stores the OS of these hosts with their IP addresses. The sensor inspects TCP SYN and SYNACK packets exchanged on the network to determine the OS type.

The sensor then uses the OS of the target host OS to determine the relevance of the attack to the victim by computing the attack relevance rating component of the risk rating. Based on the relevance of the attack, the sensor may alter the risk rating of the alert for the attack and/or the sensor may filter the alert for the attack. You can then use the risk rating to reduce the number of false positive alerts (a benefit in IDS mode) or definitively drop suspicious packets (a benefit in IPS mode). Passive OS fingerprinting also enhances the alert output by reporting the victim OS, the source of the OS identification, and the relevance to the victim OS in the alert.

Passive OS fingerprinting consists of three components:

- Passive OS learning

  Passive OS learning occurs as the sensor observes traffic on the network. Based on the characteristics of TCP SYN and SYNACK packets, the sensor makes a determination of the OS running on the host of the source IP address.

- User-configurable OS identification

  You can configure OS host mappings, which take precedence over learned OS mappings.

- Computation of attack relevance rating and risk rating

  The sensor uses OS information to determine the relevance of the attack signature to the targeted host. The attack relevance is the attack relevance rating component of the risk rating value for the attack alert. The sensor uses the OS type reported in the host posture information imported from the CSA MC to compute the attack relevance rating.

There are three sources of OS information. The sensor ranks the sources of OS information in the following order:

1. Configured OS mappings—OS mappings you enter.

   Configured OS mappings reside in the Event Action Rules policy and can apply to one or many virtual sensors.

Caution You can specify multiple operating systems for the same IP address. The last one in the list is the operating system that is matched.

2. Imported OS mappings—OS mappings imported from an external data source.

   Imported OS mappings are global and apply to all virtual sensors.

   Note Currently CSA MC is the only external data source.

3. Learned OS mappings—OS mappings observed by the sensor through the fingerprinting of TCP packets with the SYN control bit set.

   Learned OS mappings are local to the virtual sensor that sees the traffic.

When the sensor needs to determine the OS for a target IP address, it consults the configured OS mappings. If the target IP address is not in the configured OS mappings, the sensor looks in the imported OS mappings. If the target IP address is not in the imported OS mappings, the sensor looks in the learned OS mappings. If it cannot find it there, the sensor treats the OS of the target IP address as unknown.

Note Passive OS fingerprinting is enabled by default and the IPS contains a default vulnerable OS list for each signature.

Passive OS Fingerprinting Configuration Considerations

You do not have to configure passive OS fingerprinting for it to function. IPS provides a default vulnerable OS list for each signature and passive analysis is enabled by default.

You can configure the following aspects of passive OS fingerprinting:

- Define OS mappings

  We recommend configuring OS mappings to define the identity of the OS running on critical systems. It is best to configure OS mappings when the OS and IP address of the critical systems are unlikely to change.

- Limit attack relevance rating calculation to a specific IP address range

  This limits the attack relevance rating calculations to IP addresses on the protected network.

- Import OS mappings

  Importing OS mappings provides a mechanism for accelerating the learning rate and fidelity of the OS identifications made through passive analysis. If you have an external product interface, such as the CSA MC, you can import OS identifications from it.

- Define event action rules filters using the OS relevancy value of the target

  This provides a way to filter alerts solely on OS relevancy.

- Disable passive analysis

  Stops the sensor from learning new OS mappings.

- Edit signature vulnerable OS lists

  The vulnerable OS list specifies what OS types are vulnerable to each signature. The default, general-os, applies to all signatures that do not specify a vulnerable OS list.

Adding, Editing, Deleting, and Moving Configured OS Maps

Use the **os-identifications** command in the service event action rules submode to configure OS host mappings, which take precedence over learned OS mappings. You can add, edit, and delete configured OS maps. You can move them up and down in the list to change the order in which the sensor computes the attack relevance rating and risk rating for that particular IP address and OS type combination.

