ASA嵌入式事件管理器配置示例

目录

简介

本文档介绍嵌入式事件管理器(EEM)，它是在自适应安全设备(ASA)版本9.2(1)中添加的故障排除工具。 功能与Cisco IOS类似[?]基于EEM。它是根据ASA事件(syslog)运行CLI命令并保存输出的强大方法。本文档介绍此功能以及一些EEM小程序示例。

先决条件

要求

使用EEM要求在单情景模式下配置ASA。

使用的组件

本文档中的信息基于ASA 9.2(1)版或更高版本。

准则和限制

本节包括此功能的指南和限制。

情景模式指南

EEM目前仅在单情景模式下运行的ASA防火墙上受支持。当前不支持在多情景模式下配置的防火墙。

防火墙模式指南

EEM当前在路由和透明防火墙模式中均受支持。

其他准则

- 设备崩溃时，ASA的状态通常未知。当ASA处于此状态时，某些命令可能不安全运行。
- 事件管理器小程序的名称不能包含空格。
- 不能修改None事件和Crashinfo事件参数。
- 系统日志消息发送到EEM进行处理，可能会影响性能。
- 每个事件管理器小**程序的默**认输出为none输出。要更改默认输出，必须输入不同的输出值。
- 您可能只为每个事件管理器小程序定义了一个输出选项。

配置

事件**管理器小程序**命令创建/编辑事件管理器小程序，该进程将事件与操作和输出链接起来。<name>*的*长度限制为32个字符，不能有空格。这将进入事件管理器小程序子模式。

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

说明**可**以添加到小程序。此名称仅用于参考目的。<*text*>最多限制为256个字符。

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

事件配置

可以将各种事件添加到触发小程序以调用其上配置的操作的小程序。它们使用event关键字**进行**定义。可能为每个小程序配置多个事件。

系统日志事件

支持的第一个事件类型是**syslog**。ASA使用系统日志ID来识别触发小程序的系统日志。这通过id关键字完成，该关键字可能是单个系统日志或范围。可选**occurs** 关键字指示系统日志必须发生的次数，以便调用小程序（默认值为1）。 可选**period**关键字指示事件必须发生的时间量（以秒为单位）。它将小程序调用的频率限制为在配置的时间段内最多一次。**出现**5，周期**为**30，表示系统日志必须在触发事件之前30秒内发生5次。如果系统日志在30秒内发生11次，则只触发一次小程序。值0表示**未**定义期间。

可以配置多个系统日志，但范围不能重**叠**。

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

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

发**生**值<*n*>的允许范围为1到4294967295。**期间值**<seconds>的允许范围为0到604800。0（零）值表示未配置任何期间。

系统日志事件示例

在本例中，EEM在检测到内存块不足情况时采取操作。如果可用的1550字节块耗尽，它会收**集** show blocks pool 1550 dump**并**保存到磁盘。它最多每10分钟执行一次。

```
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
```

定期事件

EEM也可配置为定期执行操作。配置基于计时器的事件时，请在事件配置中使用**timer**关键字。有3个基于计时器的选项：

- absolute — 第一个计时器是**绝对**计时器，它每天在指定时间触发小程序一次并自动重新启动。
  ASA(config-applet)# **[no] event timer absolute time**

```
ASA(config-applet)# no event timer absolute
```

- 倒计时 — 第二个计时器是倒计**时计**时器，它触发小程序一次，除非删除并重新添加，否则不会重新启动。

```
ASA(config-applet)# [no] event timer countdown time
```

```
ASA(config-applet)# no event timer countdown
```

- watchdog — 第三个计时器是监视**器计**时器，它按配置的周期触发小程序一次并自动重新启动。

```
ASA(config-applet)# [no] event timer watchdog time
```

```
ASA(config-applet)# no event timer watchdog
```

定期事件示例

例如，此事件配置每1分钟ping 192.168.1.100。这可用于确保VPN隧道保持正常运行，即使在空闲流量期间也能正常运行。它使用监视**程序**计时器每60秒执行一次。

```
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
```

此小程序每小时记录内存块分配信息并将输出写入一组旋转日志文件，因为它保留一天的日志值。它使用监视**程序**计时器每1小时执行一次。

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

这些小程序在午夜到凌晨3点之间禁用给定接口(Gig 0/0)。它使用绝对计时器**每天**执行一次。

```
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"
```

手动事件

这些EEM小程序也可以手动调用。为此，小程序必须配置**event none**。要手动运行小程序，请输入事件管理**器运行**命令，后跟小程序的名称。如果为除“none”之外的任何事件触发机制配置了小程序，则尝试手动运行小程序会生成错误。使用前面的一个示例“耗尽块”，您会看到：

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

手动事件示例

手动事件可以与宏类似的方式使用。例如，手动事件可用于按顺序执行几个命令。在本例中，它保存配置、ping主机并清除所有避开。

```
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
```

崩溃事件

当ASA上发生崩溃时，**crashinfo** 事件会触发小程序。无论输出命令的**值**如何，操**作**命令都会定向到crashinfo文件。在生成crashinfo的**show tech**部分之前，会生**成输出。**

> **警告：**当ASA崩溃时，机箱状态通常未知。当设备处于此状态时，某些CLI命令可能不安全运行。

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

操作配置

触发小程序后，将执行小程序上的操作。每个**操**作都有用于指定操作顺序的序号。每个小程序可以配置多个操作；但每个序数只能使用一次。这些命令是典型的CLI命令，如**show blocks**。强烈建议使用报价，但不要求使用。

