Konfigurationsbeispiel für ASA Embedded Event Manager

Inhalt

Einführung

Dieses Dokument beschreibt Embedded Event Manager (EEM), ein Tool zur Fehlerbehebung, das in Adaptive Security Appliance (ASA) Version 9.2(1) hinzugefügt wurde. Die Funktionen ähneln Cisco IOS? EEM basiert. Es ist eine leistungsstarke Methode, um CLI-Befehle auf Basis von ASA-Ereignissen (Syslogs) auszuführen und die Ausgabe zu speichern. In diesem Dokument werden eine Einführung in die Funktion sowie einige EEM-Applets erläutert.

Voraussetzungen

Anforderungen

Für die Verwendung von EEM muss die ASA im Single-Context-Modus konfiguriert werden.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf ASA Version 9.2(1) oder höher.

Richtlinien und Einschränkungen

Dieser Abschnitt enthält die Richtlinien und Einschränkungen für diese Funktion.

Richtlinien für den Kontextmodus

EEM wird derzeit nur auf ASA-Firewalls unterstützt, die im Single-Context-Modus ausgeführt werden. Firewalls, die im Multiple-Context-Modus konfiguriert sind, werden derzeit nicht unterstützt.

Richtlinien für den Firewall-Modus

EEM wird derzeit sowohl im Routing- als auch im transparenten Firewall-Modus unterstützt.

Zusätzliche Richtlinien

- Während das Gerät abstürzt, ist der Status der ASA im Allgemeinen unbekannt. Manche Befehle sind möglicherweise nicht sicher auszuführen, solange sich die ASA in diesem Zustand befindet.
- Der Name eines Event Manager-Applets darf keine Leerzeichen enthalten.
- Sie können die Parameter None-Ereignis und Crashinfo-Ereignis nicht ändern.
- Die Leistung kann beeinträchtigt werden, da Syslog-Meldungen zur Verarbeitung an den EEM gesendet werden.
- Die Standardausgabe ist **keine Ausgabe** für jedes Event Manager-Applet. Um die Standardausgabe zu ändern, müssen Sie einen anderen Ausgabewert eingeben.
- Möglicherweise ist für jedes Event Manager-Applet nur eine Ausgabeoption definiert.

Konfigurieren

Der Befehl **event manager applet** erstellt/bearbeitet ein Ereignismanager-Applet, ein Prozess, der Ereignisse mit Aktionen und Ausgaben verknüpft. *<name>* ist auf 32 Zeichen beschränkt und darf keine Leerzeichen enthalten. Dadurch wird der Submodus eines Ereignismanager-Applets aktiviert.

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

Eine **Beschreibung** kann einem Applet hinzugefügt werden. Dies dient ausschließlich Informationszwecken. Der <*Text*> ist auf 256 Zeichen beschränkt.

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

Ereigniskonfiguration

Einem Applet können verschiedene Ereignisse hinzugefügt werden, die das Applet auslösen, um die darin konfigurierten Aktionen aufzurufen. Sie werden mit dem **event-**Schlüsselwort definiert. Für jedes Applet können mehrere Ereignisse konfiguriert werden.

Syslog-Ereignisse

Der erste unterstützte Ereignistyp ist **Syslog**. Die ASA verwendet Syslog-IDs, um Syslogs zu identifizieren, die ein Applet auslösen. Dies wird durch das id-Schlüsselwort abgeschlossen, das ein einzelnes Syslog oder ein Bereich sein kann. Das optionale Schlüsselwort "**tritt auf**" gibt an, wie oft das Syslog ausgeführt werden muss, damit das Applet aufgerufen werden kann (der Standardwert ist 1). Das optionale **period**-Schlüsselwort gibt die Zeitdauer (in Sekunden) an, in der das Ereignis stattfinden muss. Sie beschränkt die Häufigkeit des Applet-Aufrufs auf höchstens den konfigurierten Zeitraum. Wenn **5** mit einem **Zeitraum** von 30 **auftritt**, muss das Syslog innerhalb von 30 Sekunden fünf Mal erfolgen, bevor das Ereignis ausgelöst wird. Wenn das Syslog 11 Mal in 30 Sekunden auftritt, wird das Applet nur einmal ausgelöst. Ein Wert von 0 für einen **Zeitraum** bedeutet, dass kein Punkt definiert ist.

Es können mehrere Syslogs konfiguriert werden, aber die Bereiche können sich nicht überschneiden.