You can also move them up and down in the list to change the order in which the sensor resolves the OS associated with a particular IP address. Configured OS mappings allow for ranges, so for network 192.168.1.0/24 an administrator might define the following as displayed in Table 8-1:

Table 8-1 Example Configured OS Mapping

IP Address Range Set	OS
192.168.1.1	IOS
192.168.1.2-192.168.1.10,192.168.1.25	UNIX
192.168.1.1-192.168.1.255	Windows

More specific mappings should be at the beginning of the list. Overlap in the IP address range sets is allowed, but the entry closest to the beginning of the list takes precedence.

The following options apply:

- **calc-arr-for-ip-range**—Calculate the attack relevance rating for victims in this range. The value is <A.B.C.D>-<A.B.C.D>[,<A.B.C.D>-<A.B.C.D>], for example, 10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255).

  Note The second IP address in the range must be greater than or equal to the first IP address.

- **configured-os-map {edit | insert | move] name1[begin | end | inactive | before | after}**—Collection of administrator-defined mappings of IP addresses to OS IDs (configured OS mappings take precedence over imported and learned OS mappings).
- **ip**—The host IP address (or addresses) running the specified OS. The value is <A.B.C.D>-<A.B.C.D>[,<A.B.C.D>-<A.B.C.D>], for example, 10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255.

  Note The second IP address in the range must be greater than or equal to the first IP address.

- **os**—The OS type the host (or hosts) is running:
  - **general-os**—All OS types
  - **ios**—Variants of Cisco IOS
  - **mac-os**—Variants of the Apple System OS prior to OS X
  - **netware**—Netware
  - **other** —Any Other OS
  - **unix**—Variants of UNIX
  - **aix**—Variants of AIX
  - **bsd**—Variants of BSD

  - **hp-ux**—Variants of HP-UX
  - **irix**—Variants of IRIX
  - **linux**—Variants of Linux
  - **solaris**—Variants of Solaris
  - **windows**—Variants of Microsoft Windows
  - **windows-nt-2k-xp**—Variants of NT, 2000, and XP
  - **win-nt**—Specific variants of Windows NT
  - **unknown**—Unknown OS
- **default**—Sets the value back to the system default setting.
- **no**—Removes an entry or selection setting.
- **passive-traffic-analysis {enabled | disabled}**—Enables/disables passive OS fingerprinting analysis.

To configure OS mapping, 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 rules1
sensor(config-eve)#
```

Step 3 Create the OS map.

```
sensor(config-eve)# os-identification
sensor(config-eve-os)# configured-os-map insert name1 begin
sensor(config-eve-os-con)#
```

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

Step 4 Specify the values for this OS map:

a. Specify the host IP address.

```
sensor(config-eve-os-con)# ip 10.20.1.0-10.20.1.255
```

b. Specify the host OS type.

```
sensor(config-eve-os-con)# os unix
```

Caution You can specify multiple operating systems for the same IP address. The last one in the list is the operating system that is matched.

Step 5 Verify the settings for the OS map.

```
sensor(config-eve-os-con)# show settings
   NAME: name1
   ----------------------------------------------
      ip: 10.20.1.0-10.20.1.255 default:
      os: unix
   ----------------------------------------------
sensor(config-eve-os-con)#
```

Step 6 Specify the attack relevance rating range for the IP address.

```
sensor(config-eve-os-con)# exit
sensor(config-eve-os)# calc-arr-for-ip-range 10.89.30.108-10.89.30.191
```

Step 7 Enable passive OS fingerprinting.

```
sensor(config-eve-os)# passive-traffic-analysis enabled
```

Step 8 To edit an existing OS map.

```
sensor(config-eve-os)# configured-os-map edit name1
sensor(config-eve-os-con)#
```

Step 9 Edit the parameters (see Steps 4 through 7).

Step 10 To move an OS map up or down in the OS maps list.

```
sensor(config-eve-os-con)# exit
sensor(config-eve-os)# configured-os-map move name5 before name1
```

Step 11 Verify that you have moved the OS maps.

