イベント アクション ルールの設定

この章では、イベント アクション ルールを設定する方法について説明します。この章は、次の項で構成されています。

イベント アクション ルールについて

イベント アクション ルールは、センサーのイベント アクション処理コンポーネント用の設定のグループです。これらのルールは、イベントが発生したときにセンサーが実行するアクションを指定ます。

イベント アクション処理コンポーネントは、次の機能を担当します。

- リスク評価の計算
- イベント アクション オーバーライドの追加
- イベント アクションのフィルタリング
- 結果のイベント アクションの実行
- イベントの要約と集約
- 拒否された攻撃者リストの管理

Signature Event Action Processor

Signature Event Action Processor（SEAP）は、アラーム チャネル内のシグニチャ イベントから、SEAO、SEAF の処理を経由して SEAH で処理されるまでのデータ フローを調整します。これは次のコンポーネントから構成されます。

- アラーム チャネル

  Sensor App 検査パスからシグニチャ イベント処理へ向かうシグニチャ イベントと通信するエリアを表す単位。
- シグニチャ イベント アクション オーバーライド（SEAO）

  RR 値に基づいて、アクションを追加します。SEAO は、設定済みの RR しきい値の範囲に該当するすべてのシグニチャに適用されます。各 SEAO は独立しており、各アクション タイプには別個の値が設定されています。詳細については、P.6-9 の「リスク評価の計算」を参照してください。
- シグニチャ イベント アクション フィルタ（SEAF）

  シグニチャ イベントのシグニチャ ID、アドレス、および RR に基づいてアクションを削除します。SEAF へ入力するのは、SEAO によって追加される可能性のあるアクションを持つシグニチャ イベントです。

  （注）　SEAF が実行できるのはアクションの削除だけであり、新規アクションの追加はできません。

  SEAF には、次のパラメータが適用されます。
  - シグニチャ ID
  - サブシグニチャ ID
  - 攻撃者のアドレス
  - 攻撃者のポート
  - 被害先のアドレス
  - 被害先のポート
  - RR しきい値の範囲
  - 削除するアクション
  - シーケンス識別子（オプション）

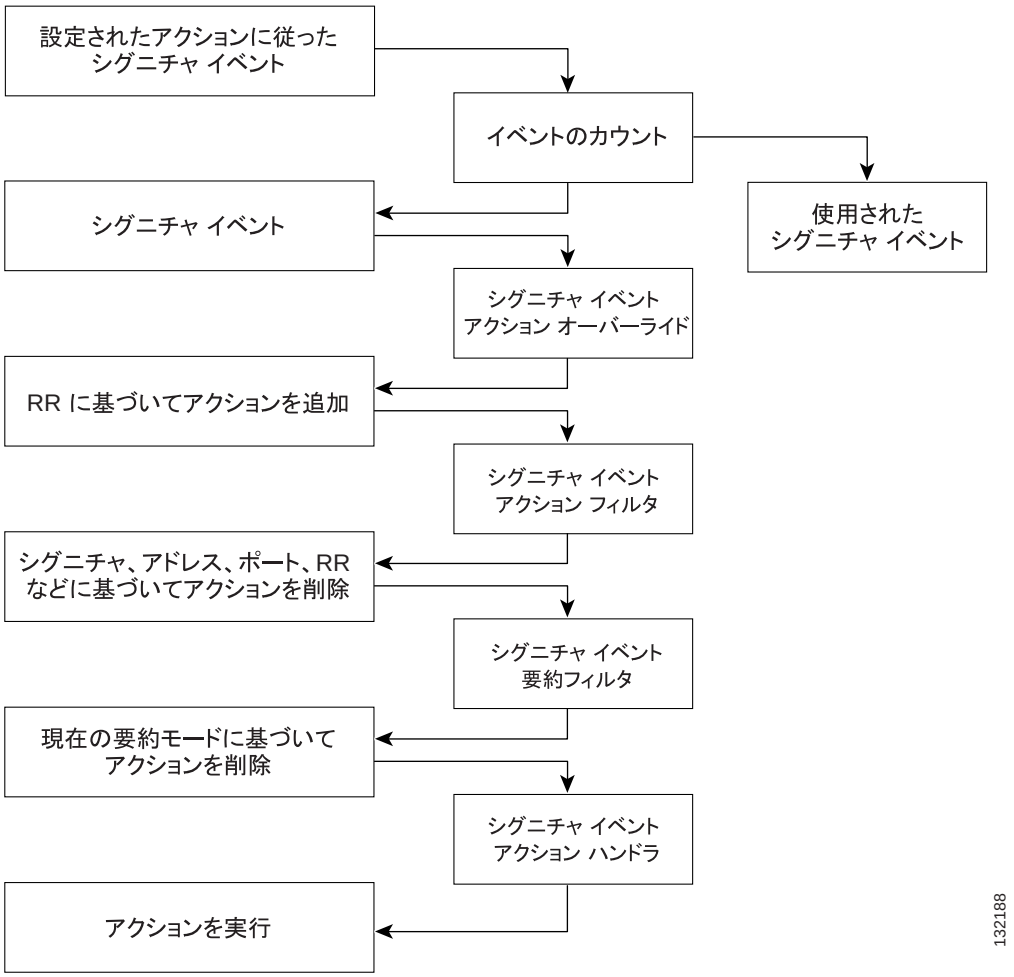
- ストップ ビットまたは継続ビット
- アクション フィルタ行をイネーブルにするビット