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

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

操作标识符*<n>*的值范围为0到4294967295。必须引用*<command>的值*，否则，如果命令包含多个字，则会出错。该命令在配置模式下以权限级别为15（最高）的用户身份执行。 命令可能不接受任何输入；如果命令具有noconfirm选项，则将禁用**as输**入。应使用此命令，因为命令不是交互处理的。

输出配置

操作的输出可通过输出命令定向到指定**位置**。一次只能启用一个输出值。默认值为**output none**。此值将丢弃action命令的所有输出。

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

output console命令将action命令的输出发送到控制台。

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

output file命令将action命令的输出定向到文件。可以使用四个选项。**new**选项将小程序的输出写入每个调用的新文件。文*件名*的格式为**eem-*<applet>*-*<timestamp>***.log。其中*<applet>*是小应用的名称，*<timestamp>*是YYYYMMDD-hhmmss格式的日期*时间戳*。

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

rotate选项用于创建一组文件，这些文件与Linux的日志旋转机制类似。文件名格**式为*eem-<applet>-<x>*.log**。其中*<applet>*是小应用的名称，*<x>*是文件编号。最新文件以数字0（零）表示，最旧文件以最高数字(*<n>*-1)表示。 当要写入新文件时，最旧的文件将被删除，并且所有后续文件都将重新编号，然后才写入第0个文件。

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

旋转值*<n>*的范围为2至100。

overwrite选项用于始终将action命令输出写入每次被截断的单个文件。

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

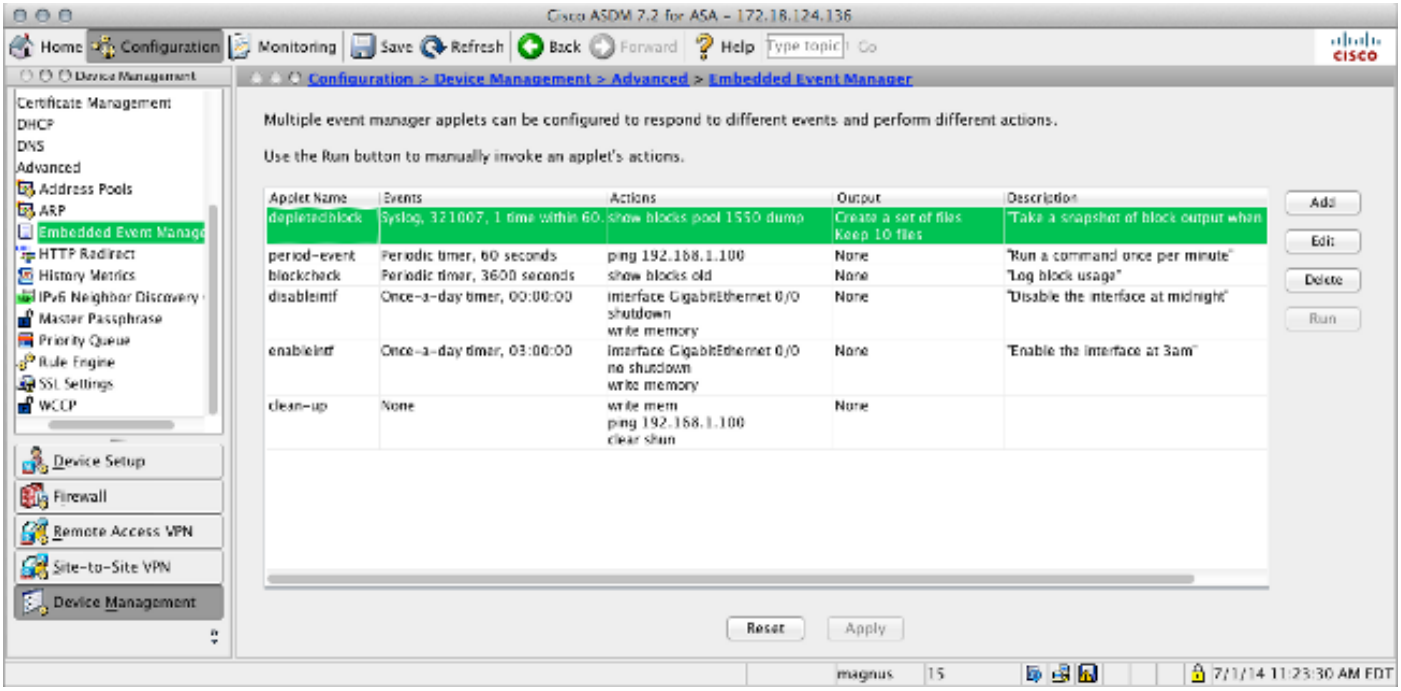
append选项用于始终将action命令输出写入单个文件，但该文件每次都会被附加到。

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

<filename>参数是本地（对于ASA）文件名。overwrite命令也可能使用**ftp:**、**tftp:**和**smb:**目标文件。

ASDM 配置

EEM也可以从ASDM内进行配置。选择**Configuration > Device Management > Advanced > Embedded Event Manager**。在ASDM的此部分，您可以使用前面讨论的相同参数配置EEM小程序。配置小程序后，单击**Apply**将配置推送到ASA。



验证

执行模式命令

使用本部分可确认配置能否正常运行。

所有这些命令都用于执行模式。

此命令显示事件管理器系统的运行配置。

```
ASA# show running-config event manager
```

此命令执行已配置了事件无的事件管理器小**程序**。如果运行的小程序尚未配置**event none**，则报告错误。

```
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 10:51:52show counter CLI,eem

ASA# show counters protocol eem show show EEM debug Debug

ASA# [no] debug event manager

ASA# show debug event manager