```
sensor(config-eve-os)# show settings
   os-identification
   ----------------------------------------------
      calc-arr-for-ip-range: 10.89.30.79 default: 0.0.0.0-255.255.255.255
      configured-os-map (ordered min: 0, max: 50, current: 2 - 2 active, 0 inactive)
      ----------------------------------------------
      ACTIVE list-contents
      ----------------------------------------------
         NAME: name2
         ----------------------------------------------
            ip: 10.89.30.79 default:
            os: aix
         ----------------------------------------------
         ----------------------------------------------
         NAME: name1
         ----------------------------------------------
            ip: 10.20.1.0-10.20.1.255 default:
            os: unix
         ----------------------------------------------
         ----------------------------------------------
      ----------------------------------------------
      passive-traffic-analysis: Enabled default: Enabled
   ----------------------------------------------
sensor(config-eve-os)#
```

Step 12 To move an OS map to the inactive list.

```
sensor(config-eve-os)# configured-os-map move name1 inactive
```

Step 13 Verify that the filter has been moved to the inactive list.

```
sensor(config-eve-os)# show settings
   os-identification
   ----------------------------------------------
      calc-arr-for-ip-range: 10.89.30.79 default: 0.0.0.0-255.255.255.255
      configured-os-map (ordered min: 0, max: 50, current: 2 - 1 active, 1 inactive)
      ----------------------------------------------
      ACTIVE list-contents
      ----------------------------------------------
         NAME: name2
         ----------------------------------------------
            ip: 10.89.30.79 default:
            os: aix
```

```
            ----------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
         INACTIVE list-contents
         ---------------------------------------------
            NAME: name1
            ----------------------------------------------
               ip: 10.20.1.0-10.20.1.255 default:
               os: unix
            ----------------------------------------------
            ----------------------------------------------
         ---------------------------------------------
         passive-traffic-analysis: Enabled default: Enabled
--MORE--#
```

Step 14 To delete an OS map.

```
sensor(config-eve-os)# no configured-os-map name2
```

Step 15 Verify the OS map has been deleted.

```
sensor(config-eve-os)# show settings
   os-identification
   ----------------------------------------------
      calc-arr-for-ip-range: 10.89.30.79 default: 0.0.0.0-255.255.255.255
      configured-os-map (ordered min: 0, max: 50, current: 1 - 0 active, 1 inactive)
      ---------------------------------------------
      INACTIVE list-contents
      ---------------------------------------------
         NAME: name1
         ----------------------------------------------
            ip: 10.20.1.0-10.20.1.255 default:
            os: unix
         ----------------------------------------------
         ----------------------------------------------
      ---------------------------------------------
      passive-traffic-analysis: Enabled default: Enabled
   ----------------------------------------------
sensor(config-eve-os)#
```

Step 16 Exit event action rules submode.

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

Step 17 Press **Enter** to apply your changes or enter **no** to discard them.

Displaying and Clearing OS Identifications

Use the **show os-identification** [*virtual-sensor*] **learned** [*ip-address*] command in EXEC mode to display OS IDs associated with IP addresses that were learned by the sensor through passive analysis.

Use the **clear os-identification** [*virtual-sensor*] **learned** [*ip-address*] command in EXEC mode to delete OS IDs associated with IP addresses that were learned by the sensor through passive analysis.

When you specify an IP address, only the OS identification for the specified IP address is displayed or cleared. If you specify a virtual sensor, only the OS identifications for the specified sensor is displayed or cleared. If you specify an IP address without a virtual sensor, the IP address is displayed or cleared on all virtual sensors.

The following options apply:

- *virtual-sensor*—(Optional) The learned addresses of the virtual sensor that should be displayed or cleared.
- *ip-address*—(Optional) The IP address to query or clear. The sensor displays or clears the OS ID mapped to the specified IP address.

To display and clear OS IDs, follow these steps:

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

Note An account with viewer privileges can display OS IDs.

Step 2 Display the learned OS IDs associated with a specific IP address.

```
sensor# show os-identification learned 10.1.1.12
Virtual Sensor vs0:
   10.1.1.12 windows
sensor# show os-identification learned
Virtual Sensor vs0:
   10.1.1.12 windows
Virtual Sensor vs1:
   10.1.0.1   unix
   10.1.0.2   windows
   10.1.0.3   windows
sensor#
```

Step 3 Clear the learned OS IDs for a specific IP address on all virtual sensors.

```
sensor# clear os-identification learned 10.1.1.12
```

Step 4 Verify that the OS IDs have been cleared.