- シグニチャ イベント アクション ハンドラ（SEAH）

  要求されたアクションを実行します。SEAH から出力されるのは、実行中のアクションだけでなく、イベント ストアに書き込まれる <evIdsAlert> である可能性があります。

図 6-1 に、SEAP を通過するシグニチャ イベントの論理的なフローと、このイベントのアクションで実行される操作を示します。これは、アラーム チャネルで受信された設定済みアクションを持つシグニチャ イベントで開始され、シグニチャ イベントが SEAP の機能コンポーネントを通過するときに上から下へ流れます。

図 6-1　SEAP を通過するシグニチャ イベント



イベント アクション

表 6-1 は、イベント アクションについて説明しています。

表 6-1　イベント アクション

イベント アクション名	説明
Deny Attacker Inline	（インライン モードのみ）指定された期間、攻撃者アドレスから発信されたこのパケットおよび将来のパケットを送信しません。[1] **（注）** これは、拒否アクションの中で最も重大です。これは、単一の攻撃者アドレスからの現在および将来のパケットを拒否します。拒否された攻撃者のエントリをすべてクリアするには、**Monitoring** > **Denied Attackers** > **Clear List** をクリックします。これによって、これらのアドレスのネットワークへの再接続が許可されます。手順については、P.6-23 の「拒否された攻撃者リストのモニタリングとクリア」を参照してください。
Deny Attacker Service Pair Inline	（インライン モードのみ）指定された期間、攻撃者アドレスと被害先ポートのペアで、このパケットおよび将来のパケットを送信しません。
Deny Attacker Victim Pair Inline	（インライン モードのみ）指定された期間、攻撃者と被害先のアドレスのペアで、このパケットおよび将来のパケットを送信しません。 **（注）** 拒否アクションの場合、指定された期間と拒否された攻撃者の最大数を指定するには、**Configuration** > **Event Action Rules** > **General Settings** をクリックします。手順については、P.6-21 の「汎用設定」を参照してください。
Deny Connection Inline	（インライン モードのみ）TCP フローで、このパケットおよび将来のパケットを送信しません。
Deny Packet Inline	（インライン モードのみ）このパケットを送信しません。
Log Attacker Packets	攻撃者アドレスを含む IP ロギング パケットを開始します。 **（注）** このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
Log Pair Packets	攻撃者と被害先のアドレスのペアを含む IP ロギング パケットを開始します。 **（注）** このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
Log Victim Packets	被害先アドレスを含む IP ロギング パケットを開始します。

表 6-1　イベント アクション（続き）

イベント アクション名	説明
Modify Packet Inline	パケット データを変更して、エンドポイントでパケットがどう処理されるかに関してあいまいな部分を除去します。 （注）　Modify Packet Inline は、Add Event Action Filter または Add Event Action Override のオプションではありません。
Produce Alert	イベントをアラートとしてイベント ストアに書き込みます。
Produce Verbose Alert	違反パケットの符号化ダンプをアラートに組み込みます。 （注）　このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
Request Block Connection	この接続をブロックする要求を ARC に送信します。 （注）　ブロッキング デバイスは、このアクションを実装するよう設定されている必要があります。詳細については、第 10 章「ブロッキングとレート制限のための ARC の設定」を参照してください。
Request Block Host	この攻撃者ホストをブロックする要求を ARC に送信します。 （注）　ブロッキング デバイスは、このアクションを実装するよう設定されている必要があります。詳細については、第 10 章「ブロッキングとレート制限のための ARC の設定」を参照してください。 （注）　ブロック アクションの場合、ブロックの期間を設定するには、**Configuration > Event Action Rules > General Settings** をクリックします。手順については、P.6-21 の「汎用設定」を参照してください。
Request Rate Limit	レート制限を実行するレート制限要求を ARC に送信します。レート制限デバイスは、このアクションを実装するよう設定されている必要があります。詳細については、第 10 章「ブロッキングとレート制限のための ARC の設定」を参照してください。
Request SNMP Trap	SNMP 通知を実行する要求を NotificationApp に送信します。 （注）　このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。SNMP は、このアクションを実装するようセンサーで設定されている必要があります。詳細については、第 11 章「SNMP の設定」を参照してください。

表 6-1　イベント アクション（続き）

イベント アクション名	説明
Reset TCP Connection	TCP リセットを送信し、TCP フローを乗っ取って終了します。 **（注）** Reset TCP Connection は、単一の接続を分析する TCP シグニチャでのみ機能します。スイープやフラッドに対しては機能しません。

1. センサーは、システムによって拒否されている攻撃者のリストを保持します。拒否された攻撃者のリストからエントリを削除するには、攻撃者のリストを表示してリスト全体をクリアするか、タイマーで有効期限が切れるのを待ちます。タイマーは、エントリごとにスライドするタイマーです。したがって、攻撃者 A が拒否されているときに別の攻撃が発行されると、攻撃者 A のタイマーはリセットされ、そのタイマーの有効期限が切れるまで攻撃者 A は拒否された攻撃者リストに残ります。拒否された攻撃者のリストがいっぱいになり、新規エントリを追加することができない場合でも、パケットは拒否されます。

注意　シグニチャに対してアラートをイネーブルにした場合、Produce Alert アクションは自動にはなりません。イベント ストアでアラートを作成するには、Produce Alert を選択する必要があります。2 番目のアクションを追加する場合、イベント ストアにアラートを送信するには、Produce Alert を組み込む必要があります。また、イベント アクションを設定するたびに、新規リストが作成され、古いリストが置換されます。各シグニチャに必要なイベント アクションを必ずすべて組み込んでください。

