Ejemplo de Configuración de Embedded Event Manager ASA

Contenido

Introducción

Este documento describe Embedded Event Manager (EEM), que es una herramienta de resolución de problemas que se agregó en Adaptive Security Appliance (ASA) versión 9.2(1). ¿La funcionalidad es similar a Cisco IOS? EEM basado Es una forma eficaz de ejecutar comandos CLI basados en eventos ASA (syslogs) y guardar el resultado. Este documento describe una introducción a la función, así como algunos ejemplos de subprogramas EEM.

Prerequisites

Requirements

El uso de EEM requiere que el ASA se configure en el modo de contexto único.

Componentes Utilizados

La información de este documento se basa en ASA versión 9.2(1) o posterior.

Pautas y limitaciones

Esta sección incluye las pautas y limitaciones para esta función.

Pautas del modo de contexto

Actualmente, EEM sólo es compatible con los firewalls ASA que se ejecutan en el modo de contexto único. Los firewalls configurados en el modo de contexto múltiple no se soportan actualmente.

Pautas del modo de firewall

Actualmente, EEM es compatible con los modos de firewall ruteado y transparente.

Directrices adicionales

- Mientras la unidad se desconecta, el estado del ASA es generalmente desconocido. Es posible que algunos comandos no sean seguros para ejecutarse mientras el ASA está en esta condición.
- El nombre de un applet del administrador de eventos no puede contener espacios.
- No puede modificar los parámetros de evento None y Crashinfo.
- El rendimiento puede verse afectado porque los mensajes syslog se envían al EEM para su procesamiento.
- El resultado predeterminado es **output none** para cada applet del administrador de eventos. Para cambiar el resultado predeterminado, debe ingresar un valor de salida diferente.
- Es posible que sólo tenga definida una opción de salida para cada applet del administrador de eventos.

Configurar

El comando **event manager applet** crea/edita un applet event manager, un proceso que vincula eventos con acciones y resultados. El *<nombre>* está limitado a 32 caracteres y no puede tener espacios. Esto ingresa en un submodo del applet del administrador de eventos.

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

Una **descripción** se puede agregar a un applet. Esto es sólo con fines informativos. El <*text*> está limitado a 256 caracteres.

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

Configuración del evento

Se pueden agregar varios eventos a un applet que activan el applet para invocar las acciones configuradas en él. Se definen con la palabra clave **event**. Se pueden configurar varios eventos para cada applet.

Eventos de Syslog

El primer tipo de evento soportado es **syslog**. El ASA utiliza ID de syslog para identificar los syslogs que activan un applet. Esto se completa a través de la palabra clave id, que puede ser un único syslog o un rango. La palabra clave opcional **se produce** indica el número de veces que se debe producir el syslog para que se invoque el applet (el valor predeterminado es 1). La palabra clave **period** opcional indica la cantidad de tiempo, en segundos, en la que debe ocurrir el evento. Limita la frecuencia de la invocación del applet al menos una vez el período configurado. Se **produce** 5 con un **período** de 30, significa que el syslog debe ocurrir 5 veces en 30 segundos antes de que se active el evento. Si el syslog ocurre 11 veces en 30 segundos, el applet sólo se activa una vez. Un valor de 0 para el **período** significa que no se define ningún período.

Se pueden configurar varios registros del sistema, pero los rangos no pueden superponerse.

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

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

El valor *<n>* **se produce** tiene un rango permitido de 1 a 4294967295. El valor **de período** <seconds> tiene un rango permitido de 0 a 604800. Un valor 0 (cero) significa que no se ha configurado ningún período.

Ejemplo de Eventos de Syslog

En este ejemplo, EEM actúa cuando detecta una condición de bloque de memoria baja. Si los bloques de 1550 bytes disponibles se agotan, recopila **show blocks pool 1550 dump** y los guarda en el disco. Lo hace, como mucho, una vez cada 10 minutos.

```
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
```

Eventos periódicos

EEM también se puede configurar para realizar una acción periódicamente. Cuando configure un evento basado en temporizador, utilice la palabra clave **timer** en la configuración de eventos. Hay 3 opciones basadas en temporizador:

- absoluto - El primer temporizador es un temporizador **absoluto** que activa el applet una vez al día a la hora especificada y se reinicia automáticamente.
  `ASA(config-applet)# **[no] event timer absolute time**`

  `ASA(config-applet)# **no event timer absolute**`
- cuenta atrás - El segundo temporizador es un temporizador **de cuenta atrás** que activa el applet una vez y no se reinicia a menos que se elimine y se vuelva a agregar.
  `ASA(config-applet)# **[no] event timer countdown time**`

  `ASA(config-applet)# **no event timer countdown**`
- watchdog - El tercer temporizador es un temporizador **watchdog** que activa el applet una vez por período configurado y se reinicia automáticamente.
  `ASA(config-applet)# **[no] event timer watchdog time**`

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

Ejemplo de Eventos Periódicos

Por ejemplo, esta configuración de evento hace ping 192.168.1.100 cada 1 minuto. Esto se podría utilizar para asegurar que un túnel VPN se mantenga activo y en funcionamiento incluso durante períodos de tráfico inactivo. Utiliza el temporizador **watchdog** para ejecutar cada 60 segundos.

```
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
```

Este applet registra la información de asignación de bloques de memoria cada hora y escribe el resultado en un conjunto rotatorio de archivos de registro, ya que mantiene el valor de un día de registros. Utiliza el temporizador **watchdog** para ejecutarse cada hora.