```
sensor# show statistics os-identification
Statistics for Virtual Sensor vs0
   OS Identification
      Configured
      Imported
      Learned
Statistics for Virtual Sensor vs1
   OS Identification
      Configured
      Imported
      Learned
sensor#
```

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 in to 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 in to 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 nonalert-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 choose from the following summarization options:

- fire-all—Fires an alert each time the signature is triggered. If the threshold is set for summarization, 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 for that signature.
- summary—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 in to global summarization mode.

- global-summarization—Fires an alert for every summary interval. Signatures can be preconfigured for global summarization.
- fire-once—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-eve-gen)# global-metaevent-status {enabled | disabled}
```

The default is enabled.

Step 5 To enable or disable the summarizer.

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

The default is 1000.

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

```
sensor(config-eve-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-eve-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-eve-gen)# global-overrides-status {nabled | disabled}
```

The default is enabled.

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

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

The default is enabled.

Step 10 Check the settings for general submode.

```
sensor(config-eve-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-eve-gen)#
```

Step 11 Exit event action rules submode.

```
sensor(config-eve-gen)# exit
sensor(config-eve)# 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** [*virtual_sensor*] [**ip-address** *ip_address*] command 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 are 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.

The following options apply:

- *virtual_sensor*—(Optional) The virtual sensor whose denied attackers list should be cleared.
- ip_address—(Optional) The IP address to clear.

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 Delete the denied attackers list for a specific virtual sensor.

```
sensor# clear denied-attackers vs0
Warning: Executing this command will delete all addresses from the list of attackers being
denied by virtual sensor vs0.
Continue with clear? [yes]:
```

Step 6 Enter **yes** to clear the list.

Step 7 Remove a specific IP address from the denied attackers list for a specific virtual sensor.

```
sensor# clear denied-attackers vs0 ip-address 10.1.1.1
Warning: Executing this command will delete ip address 10.1.1.1 from the list of attackers
being denied by virtual sensor vs0.
Continue with clear? [yes]:
```

Step 8 Enter **yes** to clear the list.

Step 9 Verify that you have cleared the list.

You can use the **show statistics denied-attackers** or **show statistics virtual-sensor** command.

```
sensor# show statistics denied-attackers
Denied Attackers and hit count for each.
```

```
Denied Attackers and hit count for each.
Statistics for Virtual Sensor vs0
   Denied Attackers with percent denied and hit count for each.


   Denied Attackers with percent denied and hit count for each.


Statistics for Virtual Sensor vs1
   Denied Attackers with percent denied and hit count for each.


   Denied Attackers with percent denied and hit count for each.
sensor#

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.
```

Step 10 To clear only the statistics.

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

Step 11 Verify that you have cleared the statistics.

```
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 = 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

- 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

The filters are applied in the following order:

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

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 risk rating 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 risk rating 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 risk rating 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 Event Store, and contains the following topics:

- Displaying Events, page 8-36
- Clearing Events from Event Store, page 8-39

Displaying Events

Use the **show events** [{**alert** [informational] [low] [medium] [high] [**include-traits** ***traits***] [**exclude-traits** ***traits***] [**min-threat-rating** *min-rr*] [**max-threat-rating** *max-rr*] | **error** [warning] [error] [fatal] | **NAC** | **status**}] [*hh:mm:ss* [*month day* [*year*]] | **past** *hh:mm:ss*] command to display events from 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 press **Ctrl-C** to cancel the request.

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. Alert events are generated by Analysis Engine whenever a signature is triggered by network activity.

  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).
- **min-threat-rating**—Displays events with a threat rating above or equal to this value. The default is 0. The valid range is 0 to 100.
- **max-threat-rating**—Displays events with a threat rating below or equal to this value. The default is 100. The valid range is 0 to 100.
- **error**—Displays error events. Error events are generated by services when error conditions are encountered.

  If no level is selected (warning, error, or fatal), all error events are displayed.
- **NAC**—Displays ARC (block) requests.

Note ARC is formerly known as NAC. This name change has not been completely implemented throughout the IDM and CLI for IPS 6.0.

- **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 press **Ctrl-C** to exit.

To display events from 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 exception:
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.