イベント アクション ルールを設定するためのタスク リスト

IPS のイベント アクション ルール用コンポーネントを設定する場合は、次の手順を実行します。

1. イベント アクション フィルタで使用するすべての変数を作成します。
2. Target Value Rating（TVR; ターゲットの価値評価）を作成します。

   RR を計算できるように、ネットワーク資産に TVR を割り当てます。
3. RR 値に基づいてアクションを追加するためのオーバーライドを作成します。

   各イベント アクション タイプに RR を割り当てます。
4. フィルタを作成します。

   シグニチャの ID、IP アドレス、および RR に基づいてアクションを削除するためのフィルタを割り当てます。
5. 汎用設定を行います。

   Summarizer、Meta Event Generator を使用するかどうかを指定します。または、拒否された攻撃者のパラメータを設定します。

イベント アクション変数

この項では、イベント アクション変数について説明します。取り上げる事項は次のとおりです。

イベント アクション変数について

イベント アクション変数を作成して、その変数をイベント アクション フィルタ内で使用することができます。複数のフィルタで同じ値を使用する場合、変数を使用します。変数の値を変更すると、その変数を使用しているフィルタは新しい値で更新されます。

（注）　変数の前にドル記号（$）を付けて、文字列ではなく変数を使用していることを示す必要があります。

シグニチャ システムに必要なため、削除できない変数もあります。変数が保護されている場合は、編集することはできません。保護された変数を削除しようとすると、エラー メッセージが表示されます。編集できる変数は一度に 1 つだけです。

IP アドレスを設定する場合は、完全な IP アドレス、範囲、または範囲のセットを指定します。次の例を参考にしてください。

- 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

ワンポイント・アドバイス　たとえば、社内の技術グループに対応する IP アドレス空間があるとします。グループ内には Windows システムがなく、Windows ベースの攻撃を心配する必要はありません。このような場合に、この技術グループの IP アドレス空間として変数を設定します。その後、この変数を使用して、このグループに対するすべての Windows ベースの攻撃を無視するフィルタを設定できます。

イベント アクション変数の設定

イベント アクション変数を設定するには、サービス イベント アクション ルール サブモードで、**variables** *variable_name* **address** *ip_address* コマンドを使用します。IP アドレスには、1 つのアドレス、1 つの範囲、またはカンマで区切った複数の範囲を指定できます。

イベント アクション変数を設定するには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　イベント アクション ルール サブモードに入ります。

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

ステップ 3　変数を作成します。

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

address の有効な値は、A.B.C.D-A.B.C.D [,A.B.C.D-A.B.C.D] です。

ステップ 4　作成した変数を確認します。

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

ステップ 5　イベント アクション ルール サブモードを終了します。

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

ステップ 6　変更を適用する場合は **Enter** キーを押し、変更を廃棄する場合は **no** と入力します。

ターゲットの価値評価

この項では、リスク評価（RR）、および RR を使用したターゲットの価値評価（TVR）の設定方法について説明します。この項では、次のトピックについて説明します。

リスク評価の計算

RR は、ネットワーク上の特定のイベントに関連付けられたリスクを、0 から 100 の間の数値で表した評価です。計算では、攻撃されているネットワーク資産（特定のサーバなど）の価値も考慮されるため、RR はシグニチャ単位（ASR および SFR）およびサーバ単位（TVR）で設定されます。

RR を使用すると、注意の必要なアラートの優先順位を高くすることができます。このような RR の要素としては、成功した場合の攻撃の重大度、シグニチャの忠実度、およびターゲット ホストの全体の価値が考慮に入れられています。RR は evIdsAlert で報告されます。

特定のイベントの RR の計算には、次の値が使用されます。

- **Attack Severity Rating（ASR; 攻撃の重大度評価）**：脆弱性の不正利用が成功した場合の重大度に関連付ける重み値。

  ASR は、シグニチャのアラート重大度パラメータから取得されます。
- **Signiture Fidelity Rating（SFR; シグニチャの忠実度評価）**：対象とする特定の情報がない場合にこのシグニチャをどの程度忠実に実行するかに関連付ける重み値。

  SFR は、シグニチャ作成者によってシグニチャごとに計算されます。シグニチャ作成者は、資格を与える情報がターゲットにない場合のシグニチャの精度について、基本的な信頼度を定義します。この信頼度は、分析中のパケットの送達が許可された場合に、検出された動作がどの程度確実にターゲット プラットフォームに目的とする効果を与えるかを表します。たとえば、きわめて具体的なルール（特定の正規表現など）で記述されたシグニチャは、汎用的なルールで記述されたシグニチャより高い SFR を持ちます。
- **Target Value Rating (TVR; ターゲットの価値評価）**：ターゲットの価値に関連付ける重み値。

  TVR は、ネットワーク資産（IP アドレスを経由する）の重要性を示す、ユーザが設定可能な値です。価値の高い企業リソースにはより厳しいセキュリティ ポリシーを作成し、そうでないリソースにはある程度緩やかなポリシーを作成できます。たとえば、企業の Web サーバには、デスクトップ ノードに割り当てる TVR より高い TVR を割り当てることができます。この場合、企業の Web サーバに対する攻撃は、デスクトップ ノードに対する攻撃よりも高い RR を持ちます。

