

PANDORAFMS
E N T E R P R I S E

Pandora FMS

Manual de Administrador

Monitorización PerfCounter



Manual de Administración Monitorización PerfCounter

© Artica Soluciones Tecnológicas 2005-2012

Indice

| | |
|---|---|
| 1Changelog..... | 3 |
| 2Introducción..... | 4 |
| 3Requisitos..... | 5 |
| 4Matriz De Compatibilidad | 6 |
| 5Módulos de agentes software generados..... | 7 |
| 6Instalación..... | 8 |
| 7Monitorización | 9 |



1 CHANGELOG

| Fecha | Autor | Cambio | Versión |
|----------|-------|---------------|---------|
| 02/03/12 | Tomas | First Version | v1r1 |

2 INTRODUCCIÓN

Este documento tiene como objetivo la descripción de la monitorización masiva de contadores de rendimiento en entornos Windows.

Para la extracción de la información se utiliza:

- Consola Powershell (instalado por defecto en Windows Server 2008 R2, y Windows 7. Disponible desde Windows XP SP2 en adelante)
- Una interfaz “abierta” (la de Pandora, como extensión de la sección de administración) para especificar consultas SQL libres.
- El sistema, que se integra con el agente Windows.

Cabe destacar que el plugin de monitorización de Performance Counters se puede usar para recoger información de tipo numérico (para hacer gestión del rendimiento).

3 REQUISITOS

Los requisitos para que funcione correctamente esta monitorización son los siguientes:

- Instalar el agente de Pandora FMS en la versión 3.2.1 o superior.
- Una consola Powershell para ejecutar el plugin. Por defecto viene instalada en sistemas Windows Server 2008 R2 y Windows 7, pero debe ser descargada para versiones anteriores de Windows. Powershell no es compatible con sistemas Windows XP SP1 e inferiores.
- Es necesario que el usuario con el que se ejecuta el agente de Pandora FMS, que es el usuario que ejecutará el plugin, disponga de los siguiente permisos del sistema:
 - Administrador Local
- La política de ejecución de scripts Powershell debe ser establecida como RemoteSigned o inferior:

```
Set-ExecutionPolicy RemoteSigned
```

- El plugin obtendrá automáticamente la información de todos los contadores que le hayamos especificado en una lista en el archivo *counters.txt* y generará un módulo por cada uno en Pandora.

4 MATRIZ DE COMPATIBILIDAD

La matriz de compatibilidad del agente es la siguiente

| | |
|-------------------------------------|--|
| Sistemas donde se ha probado | <ul style="list-style-type: none"> • Windows XP SP2 • Windows Server 2003 • Windows Server 2008 • Windows Vista • Windows 7 |
|-------------------------------------|--|

| | |
|---|--|
| Sistemas donde debería funcionar | <ul style="list-style-type: none"> • Mismo Sistema o superior |
|---|--|

Según el idioma del sistema, el formato de los contadores a monitorizar pueden cambiar, con lo que sería necesario adaptar el fichero counters.txt en función de dichas circunstancias.

5 MÓDULOS DE AGENTES SOFTWARE GENERADOS

La verificación de contadores de rendimiento se realiza mediante el plugin Pandora_Plugin_PerfCounter.ps1 y puede ser aplicada en diferentes políticas para diferentes tecnologías, cada una con diferentes listados de contadores en función de qué queramos monitorizar de cada una de dichas tecnologías. Las tecnologías a monitorizar se incluirán en el archivo counters.txt o bien si no lo utilizamos nos cargará los contadores que Get-Service nos carga por defecto.

6 INSTALACIÓN

Copiar el plugin al directorio de plugins del agente, distribuyéndolo mediante file collections. Lo mismo con el listado de contadores. La llamada desde el agente será similar a esta, pero usando los paths donde esté instalado el plugin y el listado.

```
module_plugin "<ruta-powershell>\powershell.exe" -command C:\<ruta-  
plugin>\Pandora_Plugin_PerfCounter_vx.y.ps1' -list C:\<ruta-listado>\counters.txt'  
2> counter_plugin.error
```

Si la llamada que utilizamos es:

```
module_plugin "<ruta-powershell>\powershell.exe" -command C:\<ruta-  
plugin>\Pandora_Plugin_PerfCounter_vx.y.ps1' -list none 2> counter_plugin.error
```

Entonces el plugin devolverá en formato XML la información extraída de los contadores que el cmdlet Powershell Get-Counter provee por defecto.

