ASA Embedded Event Manager の設定例

内容

概要

このドキュメントでは、Adaptive Security Appliance（ASA）バージョン 9.2(1) でトラブルシューティング ツールとして追加された Embedded Event Manager（EEM）について説明します。このツールの機能は Cisco IOS?ベースの EEM と同様です。EEM は、ASA イベント（syslog）に応じて CLI コマンドを実行し、その出力を保存する強力な手段となります。このドキュメントでは、その機能を紹介するとともに、いくつかの EEM アプレットの例を記載します。

前提条件

要件

EEM を使用するには、ASA がシングルコンテキスト モードで設定されている必要があります。

使用するコンポーネント

このドキュメントの情報は、ASA バージョン 9.2(1) 以降に基づくものです。

注意事項と制約事項

この項では、この機能のガイドラインと制限事項について説明します。

コンテキスト モードのガイドライン

現在、EEM はシングル コンテキスト モードで動作する ASA ファイアウォールでのみサポートされています。現時点では、マルチコンテキスト モードで設定されているファイアウォールではサポートされません。

ファイアウォール モードのガイドライン

EEM は現在、ルーテッド モードとトランスペアレント ファイアウォール モードの両方でサポートされています。

その他のガイドライン

- ユニットがクラッシュすると、通常は ASA の状態が不明になります。ASA の状態が不明な間、コマンドの中には実行すると悪影響を及ぼす可能性があるものもあります。
- イベント マネージャ アプレットの名前にスペースを含めることはできません。
- None イベントおよび Crashinfo イベント パラメータは変更できません。
- syslog メッセージが EEM に送信されて処理されることから、パフォーマンスに影響する可能性があります。
- 各イベント マネージャ アプレットの出力は、デフォルトで **output none に設定されます。**デフォルトの出力設定を変更するには、設定に別の値を入力する必要があります。
- 各イベント マネージャ アプレットに定義できる出力オプションは 1 つだけです。

設定

event manager applet コマンドは、イベントにアクションと出力をリンクするプロセスを作成/編集します。このプロセスは、イベント マネージャ アプレットと呼ばれています。*<name> は 32 文字に制限され、スペースを使用することはできません。*イベント マネージャ アプレット サブモードを開始するには、以下のコマンドを入力します。

```
ASA(config)# [no] event manager applet
```

アプレットには、**説明を追加することができます。**これは単なる参考情報です。<text> は 256 文字に制限されています。

```
ASA(config-applet)# [no] description
```

イベントの設定

アプレットには、さまざまなイベントを追加できます。イベントによってアプレットがトリガーされると、アプレットで設定されたアクションが起動されます。イベントを定義するには、**event キーワードを使用します。**アプレットごとに複数のイベントを設定することもできます。

syslog イベント

第一にサポートされるイベント タイプは、**syslog** です。ASA は syslog ID を使用して、アプレットをトリガーする syslog を識別します。id キーワードを使用して設定される syslog は、単一の syslog でも、syslog の範囲でも構いません。オプションの **occurs キーワードでは、アプレットを起動するのに必要な syslog の発生回数を指定します (デフォルトは 1) 。** オプションの **period キーワードでは、イベントの存続期間 (秒数) を指定します。**存続期間を設定すると、アプレットはその期間中に 1 回だけ起動されることになります。たとえば、**occurs を 5 に設定し、period を 30 に設定すると、syslog が 30 秒以内に 5 回発生するまではイベントがトリガーされません。** 30 秒間で syslog が 11 回発生したとしても、アプレットは 1 回しかトリガーされません。**period の値を 0 に設定すると、期間は未定義になります。**

複数の syslog を設定できますが、範囲を重複させることはできません。

```
ASA(config-applet)# [no] event syslog id
```

```
ASA(config-applet)# no event syslog id
```

occurs値<*n*>は1 ~ 4294967295**の許容範囲です。**period**値<**seconds**>**の許容範囲は0 ~ 604800です。0は期間が設定されていないことを意味します。