（注）　RR は、ASR、SFR、TVR、およびオプションの Promiscuous Delta（PD; 混合デルタ）から計算されます。

ターゲットの価値評価の設定

ネットワーク資産に TVR を割り当てることができます。TVR は、各アラートの RR 値の計算に使用される要素の 1 つです。ターゲットごとに異なる TVR を割り当てることができます。RR の高いイベントほど、より厳しいシグニチャ イベント アクションをトリガーします。

ネットワーク資産の TVR を設定するには、サービス イベント アクション ルール サブモードで、**target-value target-value-setting** [**zerovalue** | **low** | **medium** | **high** | **mission-critical**] **target-address** *ip_address* コマンドを使用します。デフォルトは medium です。

次のオプションが適用されます。

- **target-address** *ip_address*：IP アドレスの範囲のセット（複数も可）。
- **target-value-setting**：次のいずれかを選択します。
  - **zerovalue**：このターゲットに値はありません。
  - **low**：このターゲットの低い値。
  - **medium**：このターゲットの通常の値。
  - **high**：このターゲットの高い値。
  - **mission-critical**：このターゲットの最高の値。

ネットワーク資産の TVR を設定するには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　イベント アクション ルール サブモードに入ります。

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

ステップ 3　ネットワーク資産に TVR を割り当てます。

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

ステップ 4　TVR の設定内容を確認します。

```
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)#
```

ステップ 5　イベント アクション ルール サブモードを終了します。

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

ステップ 6　変更を適用する場合は **Enter** キーを押し、変更を廃棄する場合は **no** と入力します。

イベント アクション オーバーライド

この項では、イベント アクション オーバーライドについて説明します。取り上げる事項は次のとおりです。

イベント アクション オーバーライドについて

イベント アクション オーバーライドを追加すると、特定のイベントに関連付けられたアクションを、そのイベントの RR に基づいて変更することができます。イベント アクション オーバーライドを使用すると、各シグニチャを個別に設定しなくても、グローバルにイベント アクションを追加することができます。各イベント アクションには、一定範囲の RR が関連付けられています。シグニチャ イベントが発生し、そのイベントの RR が特定のイベント アクションの範囲内にある場合は、そのアクションがイベントに追加されます。たとえば、RR が 85 以上のすべてのイベントで SNMP トラップを生成する場合は、Request SNMP Trap の RR 範囲を 85 ～ 100 に設定します。アクションのオーバーライドを使用しない場合は、イベント アクション オーバーライド コンポーネント全体をディセーブルにできます。

イベント アクション オーバーライドの設定

イベント アクション オーバーライドのパラメータを設定するには、サービス イベント アクション ルール サブモードで、**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**] コマンドを使用します。

すべてのイベント アクションの説明については、P.6-4 の「イベント アクション」を参照してください。

次のオプションが適用されます。

- **no**：エントリまたは選択設定を削除します。
- **override-item-status** [**enabled** | **disabled**]：このオーバーライド項目の使用をイネーブルまたはディセーブルにします。デフォルトは enabled です。
- **risk-rating-range**：このオーバーライド項目の RR 値の範囲。デフォルトは 1 ～ 100 です。
- **show**：システム設定または履歴情報（あるいはその両方）を表示します。

イベント アクション オーバーライドを追加するには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　イベント アクション ルール サブモードに入ります。

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

ステップ 3　オーバーライドでのパケットの処理方法を設定するには、次の手順を実行します。

a. 攻撃者の送信元 IP アドレスからのパケットを拒否する場合

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

b. アラートの原因となっている単一のパケットを送信しないようにする場合

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

c. 指定された TCP 接続のパケットを送信しないようにする場合

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

d. TCP RST パケットを送信して接続を終了する場合

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

ステップ 4　ブロックを要求するオーバーライドを設定するには、次の手順を実行します。

a. 接続のブロックを要求する場合

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

b. 攻撃者のホストのブロックを要求する場合

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

ステップ 5　オーバーライドでパケットをログに記録するには、次の手順を実行します。

a. 攻撃者の IP アドレスからのパケットをログに記録する場合

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

b. 被害先の IP アドレスからのパケットをログに記録する場合

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

c. 攻撃者と被害先の IP アドレスからのパケットをログに記録する場合

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

ステップ 6　イベント ストアにアラートを書き込むには、次の手順を実行します。

a.　イベント ストアにアラートを書き込む場合

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

b.　イベント ストアに詳細なアラートを書き込む場合

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

c.　SNMP トラップを要求するイベントをイベント ストアに書き込む場合

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

ステップ 7　このオーバーライド項目の RR を設定するには、次の操作を実行します。

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

（注）　デフォルトの RR 範囲は 0 ～ 100 です。85 ～ 100 などの別の範囲に設定してください。

ステップ 8　このオーバーライド項目をイネーブルまたはディセーブルにするには、次のコマンドを実行します。

```
sensor(config-rul-ove)# override-item-status [enabled | disabled]
```

デフォルトは enabled です。

ステップ 9　設定を確認します。

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

ステップ 10　イベント アクション ルール サブモードを終了します。

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

ステップ 11　変更を適用する場合は **Enter** キーを押し、変更を廃棄する場合は **no** と入力します。

イベント アクション フィルタ

この項では、イベント アクション フィルタについて説明します。取り上げる事項は次のとおりです。

イベント アクション フィルタについて

イベント アクション フィルタは、順に並べられたリストで処理されます。フィルタはリスト内で上下に移動することができます。

センサーは、フィルタの使用により、イベントに応答してすべてのアクションを実行せずに特定のアクションを実行したり、イベント全体を削除したりすることができます。フィルタは、イベントからアクションを削除することによって機能します。イベントからすべてのアクションを削除するフィルタは、効果的にイベントを消滅させます。