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

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

Der **eintretende** Wert *<n>* liegt im zulässigen Bereich von 1 bis 4294967295. Der **Periodenwert** <Sekunden> liegt im zulässigen Bereich von 0 bis 604.800. Ein 0-Wert (Null) bedeutet, dass kein Punkt konfiguriert ist.

Beispiel für Syslog-Ereignisse

In diesem Beispiel ergreift EEM Maßnahmen, wenn es eine Bedingung für einen niedrigen Speicherblock erkennt. Wenn die verfügbaren 1550-Byte-Blöcke erschöpft sind, werden **die Blöcke "Pool 1550 dump"** gesammelt und auf der Festplatte gespeichert. Dies geschieht höchstens einmal alle 10 Minuten.

```
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
```

Regelmäßige Ereignisse

EEM kann auch für eine regelmäßige Aktion konfiguriert werden. Wenn Sie ein Timer-basiertes Ereignis konfigurieren, verwenden Sie das **timer**-Schlüsselwort in der Ereigniskonfiguration. Es gibt drei Timer-basierte Optionen:

- absolut - Der erste Timer ist ein **absoluter** Timer, der das Applet einmal pro Tag zur angegebenen Zeit auslöst und automatisch neu startet.
  ASA(config-applet)# **[no] event timer absolute time**

  ASA(config-applet)# **no event timer absolute**
- Countdown - Der zweite Timer ist ein **Countdown**-Timer, der das Applet einmal auslöst und erst dann neu startet, wenn es entfernt und neu hinzugefügt wurde.
  ASA(config-applet)# **[no] event timer countdown time**

  ASA(config-applet)# **no event timer countdown**
- Watchdog - Der dritte Timer ist ein **Überwachungs-**Timer, der das Applet einmal pro konfiguriertem Zeitraum auslöst und automatisch neu startet.
  ASA(config-applet)# **[no] event timer watchdog time**

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

Beispiel für periodische Ereignisse

Diese Ereigniskonfiguration pingt beispielsweise alle 1 Minute 192.168.1.100. So kann sichergestellt werden, dass ein VPN-Tunnel selbst bei Leerlaufzeiten betriebsbereit bleibt. Er verwendet den **Überwachungs-**Timer, um alle 60 Sekunden auszuführen.

```
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
```

Dieses Applet zeichnet stündlich Speicherblockzuweisungsinformationen auf und schreibt die Ausgabe in einen rotierenden Satz von Protokolldateien, da die Protokolldateien einen Tag wert sind. Er verwendet den **Überwachungs-**Timer, um alle 1 Stunde auszuführen.

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

Diese Applets deaktivieren die angegebene Schnittstelle (Gig 0/0) zwischen Mitternacht und 3 Uhr. Sie verwendet den **absoluten** Timer, um einmal pro Tag auszuführen.

```
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"
```

Manuelle Veranstaltung

Diese EEM-Applets können auch manuell aufgerufen werden. Dazu muss das Applet **event none** konfigurieren. Um ein Applet manuell auszuführen, geben Sie den Befehl **event manager run** gefolgt vom Namen des Applets ein. Wenn das Applet für einen Ereignisauslösungsmechanismus konfiguriert ist, der von 'none' abweicht, generiert der Versuch, es manuell auszuführen, einen Fehler. Wenn Sie eines der vorherigen Beispiele, 'depletblock', verwenden, sehen Sie:

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

Beispiel für ein manuelles Ereignis

Manuelle Ereignisse können ähnlich wie Makros verwendet werden. Beispielsweise kann ein manuelles Ereignis verwendet werden, um einige Befehle in der Reihenfolge auszuführen. In diesem Beispiel wird die Konfiguration gespeichert, ein Host gepingt und alle Shuns gelöscht.

```
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
```

Crash-Ereignis

Das **Crashinfo**-Ereignis löst ein Applet aus, wenn ein Absturz auf der ASA auftritt. Unabhängig vom Wert des Befehls **output** werden die **Action-**Befehle an die Datei crashinfo weitergeleitet. Die Ausgabe wird generiert, bevor der **show tech** Teil des Crashinfo generiert wird.

> **Warnung**: Wenn die ASA abstürzt, ist der Status der Box allgemein unbekannt. Einige CLI-Befehle sind möglicherweise nicht sicher auszuführen, wenn sich die Einheit in diesem Zustand befindet.

