ASA Embedded Event Manager の設定例

目次

概要

このドキュメントでは、適応型セキュリティ アプライアンス（ASA）バージョン 9.2(1) に追加されたトラブルシューティング ツールである Embedded Event Manager（EEM）について説明します。 機能性は Cisco IOS に類似したですか。 基づいた EEM。 これは、ASA イベント（syslog）に基づいて CLI コマンドを実行し、その出力を保存するための強力な手段です。 このドキュメントでは、この機能の概要について説明し、EEM アプレットの例を示します。

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。

ガイドラインと制限事項

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

EEM は現在、シングル コンテキスト モードで稼働する ASA ファイアウォールでのみサポートされています。 マルチコンテキスト モードで設定されたファイアウォールは現在サポートされていません。

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している場合、ASA の状態は一般に不明です。 ASA がこの状態になっている間は、一部のコマンドを安全に実行できない可能性があります。
- 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 イベントと Crashinfo イベントの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ため、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 アプレットのデフォルト出力は **output none** です。 デフォルト出力を変更するには、異なる出力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、イベントをアクションと出力にリンクするプロセスであるイベント マネージャ アプレットを作成または編集します。 *<name>* は 32 文字までに制限され、スペースは使用できません。 これにより、イベント マネージャ アプレット サブモードに入ります。

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

アプレットに **description** を追加できます。 これは単なる参考情報です。 <text> は 256 文字までに制限されています。

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

イベントの設定

アプレットをトリガーして設定されたアクションを呼び出すさまざまなイベントをアプレットに追加できます。 これらは **event** キーワードで定義されます。 1 つのアプレットに複数のイベントを設定できます。

syslog イベント

サポートされる 1 つ目のイベント 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。 これは、単一の syslog または範囲である id キーワードによって行われます。 オプションの **occurs** キーワードは、アプレットを呼び出すために必要な syslog の発生回数を示します (デフォルトは 1 です) 。 オプションの **period** キーワードは、イベントが発生するまでにかかる時間 (秒単位) を示します。 これによって、設定した期間内はアプレットの呼び出しが 1 回までに制限されます。 **occurs** を 5、**period** を 30 に設定した場合は、イベントがトリガーされる前に syslog が 30 秒以内に 5 回発生する必要があります。 syslog が 30 秒間に 11 回発生した場合でも、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、期間は定義されません。

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 <nnnnnn>[-<nnnnnn>] [occurs <n>]
[period <seconds>]ASA(config-applet)# no event syslog id <nnnnnn>[-<nnnnnn>]
```

occurs の値 *<n>* は、1 から 4294967295 までの範囲で設定できます。 **period** の値 <seconds> は、0 から 604800 までの範囲で設定できます。 値を 0 (ゼロ) に設定すると、期間は設定されません。

syslog イベントの例

この例では、メモリ ブロックの不足状態が検出されたときに EEM が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削除されると、**show blocks pool 1550 dump** を収集してディスクに保存します。 これは 10 分ごとに最大 1 回行われます。

```
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

アクションを定期的に実行するように 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ときは、イベントの設定で **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には次の 3 つがあります。

- absolute：1 つ目のタイマーは、1 日 1 回指定された時間にアプレットをトリガーして自動的に再開する**絶対**タイマーです。`ASA(config-applet)# [no] event timer absolute time <hh:mm:ss>`
  `ASA(config-applet)# no event timer absolute`
- countdown：2 つ目のタイマーは、アプレットを 1 回だけトリガーし、削除して再度追加しない限り再開しない**カウントダウン** タイマーです。`ASA(config-applet)# [no] event timer countdown time <seconds>`
  `ASA(config-applet)# no event timer countdown`

- watchdog：3 つ目のタイマーは、設定された期間ごとに 1 回ずつアプレットをトリガーして自動的に再開する**ウォッチドッグ** タイマーです。ASA(config-applet)# **[no] event timer watchdog time <seconds>**
  ASA(config-applet)# **no event timer watchdog**

定期イベントの例

たとえば、次のイベント設定は 1 分ごとに 192.168.1.100 に対して ping を実行します。 これは、VPN トンネルが空きトラフィックの期間中も継続して稼働していることを確認するために使用できます。 60 秒ごとに実行する**ウォッチドッグ タイマー**が使用されています。

```
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
```

次のアプレットは、1 日分のログを保持するため、メモリ ブロックの割り当て情報を 1 時間ごとに記録し、ログ ファイルの循環セットに出力を書き込みます。 1 時間ごとに実行する**ウォッチドッグ タイマー**が使用されています。

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

次のアプレットは、特定のインターフェイス (ギガビット イーサネット 0/0) を午前 0 時から 3 時までディセーブルにします。 それは絶対**タイマーを 1** 日あたりに一度実行するのに使用します。

```
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 アプレットを手動で起動することもできます。 そのためには、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 コマンドに続けてアプレットの名前を入力します。 アプレットを「none」以外のイベント トリガー メカニズム用に設定した場合は、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。 前の例の「depletedblock」を使用すると、次のように表示されます。