（注）　スイープ　シグニチャをフィルタリングする場合は、宛先アドレスをフィルタリングしないことをお勧めします。複数の宛先アドレスが存在する場合は、最後のアドレスだけがフィルタとの照合に使用されます。

イベント アクション フィルタの設定

イベント アクション フィルタを設定すると、イベントから特定のアクションを削除したり、イベント全体を廃棄したりすることにより、センサーがそれ以上処理しないようにすることができます。フィルタでは、アドレスをグループ化するために定義したイベント アクション変数を使用できます。イベント アクション変数の設定手順については、P.6-8 の「イベント アクション変数の設定」を参照してください。

（注）　変数の前にドル（$）記号を付けて、文字列ではなく変数を使用していることを示す必要があります。「$」を付けないと、`Bad source and destination` エラーが生じます。

イベント アクション フィルタを設定するには、サービス イベント アクション ルール サブモードで、**filters [edit | insert | move] name1 [begin | end | inactive | before | after]** コマンドを使用します。

次のオプションが適用されます。

- **actions-to-remove**：このフィルタ項目で削除するイベント アクション。
  - **deny-attacker-inline**：（インライン モードのみ）指定された期間、攻撃者アドレスから、このパケットおよび将来のパケットを送信しません。
  - **deny-attacker-service-pair-inline**：（インラインのみ）指定された期間、攻撃者アドレスと被害先のポートのペアで、このパケットおよび将来のパケットを送信しません。
  - **deny-attacker-victim-pair-inline**：（インラインのみ）指定された期間、攻撃者 / 被害先のアドレスのペアで、このパケットおよび将来のパケットを送信しません。
  - **deny-connection-inline**：（インライン モードのみ）TCP フローで、このパケットおよび将来のパケットを送信しません。
  - **deny-packet-inline**：（インライン モードのみ）このパケットを送信しません。

- **log-attacker-packets**：攻撃者のアドレスを含むパケットの IP ロギングを開始します。このアクションを実行すると、**produce-alert** が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
- **log-pair-packets**：攻撃者と被害先のアドレスのペアを含むパケットの IP ロギングを開始します。このアクションを実行すると、**produce-alert** が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
- **log-victim-packets**：被害先のアドレスを含むパケットの IP ロギングを開始します。このアクションを実行すると、**produce-alert** が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
- **produce-alert**：イベントをアラートとしてイベント ストアに書き込みます。
- **produce-verbose-alert**：違反パケットの符号化ダンプ（切り捨てられている可能性があります）をアラートに組み込みます。このアクションを実行すると、**produce-alert** が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
- **request-block-connection**：この接続をブロックする要求を ARC に送信します。ブロッキング デバイスは、このアクションを実装するよう設定されている必要があります。
- **request-block-host**：この攻撃者ホストをブロックする要求を ARC に送信します。ブロッキング デバイスは、このアクションを実装するよう設定されている必要があります。
- **request-rate-limit**：レート制限を実行するレート制限要求を ARC に送信します。レート制限デバイスは、このアクションを実装するよう設定されている必要があります。
- **request-snmp-trap**：SNMP 通知を実行する要求をセンサーの通知アプリケーション コンポーネントに送信します。このアクションを実行すると、**produce-alert** が選択されていない場合でも、イベント ストアにアラートが書き込まれます。SNMP は、このアクションを実装するようセンサーで設定されている必要があります。
- **reset-tcp-connection**：TCP リセットを送信し、TCP フローを乗っ取って終了します。**Reset TCP Connection** は、単一の接続を分析する TCP シグニチャでのみ機能します。スイープやフラッドに対しては機能しません。
- **modify-packet-inline**：パケット データを変更して、エンドポイントでパケットがどう処理されるかに関してあいまいな部分を除去します。

- **attacker-address-range**：この項目に対応する攻撃者アドレスの範囲のセット（10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255 など）。
- **attacker-port-range**：この項目に対応する攻撃者ポートの範囲のセット（147-147,8000-10000 など）。
- **default**：値をシステム デフォルト設定に戻します。
- **deny-attacker-percentage**：攻撃者拒否機能で拒否するパケットの比率。有効な範囲は 1 ～ 100 です。デフォルトは 100 です。
- **filter-item-status** [**enabled** | **disabled**]：このフィルタ項目の使用をイネーブルまたはディセーブルにします。
- **no**：エントリまたは選択設定を削除します。
- **risk-rating-range**：このフィルタ項目の RR の範囲。
- **signature-id-range**：この項目に対応するシグニチャ ID の範囲のセット（1000-2000,3000-3000 など）。
- **stop-on-match**：このフィルタ項目に一致した場合に、フィルタの評価を継続するか中止するかの指定。
- **subsignature-id-range**：この項目に対応するサブシグニチャ ID の範囲のセット（0-2,5-5 など）。
- **user-comment**：このフィルタ項目に関するコメントを追加できます。
- **victim-address-range**：この項目に対応する被害先アドレスの範囲のセット（10.20.1.0-10.20.1.255,10.20.5.0-10.20.5.255 など）。
- **victim-port-range**：この項目に対応する被害先ポートの範囲のセット（147-147,8000-10000 など）。