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

Aktionskonfiguration

Wenn das Applet ausgelöst wird, werden die Aktionen im Applet ausgeführt. Jede **Aktion** verfügt über eine Anordnung, mit der die Reihenfolge der Aktionen festgelegt wird. Mehrere Aktionen können pro Applet konfiguriert werden. aber jede Ordinalzahl kann nur einmal verwendet werden. Bei den Befehlen handelt es sich um typische CLI-Befehle, z. B. **Anzeigeblöcke**. Die Kostenvoranschläge werden dringend empfohlen, sind aber nicht erforderlich.

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

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

Der Wert der Action Identifier *<n>* liegt zwischen 0 und 4294967295. Der Wert von *<command>* muss angegeben werden. Andernfalls tritt ein Fehler auf, wenn der Befehl aus mehr als einem Wort besteht. Der Befehl wird im Konfigurationsmodus als Benutzer mit der Berechtigungsstufe 15 ausgeführt (der höchste Wert). Der Befehl akzeptiert möglicherweise keine Eingaben. da Eingaben deaktiviert werden, wenn ein Befehl die Option **noconfirm** besitzt. Dies sollte verwendet werden, da die Befehle nicht interaktiv verarbeitet werden.

Ausgabekonfiguration

Die Ausgabe der Aktionen kann über den Befehl **output** an einen angegebenen Speicherort weitergeleitet werden. Es kann jeweils nur ein Ausgabewert aktiviert werden. Der Standardwert ist **"output none"**. Dieser Wert verwirft alle Ausgaben der Aktionsbefehle.

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

Der Befehl der **Ausgabekonsole** sendet die Ausgabe der Aktionsbefehle an die Konsole.

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

Der Befehl Ausgabedatei leitet die Ausgabe der Aktionsbefehle an Dateien weiter. Es gibt vier Optionen, die verwendet werden können. Die **neue** Option schreibt die Ausgabe des Applets für jeden Aufruf in eine neue Datei. Der *Dateiname* hat das Format **eem-*<applet>*-*<timestamp>*.log**. wobei *<applet>* der Name des Applets und *<timestamp>* ein Zeitstempel im Format *YYYMMTT-hmmss* ist.

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

Die **Rotation**-Option wird verwendet, um eine Gruppe von Dateien zu erstellen, die ähnlich wie der Rotation von Protokolldateien durch Linux gedreht werden. Das Format für den Dateinamen ist **eem-*<applet>*-*<x>*.log**. Dabei ist *<applet>* der Name des Applets und *<x>* die Dateinummer. Die neueste Datei wird durch die Nummer 0 (Null) und die älteste Datei durch die höchste Zahl (*<n>*-1) angezeigt. Wenn eine neue Datei geschrieben werden soll, wird die älteste Datei gelöscht und alle nachfolgenden Dateien werden neu nummeriert, bevor die 0. Datei geschrieben wird.

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

Der Drehwert *<n>* liegt zwischen 2 und 100.

Die **Überschreiboption** wird verwendet, um immer die Ausgabe des Action-Befehls in eine einzelne Datei zu schreiben, die jedes Mal abgeschnitten wird.

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

Die Option **anhängen** wird verwendet, um immer die Ausgabe des Aktionsbefehls in eine einzelne Datei zu schreiben, diese Datei wird jedoch jedes Mal an eine Datei angehängt.