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

Estos applets desactivan la interfaz dada (Gig 0/0) entre la medianoche y las 3 a.m. Utiliza el temporizador **absoluto** para ejecutarse una vez al día.

```
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"
```

Evento manual

Estos subprogramas EEM también se pueden invocar manualmente. Para hacer esto, el applet debe configurar **event none**. Para ejecutar un applet manualmente, ingrese el **comando event manager run** seguido del nombre del applet. Si el applet está configurado para cualquier mecanismo de disparador de eventos aparte de 'none', el intento de ejecutarlo manualmente genera un error. Con el uso de uno de los ejemplos anteriores, 'depletedblock', verá:

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

Ejemplo de Evento Manual

Los eventos manuales se pueden utilizar de forma similar a una macro. Por ejemplo, un evento manual podría utilizarse para ejecutar algunos comandos en orden. En este ejemplo, guarda la configuración, hace ping a un host y borra todos los rechazos.

```
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
```

Evento de desperfecto

El evento **crashinfo** activa un applet cuando se produce un desperfecto en el ASA. Independientemente del valor del comando **output**, los comandos **action** se dirigen al archivo crashinfo. El resultado se genera antes de que se genere la parte **show tech** de crashinfo.

> **Advertencia:** Cuando el ASA se cae, el estado de la caja generalmente se desconoce. Es posible que algunos comandos CLI no sean seguros cuando la unidad se encuentra en esta condición.

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

Configuración de acciones

Cuando se activa el applet, se realizan las acciones del applet. Cada **acción** tiene un ordinal que se utiliza para especificar el orden de las acciones. Se pueden configurar varias acciones por applet; pero cada ordinal sólo puede utilizarse una vez. Los comandos son comandos CLI típicos, como **show blocks**. Se recomienda encarecidamente utilizar los presupuestos, pero no son obligatorios.

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

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

El valor del identificador de acción *<n>* tiene un rango de 0 a 4294967295. Se debe presupuestar el valor del *<comando>*; de lo contrario, se produce un error si el comando consta de más de una palabra. El comando se ejecuta en modo de configuración como usuario con el nivel de privilegio 15 (el más alto). Es posible que el comando no acepte ninguna entrada; como input se inhabilitará si un comando tiene la opción **noconfirm**. Esto debería usarse ya que los comandos no se procesan interactivamente.

Configuración de salida

El resultado de las acciones se puede dirigir a una ubicación especificada a través del comando **output**. Sólo se puede habilitar un valor de salida en cualquier momento. El valor predeterminado es **output none**. Este valor descarta cualquier resultado de los comandos de acción.

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

El comando **output console** envía el resultado de los comandos de acción a la consola.

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

El comando **output file** dirige el resultado de los comandos de acción a los archivos. Hay cuatro opciones que se pueden utilizar. La **nueva** opción escribe el resultado del applet en un nuevo archivo para cada invocación. El *nombre de archivo* tiene el formato **eem-*<applet>*-*<timestamp>*.log**. Donde *<applet>* es el nombre del applet y *<timestamp>* es una marca de tiempo fechada en el formato de *YYYMMDD-hmmss*.

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

La opción **rotate** se utiliza para crear un conjunto de archivos que se rotan de manera similar al mecanismo de rotación de registros de Linux. El formato de nombre de archivo es **eem-*<applet>*-*<x>*.log**. Donde *<applet>* es el nombre del applet, y *<x>* es el número de archivo. El archivo más reciente se indica con el número 0 (cero) y el archivo más antiguo se indica con el número más alto (*<n>*-1). Cuando se va a escribir un archivo nuevo, se elimina el archivo más antiguo y se renumeran todos los archivos posteriores antes de que se escriba el archivo 0º.

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

El valor de rotación *<n>* tiene un rango de 2 a 100.

La opción **overwrite** se utiliza para escribir siempre el resultado del comando action en un solo archivo que se trunca cada vez.