イベント アクション フィルタを設定するには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　イベント アクション ルール サブモードに入ります。

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

ステップ 3　フィルタ名を作成します。

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

複数のイベント アクション フィルタに名前を付ける場合は、*name1*、*name2* の順に使用し、以下同様に続きます。フィルタの挿入場所の指定には、**begin | end | inactive | before | after** キーワードを使用します。

ステップ 4　フィルタの値を設定します。

a. シグニチャ ID の範囲を設定します。

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

デフォルトは 900 ～ 65535 です。

b. サブシグニチャ ID の範囲を設定します。

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

デフォルトは 0 ～ 255 です。

c. 攻撃者アドレスの範囲を設定します。

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

デフォルトは 0.0.0.0 ～ 255.255.255.255 です。

d. 被害先アドレスの範囲を指定します。

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

デフォルトは 0.0.0.0 ～ 255.255.255.255 です。

e. 被害先ポートの範囲を指定します。

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

デフォルトは 0 ～ 65535 です。

f. リスク評価の範囲を指定します。

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

デフォルトは 0 ～ 100 です。

g. 削除するアクションを設定します。

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

h. 拒否アクションのフィルタリングを行う場合は、拒否アクションの比率を設定します。

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

デフォルトは 100 です。

i. フィルタのステータスをディセーブルまたはイネーブルに設定します。

```
sensor(config-rul-fil)# filter-item-status [enabled | disabled]
```

デフォルトは enabled です。

j. stop-on-match パラメータを設定します。

```
sensor(config-rul-fil)# stop-on-match [true | false]
```

true を指定すると、この項目に一致した場合にセンサーは処理を中止します。**false** を指定すると、この項目に一致してもセンサーは処理を継続します。

k. フィルタを説明する任意のコメントを追加します。

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

ステップ 5　フィルタの設定を確認します。

```
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)#
```

ステップ 6　既存のフィルタを編集するには、次のコマンドを実行します。

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

ステップ 7　パラメータを編集します（ステップ 4a ～ 4k を参照）。

ステップ 8　フィルタ リストでフィルタを上下に移動するには、次のコマンドを実行します。

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

ステップ 9　フィルタが移動したことを確認します。

```
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)#
```

ステップ 10　フィルタを非アクティブ リストに移動するには、次のコマンドを実行します。

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

ステップ 11 フィルタが非アクティブ リストに移動したことを確認します。

```
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)#
```

ステップ 12 イベント アクション ルール サブモードを終了します。

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

ステップ 13 変更を適用する場合は **Enter** キーを押し、変更を廃棄する場合は **no** と入力します。

汎用設定

この項では、汎用設定について説明します。取り上げる事項は次のとおりです。

汎用設定について

イベント アクション ルールに適用する汎用設定を指定できます。汎用設定には、Summarizer や Meta Event Generator を使用するかどうかなどがあります。Summarizer は、イベントを 1 つのアラートにグループ化することによって、センサーが送出するアラートの数を減らします。Meta Event Generator は、コンポーネント イベントを処理します。これにより、センサーは一連のイベントにまたがって発生する不審なアクティビティを監視することができます。

攻撃者を拒否する期間、拒否された攻撃者の最大数、ブロックを続ける期間を設定できます。

イベント アクションの要約について

要約は、複数のイベントを 1 つのアラートにまとめる基本的な集約を実行することにより、センサーから送出されるアラートの量を減らします。各シグニチャに対して特別なパラメータが指定され、それぞれアラートの処理に影響を与えます。各シグニチャは、最適な通常の動作を反映したデフォルトの設定で作成されます。ただし、各シグニチャを調整して、各エンジン タイプの制限内でこのデフォルトの動作を変更できます。

要約されていない各シグニチャ イベントでは、アラートを生成しないアクション（拒否、ブロック、TCP リセット）はフィルタを通過します。このような要約されたアラートでは、アラートを生成するアクションは実行されません。代わりに、そのようなアクションは 1 つのサマリー アラートに適用され、その後フィルタにかけられます。

アラートを生成するその他のアクションのいずれかを選択し、そのアクションをフィルタリングで排除しない場合は、Produce Alert を選択していなくてもアラートが生成されます。アラートが生成されないようするには、アラートを生成するすべてのアクションがフィルタリングによって排除されるようにする必要があります。

要約とイベント アクションは、Meta エンジンがコンポーネント イベントを処理した後に処理されます。これにより、センサーは一連のイベントにまたがって発生する不審なアクティビティを監視することができます。

イベント アクションの集約について

基本集約機能には、2 種類の動作モードがあります。簡易モードでは、いくつのヒットがあるとアラートが送信されるかを示すしきい値をシグニチャに設定します。より高度なモードでは、間隔カウントを行います。このモードでは、センサーは 1 秒あたりのヒット数を追跡し、そのしきい値に達したときのみアラートを送信します。この例では、*ヒット*とはイベントを表すために使用する用語で、基本的にはアラートのことです。ただし、アラートは、ヒットのしきい値を超えるまではセンサーから送出されません。