7 MONITORIZACIÓN

Partiendo de la base de que ya tenemos instalado y configurado tanto Pandora como Powershell, vamos a explicar como obtener información sobre el estado de los discos duros montados en las máquinas a monitorizar.

Para este caso instalaremos tanto un agente software de Pandora como el plugin PerfCounter en la máquina en cuestión.

Es muy importante que la política de ejecución de scripts Powershell se establezca como RemoteSigned o inferior de la siguiente forma:

```
Set-ExecutionPolicy RemoteSigned
```

Para hacer que el agente software de Pandora que tenemos instalado en nuestro servidor a monitorizar ejecute dicho script, debemos editar el archivo de configuración del agente y hacer la llamada al plugin mediante el token de configuración **module_plugin**.

```
module_plugin "<ruta-powershell>\powershell.exe" -command C:\'<ruta-  
plugin>\Pandora_Plugin_PerfCounter_vx.y.ps1' -list C:\'<ruta-listado>\counters.txt'  
2> counter_plugin.error
```

Guardamos el archivo y reiniciamos el agente de Pandora.

Para evitar incrementar el log sin control y aun así recibir todos los errores ocurridos durante la ejecución del plugin en el último intervalo (por si hay algún error real), hacer la redirección usando el símbolo “2>” tal como viene especificado en la línea que se debe introducir al archivo de configuración.

Una de las característica mas potentes del plugin en Powershell es la posibilidad de especificar en vez de ir creando uno a uno módulos para cada contador de rendimiento, seleccionar todos los contadores especificados en una lista, para que el plugin haga una única comprobación y genere de forma automática un módulo para todos los estos contadores, optimizando lo máximo posible el tiempo necesario para extraer toda la información. Dicha lista la deberemos situar en la misma carpeta donde se encuentra el plugin, con el nombre **counters.txt** Veamos un ejemplo de su contenido:

```
\Web Service(*)\Total Bytes Sent  
\Web Service(*)\Bytes Sent/sec  
\Web Service(*)\Total Bytes Received
```

En base a este cmdlet, podemos añadirlo después de cualquier otro cmdlet que nos proporcione estadísticas, precedido de una tubería (|), y nos dará información de **todas** las propiedades que contiene dicho cmdlet, a pesar de que al ejecutar el primer cmdlet de forma general sin usar parámetros, sólo nos devolviera una lista predeterminada de info.

De esta forma nuestras posibilidades de monitorización haciendo uso de Powershell se amplía notablemente.

En caso de que quisiéramos añadir nuevos módulos a nuestro plugin, antes de hacer nada, probar a ejecutar el cmdlet del cual queremos sacar información junto con el mencionado, para así sacar toda la información posible.

Un ejemplo del uso de este comando sería el siguiente:

```
Get-Service | Select-Object -Property *
```

Normalmente el resultado del cmdlet Get-Service sería un listado en formato tabla de todos los servicios con su descripción y su estado. Sin embargo, al aplicarle este segundo cmdlet, conseguimos para cada servicio información de todas las propiedades de las que dispone ese servicio:

```
Name           : service
RequiredServices : {service1, service2}
CanPauseAndContinue : False
CanShutdown     : True
CanStop         : True
DisplayName      : This is a Windows Service
DependentServices : {service3}
MachineName     : .
ServiceName     : service
ServicesDependedOn : {service1, service2}
ServiceHandle    : SafeServiceHandle
Status          : Stopped
ServiceType     : Win32ShareProcess
Site            :
Container       :
```

NOTA: Es extremadamente importante tener en cuenta que los archivos pensados para el plugin en WINDOWS deben estar editados y almacenados con retornos de carro tipo “WINDOWS” y que si se usan retornos de carro tipo “UNIX” el plugin no funcionará adecuadamente.