Syslog イベントの例

この例では、EEM はメモリ ブロック不足の状態を検出するとアクションを実行します。使用可能な 1550 バイトのブロックが枯渇すると、**show blocks pool 1550 dump を収集してディスクに保存します。**このアクションは 10 分間隔で最大 1 回実行されます。

```
event manager applet depletedblock
description "Take a snapshot of block output when it is depleted"
event syslog id 321007 period 600
action 1 cli command "show blocks pool 1550 dump"
output file rotate 10
```

定期イベント

定期的にアクションを実行するように EEM を設定することもできます。タイマー ベースのイベントを設定する場合は、イベント設定で **timer キーワードを使用します。**タイマー ベースのオプションは 3 つあります。

- absolute：最初のタイマーは、毎日指定された時刻に 1 回アプレットをトリガーした後、自動的に再起動する**絶対タイマーです。**
  ```
  ASA(config-applet)# [no] event timer absolute time
  ```
  ```
  ASA(config-applet)# no event timer absolute
  ```
- countdown：2 番目のタイマーは、アプレットを 1 回トリガーした後に再起動しない**カウントダウン タイマーです。このタイマーが再起動するのは、削除または追加しなおされた場合のみです。**
  ```
  ASA(config-applet)# [no] event timer countdown time
  ```
  ```
  ASA(config-applet)# no event timer countdown
  ```
- watchdog：3 番目のタイマーは、設定された間隔で 1 回アプレットをトリガーして自動的に再起動する**ウォッチドッグ タイマーです。**
  ```
  ASA(config-applet)# [no] event timer watchdog time
  ```
  ```
  ASA(config-applet)# no event timer watchdog
  ```

定期イベントの例

たとえば、以下のイベント設定では 192.168.1.100 に対して 1 分間隔で pinp を実行します。空きトラフィック期間中も VPN トンネルがアップ状態に維持されて動作可能であることを確認するには、この設定を使用できます。この設定では、**ウォッチドッグ タイマーを使用して 60 秒間隔でアクションを実行します。**

```
event manager applet period-event
description "Run a command once per minute"
event timer watchdog time 60
action 0 cli command "ping 192.168.1.100"
output none
```

以下のアプレットは、メモリ ブロック割り振り情報を 1 時間ごとに記録して、出力を口グ ファ

イルに書き込みます。記録されるログは 1 日分に相当するため、一連の ログ ファイルが循環して使用されます。この設定では、**ウォッチドッグ タイマーを使用して 1 時間間隔でアクションを実行します。**

```
event manager applet blockcheck
description "Log block usage"
event timer watchdog time 3600
output rotate 24
action 1 cli command "show blocks old"
```

これらのアプレットは、指定されたインターフェイス(Gig 0/0)を午前0時から午前3時の間は無効にします。これは絶対タイマーを使用して、1日1回実行します。

```
event manager applet disableintf
description "Disable the interface at midnight"
event timer absolute time 0:00:00
output none
action 1 cli command "interface GigabitEthernet 0/0"
action 2 cli command "shutdown"
action 3 cli command "write memory"
!
event manager applet enableintf
description "Enable the interface at 3am"
event timer absolute time 3:00:00
output none
action 1 cli command "interface GigabitEthernet 0/0"
action 2 cli command "no shutdown"
action 3 cli command "write memory"
```

手動イベント

これらの EEM アプレットは手動で起動することもできます。それには、アプレットで **event none** を設定する必要があります。アプレットを手動で実行するには、**event manager run コマンドの後にアプレットの名前を続けて入力します。**「none」は別として、アプレットが何らかのイベント トリガー メカニズムに対して設定されている場合、アプレットを手動で実行しようとするとエラーが発生します。上記の例のうち、「depletedblock」を手動で実行しようとすると、以下のエラーが出力されます。

```
ASA# event manager run depletedblock
ERROR: Applet not configured with 'event none'
```

手動イベントの例

手動イベントはマクロと同じような方法で使用できます。たとえば、手動イベントを使用すると、いくつかのコマンドを順番に実行することができます。以下の例では、設定の保存、ホストに対する ping、すべての shun のクリアを順に実行します。

```
event manager applet clean-up
event none
action 0 cli command "write mem"
action 1 cli command "ping 192.168.1.100"
action 2 cli command "clear shun"
output none
```