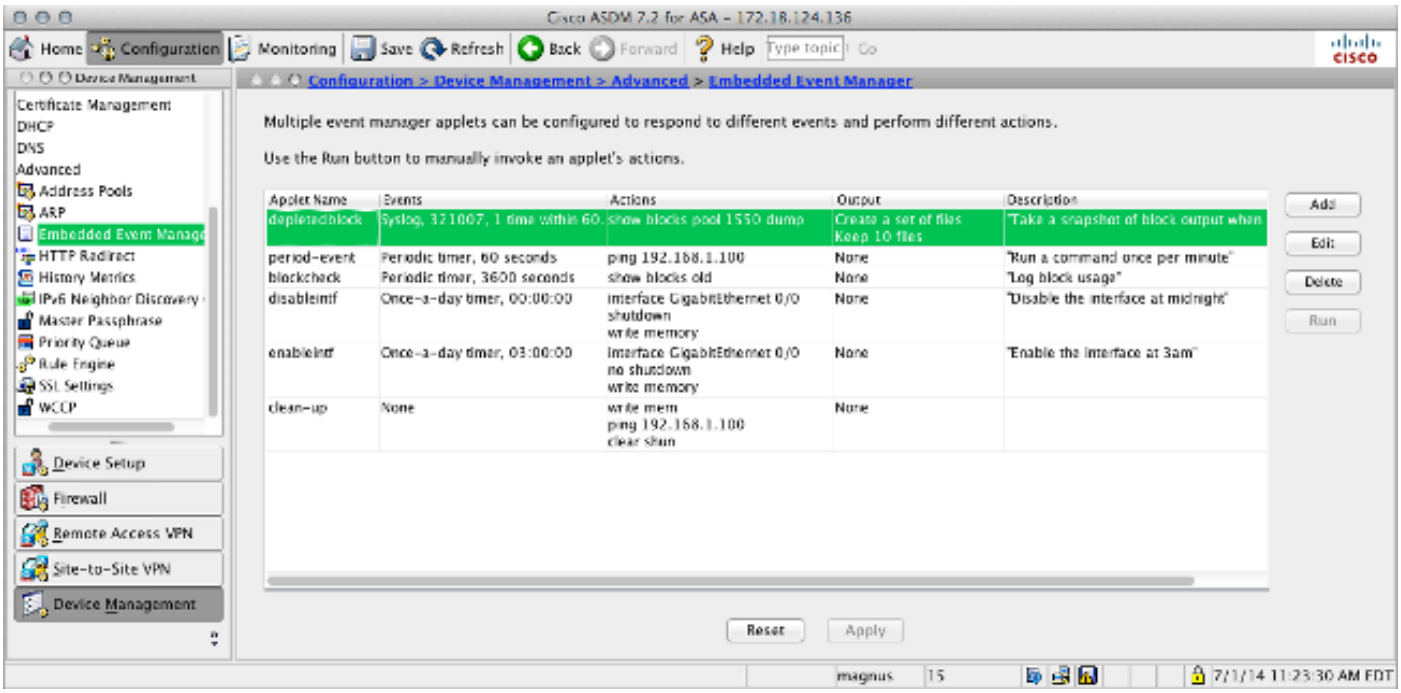
```
ASA(config-applet)# [no] output file append
```

Das *<Dateiname>*-Argument ist ein lokaler Dateiname (im ASA-Format). Der Befehl overwrite kann auch **ftp**:, **tftp:** und **smb:** Zieldateien.

ASDM-Konfiguration

EEM kann auch innerhalb des ASDM konfiguriert werden. Wählen Sie **Configuration > Device Management > Advanced > Embedded Event Manager aus**. In diesem Abschnitt des ASDM können Sie Ihre EEM-Applets mit den zuvor beschriebenen Parametern konfigurieren. Nachdem

Sie ein Applet konfiguriert haben, klicken Sie auf **Apply**, um die Konfiguration auf die ASA zu übertragen.



Überprüfen

Exec-Modus-Befehle

In diesem Abschnitt überprüfen Sie, ob Ihre Konfiguration ordnungsgemäß funktioniert.

Alle diese Befehle werden im exec-Modus verwendet.

Dieser Befehl zeigt die aktuelle Konfiguration des Event Manager-Systems an.

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

Dieser Befehl führt ein Ereignismanager-Applet aus, das mit **event none** konfiguriert wurde. Wenn Sie ein Applet ausführen, das nicht mit **Ereignis none** konfiguriert wurde, wird ein Fehler ausgegeben.

```
ASA# event manager run
```

```
      Dieser Befehl zeigt Informationen über die konfigurierten Applets an, die Trefferzählungen
enthalten und den Zeitpunkt, zu dem das Applet zuletzt aufgerufen wurde.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 10:51:52Der Ereignismanager
verwendet die Standardzähler. Aufgrund von Einschränkungen innerhalb dershow counter CLI wird
das eem-Schlüsselwort für die Protokollfilterung verwendet.
```

ASA# show counters protocol eem Das Output Interpreter Tool (nur registrierte Kunden) unterstützt bestimmteshow-Befehle. Verwenden Sie das Output Interpreter Tool, um eine Analyse der Ausgabe des Befehls show anzuzeigen.

DebuggenGeben Sie diese Befehle ein, um den EEM zu debuggen und die Ausgabe anzuzeigen.Hinweis: Weitere Informationen zu Debug-Befehlen vor der Verwendung von Debug-Befehlen finden Sie unterWichtige Informationen. ASA# [no] debug event manager

ASA# show debug event managerFehlerbehebungFür diese Konfiguration sind derzeit keine spezifischen Informationen zur Fehlerbehebung verfügbar. Wenn der Fehler nicht wie erwartet funktioniert, können Sie mithilfe der im vorherigen Abschnitt aufgeführten Schritte zum Debuggen und Überprüfen feststellen, ob ein Fehler aufgetreten ist.