次の要約オプションを選択できます。

- **Fire All**：Fire All モードでは、シグニチャがトリガーされるたびにアラートが発生します。要約にしきい値が設定されている場合は、要約が発生するまでは実行のたびにアラートが発生します。要約が開始されると、各アドレス セットで要約の間隔ごとに 1 つのアラートのみが発生します。異なるアドレス セットのアラートは、すべて表示されるか、個別に要約されます。該当のシグニチャで一定期間アラートが発生しないと、シグニチャは Fire All モードに戻ります。
- **Summary**：Summary　モードでは、シグニチャが最初にトリガーされるとアラートが発生します。その後は、サマリー間隔の期間ごとにそのシグニチャの追加のアラートが要約されます。各アドレス セットで要約の間隔ごとに 1 つのアラートのみが発生します。グローバル サマリーのしきい値に達すると、シグニチャは Global Summarization モードに入ります。
- **Global Summarization**：Global Summarization モードでは、サマリー間隔ごとに 1 つのアラートが発生します。シグニチャにグローバル サマリーを事前設定しておくこともできます。
- **Fire Once**：Fire Once モードでは、アドレス セットごとに 1 つのアラートが発生します。このモードを Global Summarization モードにアップグレードすることができます。

インラインの攻撃者拒否イベント アクションについて

インラインの攻撃者拒否イベント アクションの一定の特性を設定することができます。インラインで攻撃者拒否を実行する時間（秒単位）を設定したり、システム内で同時に拒否する攻撃者の数を制限したりすることができます。

汎用設定

イベント アクション ルールの汎用設定を行うには、サービス イベント アクション ルール サブモードで次の各コマンドを使用します。

- **global-block-timeout**：ホストまたは接続をブロックする時間（分単位）。

  有効な範囲は 0 ～ 10000000 です。デフォルトは 30 分です。
- **global-deny-timeout**：インラインで攻撃者を拒否する時間（秒単位）。

  有効な範囲は 1 ～ 518400 です。デフォルトは 3600 です。
- **global-filters-status** [**enabled | disabled]**：フィルタの使用をイネーブルまたはディセーブルにします。

  デフォルトは enabled です。
- **global-metaevent-status [enabled | disabled]**：Meta Event Generator の使用をイネーブルまたはディセーブルにします。

  デフォルトは enabled です。
- **global-overrides-status** [**enabled | disabled]**：オーバーライドの使用をイネーブルまたはディセーブルにします。

  デフォルトは enabled です。
- **global-summarization-status [enabled | disabled]**：Summarizer の使用をイネーブルまたはディセーブルにします。

  デフォルトは enabled です。
- **max-denied-attackers**：システム内で同時に拒否できる攻撃者数を制限します。

  有効な範囲は 1 ～ 100000000 です。デフォルトは 10000 です。

イベント アクションの汎用設定を行うには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　イベント アクション ルール サブモードに入ります。

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

ステップ 3　汎用サブモードに入ります。

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

ステップ 4　Meta Event Generator をイネーブルまたはディセーブルにするには、次のコマンドを実行します。

```
sensor(config-rul-gen)# global-metaevent-status [enabled | disabled]
```

デフォルトは enabled です。

ステップ 5　Summarizer をイネーブルまたはディセーブルにするには、次のコマンドを実行します。

```
sensor(config-rul-gen)# global-summarization-status [enabled | disabled]
```

デフォルトは enabled です。

ステップ 6　インラインの拒否攻撃者イベント アクションを設定するには、次の操作を実行します。

a.　システム内で同時に拒否される攻撃者数を制限するには、次のコマンドを実行します。

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

デフォルトは 1000 です。

b.　システムで攻撃者を拒否する時間（秒単位）を設定するには、次のコマンドを実行します。

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

デフォルトは 3600 秒です。

ステップ 7　ホストまたは接続をブロックする時間（分単位）を設定するには、次のコマンドを実行します。

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

デフォルトは 30 分です。

ステップ 8　設定したすべてのオーバーライドをイネーブルまたはディセーブルにするには、次のコマンドを実行します。

```
sensor(config-rul-gen)# global-overrides-status [enabled | disabled]
```

デフォルトは enabled です。

ステップ 9　設定したすべてのフィルタをイネーブルまたはディセーブルにするには、次のコマンドを実行します。

```
sensor(config-rul-gen)# global-filters-status [enabled | disabled]
```

デフォルトは enabled です。

ステップ 10 汎用サブモードの設定を確認します。

```
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)#
```

ステップ 11 イベント アクション ルール サブモードを終了します。

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

ステップ 12 変更を適用する場合は **Enter** キーを押し、変更を廃棄する場合は **no** と入力します。

拒否された攻撃者リストのモニタリングとクリア

拒否された攻撃者のリストを表示するには、**show statistics denied-attackers** コマンドを使用します。拒否された攻撃者のリストを削除して仮想センサーの統計情報をクリアするには、サービス イベント アクション ルール サブモードで、**clear denied-attackers** コマンドを使用します。

センサーがインライン モードで動作するよう設定されている場合、トラフィックはセンサーを通過します。シグニチャを設定すると、インライン モードでパケット、接続、および攻撃者を拒否することができます。これは、センサーが単一パケット、接続、および特定の攻撃者を検出したときにそれらを拒否する、つまり送信しないことを意味します。

シグニチャが起動すると、攻撃者は拒否され、リストに入れられます。センサー管理の一環として、リストの削除やリスト内の統計情報のクリアを実行することもできます。

拒否された攻撃者リストの表示、リストの削除、および統計情報のクリアを実行するには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　拒否された IP アドレスのリストを表示します。

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

今回 2 つの IP アドレスが拒否されたことが統計情報に示されています。

ステップ 3　拒否された攻撃者のリストを削除します。

```
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]:
```