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

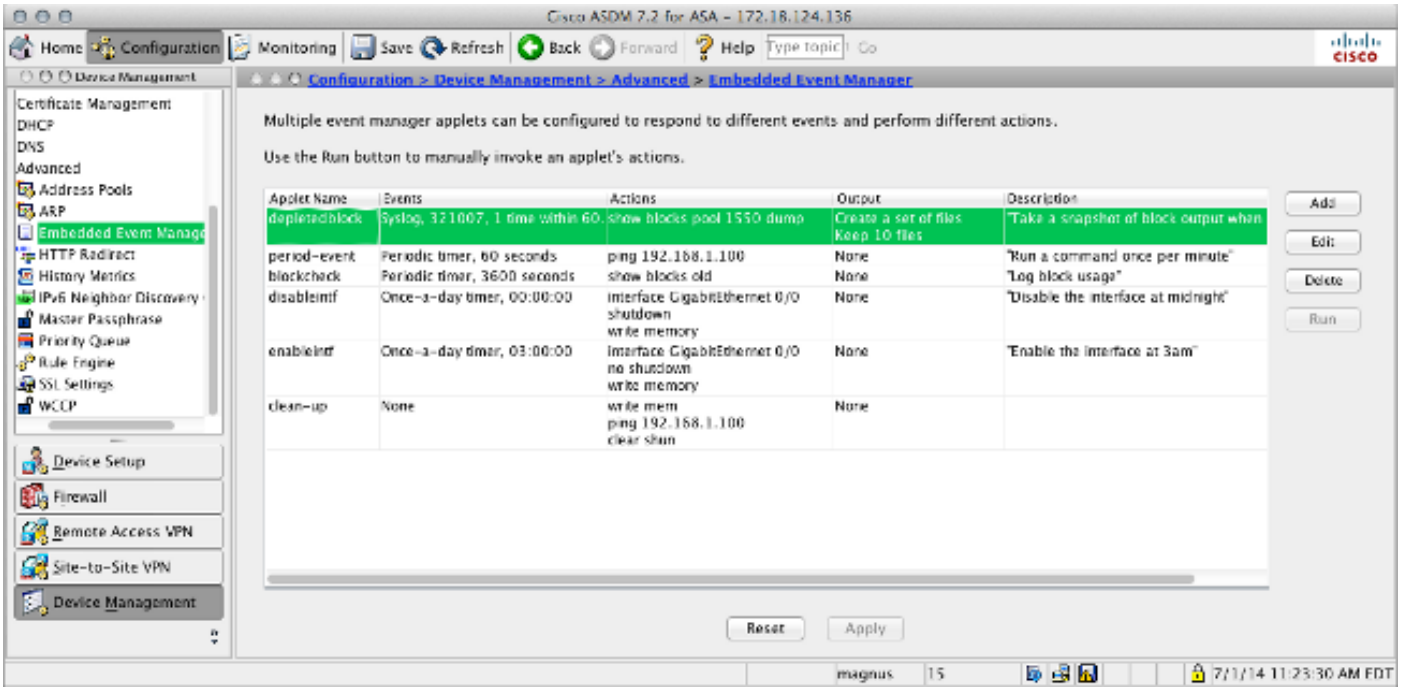
La opción **append** se utiliza para escribir siempre el resultado del comando action en un solo archivo, pero ese archivo se agrega a cada vez.

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

El argumento *<filename>* es un nombre de archivo local (al ASA). El comando overwrite también podría utilizar **ftp**:, **tftp:** y **smb:** archivos objetivo.

Configuración de ASDM

EEM también se puede configurar desde el ASDM. Elija **Configuration > Device Management > Advanced > Embedded Event Manager**. En esta sección de ASDM, puede configurar sus subprogramas EEM con los mismos parámetros que se han analizado anteriormente. Después de configurar un applet, haga clic en **Apply** para enviar la configuración al ASA.



Verificación

Comandos de Modo Exec

Use esta sección para confirmar que su configuración funciona correctamente.

Todos estos comandos se utilizan en el modo exec.

Este comando muestra la configuración en ejecución del sistema del administrador de eventos.

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

Este comando ejecuta un applet del administrador de eventos que se ha configurado con **event none**. Si ejecuta un applet que no se ha configurado con **event none**, se informa de un error.

ASA# **event manager run**

Este comando muestra información sobre los applets configurados, que incluye los recuentos de aciertos y cuándo se invocó el applet por última vezASA# 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 10:51:52El administrador de eventos utiliza los contadores estándar. Debido a las limitaciones dentro de la CLIshow counter, la palabra clave eem se utiliza para el filtrado de protocolos.

ASA# show counters protocol eem La herramienta de interpretación de información de salida (disponible para clientes registrados únicamente) admite ciertos comandos show. Utilice la herramienta para ver una análisis de información de salida del comando show.DepurarIngrese

estos comandos para depurar el EEM y mostrar el resultado.Nota: Consulte Información Importante sobre Comandos de Debug antes de usar un comando debug.

```
ASA# [no] debug event manager
```

ASA# show debug event manager

Troubleshoot

Actualmente, no hay información específica de troubleshooting disponible para esta configuración. Si no funciona como se esperaba, utilice los pasos de depuración y verificación enumerados en la sección anterior para determinar si se ha producido un error.