クラッシュ イベント

ASA 上でクラッシュが発生すると、**crashinfo イベントによってアプレットがトリガーされます。output コマンドの値とは関係なく、action コマンド出力は crashinfo ファイルにリダイレクトされます。**この出力が生成された後、crashinfo の **show tech の部分が生成されます。**

> **警告**：ASA がクラッシュすると、通常は ASA の状態が不明になります。ASA の状態が不明な間、CLI コマンドの中には実行すると悪影響を及ぼす可能性があるものもあります。

```
ASA(config-applet)# [no] event crashinfo
```

アクションの設定

アプレットがトリガーされると、そのアプレットで設定されているアクションが実行されます。各 **action には、アクションの順序を指定するために使用される序数が設定されます。**アプレットごとに複数のアクションを設定できますが、同じ序数を重複させることはことはできません。コマンドは典型的な CLI コマンドです（**show blocks** など）。引用符を使用することを強く推奨しますが、必須ではありません。

```
ASA(config-applet)# [no] action
```

```
ASA(config-applet)# no action
```

アクション識別子*<n>の値の範囲*は0 ~ 4294967295です。<command>の値は引用符で囲む必要があります。引用符で囲まないと、コマンドが複数の単語で構成されている場合にエラーが発生します。コマンドは、特権レベル 15（最高位）を持つユーザとしてコンフィギュレーション モードで実行されます。 コマンドが入力を受け入れないようにすることもできます。それには、コマンドに **noconfirm オプションを使用し、入力を無効にします。**これらのコマンドはインタラクティブに処理されないため、このオプションを使用してください。

出力の設定

action コマンドによる出力は、**output コマンドで指定した場所にリダイレクトできます。**一度に有効にできる outupt 値は 1 つに限られます。デフォルト値は **output none** です。この値は、action コマンドによるすべての出力を破棄します。

```
ASA(config-applet)# [no] output none
```

output console コマンドは、action コマンドの出力をコンソールに送信します。

```
ASA(config-applet)# [no] output console
```

output file コマンドは、action コマンドの出力をファイルに送信します。使用できるオプションは

4 つあります。**new オプションを使用すると、アプレットを起動するごとに、その出力が新しいファイルに書き込まれます。***filename は、***eem-<applet>-<timestamp>.log** の形式で指定します。ここで、*<applet> はアプレットの名前、<timestamp> は YYYYMMDD-hhmmss* 形式の日付のあるタイムスタンプです。

```
ASA(config-applet)# [no] output file new
```

rotate オプションを使用すると、Linux のログ循環メカニズムと同じように一連のファイルが作成されます。作成されるファイル名の形式は **eem-<applet>-<x>.log** です。ここで、*<applet> はアプレットの名前、<x> はファイル番号です。*最も新しいファイルは番号 0（ゼロ）で示され、最も古いファイルは最大の番号（*<n>-1）で示されます。* 新しいファイルが書き込まれる際は、最も古いファイルが削除され、後続のすべてのファイルに番号が再度割り振られてから、番号 0 のファイルに出力が書き込まれます。

```
ASA(config-applet)# [no] output file rotate
```

rotate の値 *<n> に有効な範囲は 2 ~ 100 です。*

overwrite オプションを使用すると、action コマンドの出力が常に単一のファイルに書き込まれ、毎回ファイルの既存の内容が新しい出力で上書きされます。

```
ASA(config-applet)# [no] output file overwrite
```