ステップ 4　**yes** を入力してリストをクリアします。

ステップ 5　リストをクリアしたことを確認します。

```
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.
```

`Denied Attackers and hit count for each` カテゴリの情報がなくなっています。

ステップ 6　統計情報だけをクリアするには、次の手順を実行します。

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

ステップ 7　統計情報をクリアしたことを確認します。

```
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
```

`Number of Active Denied Attackers` と `Number of exec Clear commands during uptime` のカテゴリ以外の統計情報がクリアされています。リストがクリアされたかどうかを認識していることが重要です。

イベント アクション ルールの例

次の例は、イベント アクション ルールの個々のコンポーネントが相互に影響し合い、どのような動作になるかを示しています。

例 1 のリスク評価範囲

- **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

例 1 のイベント アクション フィルタ

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

例 1 の結果

SIG 2004 が検出された場合：

- 攻撃者アドレスが 30.1.1.1 であるか、または被害先のアドレスが 20.1.1.1 である場合、イベントは消滅します（ALL アクションは削除されます）。

攻撃者アドレスが 30.1.1.1 以外で、被害先アドレスが 20.1.1.1 以外である場合：

- RR が 50 の場合、イベント アクション オーバーライド コンポーネントによって Produce Alert と Request SNMP Trap が追加されますが、Produce Alert はイベント アクション フィルタによって削除されます。ただし、Request SNMP Trap は <evIdsAlert> に依存しているため、イベント アクション ポリシーによってアラート アクションが強制されます。
- RR が 89 の場合は、イベント アクション オーバーライド コンポーネントによって Request SNMP Trap と Request Block Connection が追加されます。ただし、Request Block Connection はイベント アクション フィルタによって削除されます。

- RR が 96 の場合は、イベント アクション オーバーライド コンポーネントにより、Deny Attacker Inline と Request Block Connection を除くすべてのアクションが追加され、イベント アクション フィルタによって削除されるアクションはありません。フィルタ アクションが NONE に指定された 3 番目のフィルタ行はオプションですが、このタイプのフィルタを定義するより明確な方法として示されています。

イベントのモニタリング

この項では、イベント ストアからイベントを表示およびクリアする方法について説明します。取り上げる事項は次のとおりです。

- イベントの表示（P.6-26）
- イベント ストアからのイベントのクリア（P.6-29）

イベントの表示

イベント ストアからイベントを表示するには、**show events** [{[**alert** [informational] [low] [medium] [high] [**include-traits** ***traits***] [**exclude-traits** ***traits***]] | **error** [warning] [error] [fatal] | **log** | **NAC** | **status**}] [*hh:mm:ss* [*month day* [*year*]] | **past** *hh:mm:ss*] コマンドを使用します。

開始時刻から、イベントが表示されます。開始時刻を指定しない場合は、現在時刻から、イベントが表示されます。イベント タイプを指定しない場合は、すべてのイベントが表示されます。

（注）　イベントは、Ctrl+C キーを押して要求をキャンセルするまで、ライブ フィードとして表示されます。

次のオプションが適用されます。

- **alert**：アラートを表示します。攻撃が進行中であること、または攻撃が試みられたことを示している可能性のある不審なアクティビティを通知します。

  レベル（informational、low、medium、または high）が選択されていない場合は、すべてのアラート イベントが表示されます。
- **include-traits**：指定した特性を持つアラートを表示します。
- **exclude-traits**：指定した特性を持つアラートを表示しません。
- **traits**：10 進数（0 ～ 15）で表した特性ビットの位置。
- **error**：エラー イベントを表示します。エラー イベントは、エラー条件が発生したときにサービスによって生成されます。
- **log**：ログ イベントを表示します。ログ イベントは、トランザクションが受信され、アプリケーションの応答があったときに生成されます。トランザクションの要求、応答、および成功または失敗に関する情報が含まれています。
- **NAC**：Attack Response Controller（ARC）要求を表示します。

  （注）　ARC は、以前は Network Access Controller（NAC）と呼ばれていました。この名前の変更は、IDM および CLI for IPS 5.1 で完全には反映されていません。
- **status**：ステータス イベントを表示します。
- **past**：指定された時間数、分数、秒数の間に開始されたイベントを表示します。
- *hh:mm:ss*：表示を開始する、過去の時、分、秒。

（注）　**show events** コマンドは、指定されたイベントが使用可能になるまで待機します。Ctrl+C キーを押して終了するまでの間、イベントの待機と表示が続きます。

イベント ストアからイベントを表示するには、次の手順を実行します。

ステップ 1　CLI にログインします。

ステップ 2　現在開始されているすべてのイベントを表示します。

```
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.
```

フィードは、**Ctrl+C** キーを押すまですべてのイベントを表示し続けます。

ステップ 3　2005 年 2 月 9 日の午前 10 時から、ブロック要求を表示します。

```
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#
```

ステップ 4　2005 年 2 月 9 日の午前 10 時から、警告レベルのエラーを表示します。

```
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
```

ステップ 5　45 秒前からのアラートを表示します。

```
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--
```

ステップ 6　過去 30 秒間に始まったイベントを表示します。

```
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)
```

イベント ストアからのイベントのクリア

イベント ストアをクリアするには、**clear events** コマンドを使用します。

イベント ストアからイベントをクリアするには、次の手順を実行します。

ステップ 1　管理者特権を持つアカウントを使用して CLI にログインします。

ステップ 2　イベント ストアをクリアします。

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

ステップ 3　**yes** を入力してイベントをクリアします。