```
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、マクロと同じように使用できます。 たとえば、手動イベントを使用していくつかのコマンドを順番に実行できます。 次の例では、設定を保存し、ホストに対して ping を実行し、すべての排除をクリアします。

```
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

crashinfo イベントは、ASA でクラッシュが発生したときにアプレットをトリガーします。 **output** コマンドの値に関係なく、**action** コマンドはクラッシュ情報ファイルを対象とします。 この出力は、crashinfo の **show tech** 部分が生成される前に生成されます。

> **警告**： ASA がクラッシュしている場合、ボックスの状態は一般に不明です。 ユニットがこの状態になっているときは、一部の CLI コマンドを安全に実行できない可能性があります。

```
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、アプレットのアクションが実行されます。 各**アクション**は、アクションの順序を指定するために使われる序数を持っています。 1 つのアプレットに複数のアクションを設定できますが、 同じ序数は 1 回しか使用できません。 コマンドは、**show blocks** などの一般的な CLI コマンドです。 引用符を付けることが（必須ではありませんが）強く推奨されます。

```
ASA(config-applet)# [no] action <n> cli command "<command>"ASA(config-applet)# no action <n>
```

アクション ID の値 *<n>* は、0 から 4294967295 までの範囲で指定できます。 *<command>* の値は引用符で囲む必要があります。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）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場合があります。 これは、コマンドに **noconfirm** オプションが指定された場合は入力がディセーブルになるためです。 コマンドは対話形式で処理されないため、このオプションを使用する必要があります。

出力の設定

アクションの出力は、**output** コマンドを使用して指定された場所に送信できます。 一度にイネーブルにできる出力値は 1 つだけです。 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。 4 つのオプションを使用できます。 **new** オプションは、アプレットの出力を呼び出しごとに新しいファイルに書き込みます。 *filename* の形式は **eem-<applet>-<timestamp>.log** です。 *<applet>* はアプレットの名前、*<timestamp>* は *YYYYMMDD-hhmmss* 形式の日付付き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な循環するファイル セットが作成されます。 ファイル名の形式は **eem-<applet>-<x>.log** です。 *<applet>* はアプレットの名前、*<x>* はファイル番号です。 ファイル番号は、最も新しいファイルが 0 で、最も古いファイルが最大数（*<n>-1*）になります。 新しいファイルを書き込むときは、最も古いファイルが削除され、後続のすべてのファイルの番号を振り直してから 0 番目のファイルが書き込まれます。

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

rotate の値 *<n>* は、2 から 100 までの範囲で設定できます。

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、ファイルの内容が毎回切り捨てられます。

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

append オプションを使用した場合、action コマンドの出力は常に単一のファイルの書き込まれますが、ファイルの末尾に追加で書き込まれます。

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

<filename> 引数は、（ASA に対して）ローカルなファイル名です。 overwrite コマンドの書き込み先として **ftp:**、**tftp:** 、および **smb:** ファイルを使用することもできます。

ASDM の設定

ASDM 内から EEM を設定することもできます。 [Configuration] > [Device Management] > [Advanced] > [Embedded Event Manager] を選択します。 ASDM のこのセクションで、前述のパラメータを使用して EEM アプレットを設定できます。 アプレットを設定したら、[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、設定が正常に動作していることを確認します。

これらのコマンドは、すべて EXEC モードで使用されます。

次のコマンドは、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
```

次のコマンドは、**event none** を使用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** を使用して設定されていないアプレットを実行すると、エラーが報告されます。

```
ASA# event manager run <applet>
```

次のコマンドは、設定されたアプレットに関する情報を表示します。これには、ヒット カウントやアプレットが最後に起動された時間が含まれています。

```
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
```

イベント マネージャは標準のカウンタを使用します。 **show counter** CLI での制限のため、プロトコル フィルタリング用に **eem** キーワードが使用されています。

```
ASA# show counters protocol eem
```

特定の show コマンドがアウトプット インタープリタ ツール（登録ユーザ専用）でサポートされています。 show コマンド出力の分析を表示するには、アウトプット インタープリタ ツールを使用します。

デバッグ

EEM をデバッグし、その出力を表示するには、次のコマンドを入力します。

> **注:** debug コマンドを使用する前に、『**debug コマンドの重要な情報**』を参照してください。

```
ASA# [no] debug event manager <n>
ASA# show debug event manager
```

トラブルシューティング

現在のところ、この設定に関する特定のトラブルシューティング情報はありません。 期待どおりに動作しない場合は、前の項に示したデバッグと検証の手順を使用して、エラーが発生したかどうかを確認してください。