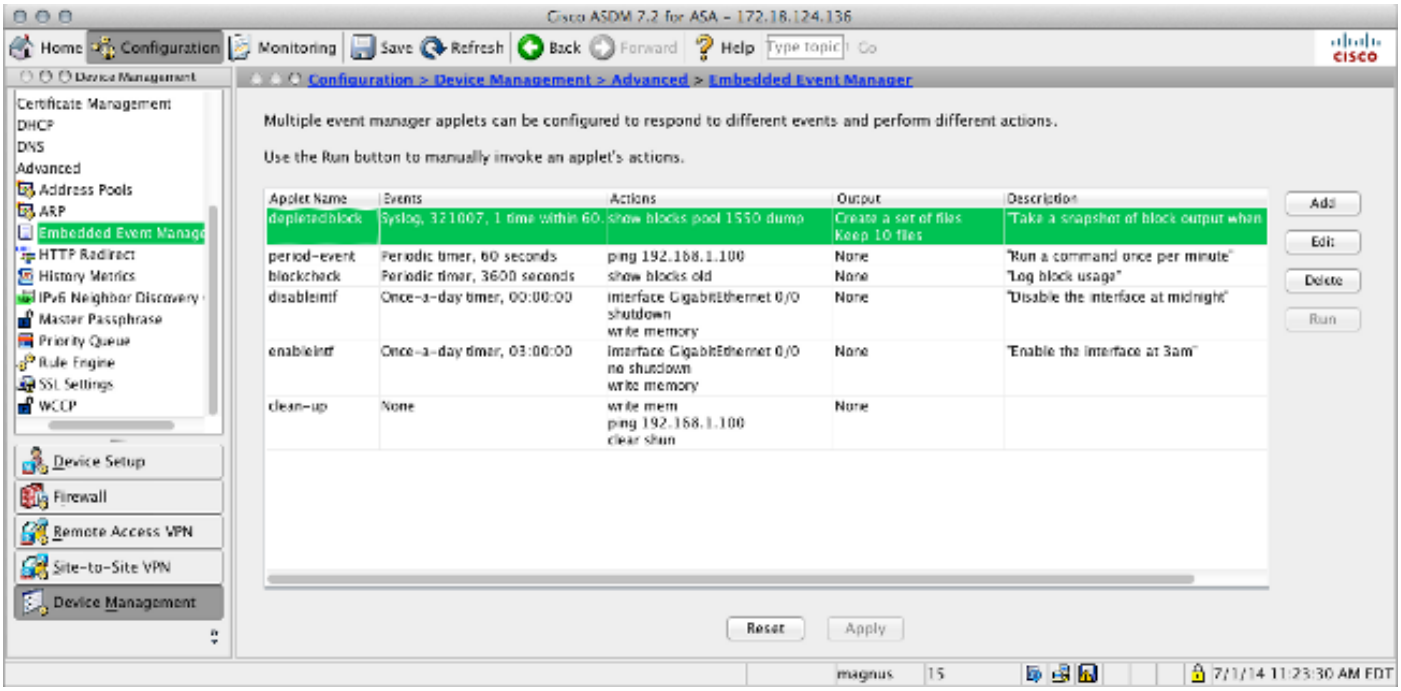
append オプションを使用すると、action コマンドの出力が常に単一のファイルに書き込まれ、毎回ファイルの既存の内容に新しい出力が追加されます。

```
ASA(config-applet)# [no] output file append
```

*<filename> 引数はローカル（ASA の）ファイル名です。*overwrite コマンドには、**ftp:**、**tftp:**、および **smb:**をターゲットとしたファイルを使用することもできます。

ASDM の設定

EEM は、ASDM 内から設定することもできます。その場合は、[Configuration] > [Device Management] > [Advanced] > [Embedded Event Manager]の順に選択します。ASDM のこのセクションで、前述のパラメータの例を使用して EEM アプレットを設定できます。アプレットを設定した後、[Apply]をクリックして設定を ASA にプッシュします。



確認

EXEC モード コマンド

ここでは、設定が正常に機能しているかどうかを確認します。

ここに記載するコマンドはすべて、EXEC モードで使用します。

以下のコマンドは、イベント マネージャ システムの実行コンフィギュレーションを表示します。

ASA# **show running-config event manager**
以下のコマンドは、event none を指定して設定されたイベント マネージャ アプレットを実行します。event none を指定して設定されていないアプレットを実行すると、エラーが発生します。

ASA# **event manager run**

```
       ASA# event manager applet period-event, hits 1, last 2014/07/01 10:51:52
last file none
event watchdog 60 secs, left 54 secs, hits 1, last 2014/07/01 10:51:52
action 0 cli command "ping 192.168.1.100", hits 1, last 2014/07/01 10:51:52 show counter CLI
eem
```

```
   ASA# show counters protocol eem    show show Output Interpreter Tool EEM debug _ ASA# [no]
debug event manager
```

```
ASA# show debug event manager
```