Monitorizar la llegada de un evento al registro

Problemática

Si tenemos alteraciones del sueño porque los problemas de seguridad informática nos impiden dormir, incluso a veces nos invaden pesadillas durante la noche y nos despiertan sobresaltados. Si esto también le ocurre, entonces lea atentamente lo siguiente…

Con la esperanza de detectar una intrusión en el sistema o simplemente para saber si nuevos administradores de dominio han sido nombrados sin su conocimiento, puede ser especialmente interesante vigilar las adiciones de cuentas al grupo «Admins del dominio». Por consiguiente, deseamos estar informados por e-mail tan pronto como una adición de este tipo se produzca, y en la medida de lo posible, ¡en tiempo real!

Solución

Escribir un script básico sobre los eventos WMI que funcionará en un bucle sin fin. Éste monitorizará la llegada de un evento especial en el registro de seguridad. Este evento es el evento de adición en el interior de un grupo global de seguridad y tiene el ID 632. En efecto, cuando una modificación de un grupo de seguridad tiene lugar, y si la política de auditoría está activada, entonces los eventos se consignan automáticamente al registro de seguridad del/de los controlador(es) de dominio. Este script deberá entonces funcionar preferentemente sobre un controlador de dominio.

Debemos procurar que el correo electrónico se envié únicamente en el caso de que se modifique el grupo «Admins del dominio» y sólo éste grupo, para evitar recibir un bombardeo de mensajes.

Veamos el script:

```
# Watch-AdminGroup.ps1
$strComputer = '.'
$query = New-Object System.Management.WQlEventQuery `
            "SELECT * FROM __InstanceCreationEvent
             WITHIN 30
             WHERE Targetinstance ISA 'Win32_NTLogEvent'
             AND TargetInstance.EventCode = '632'"

$scope =
   New-Object System.Management.ManagementScope "\\$strComputer\root\cimv2"
$watcher =
   New-Object System.Management.ManagementEventWatcher $scope,$query
$watcher.Start()

while ($true)
{
   $event=$watcher.WaitForNextEvent()

   if ($($event.TargetInstance.Message) -match 'Admins del dominio')
   {
     # envío de un correo
     $emisor = 'moreno@ps-scripting.com'
     $destinatario = 'oscar@ps-scripting.com'
     $servidor = 'mailhost.ps-scripting.com'
     $asunto = '¡Alerta: Adición de un miembro al grupo Admins del dominio!'

     $cuerpo = "$($event.TargetInstance.User)`n"
     $cuerpo += "$($event.TargetInstance.TimeWritten)`n"
     $cuerpo += "$($event.TargetInstance.Message)"
```

```
    $mensaje =
        new-object System.Net.Mail.MailMessage $emisor, $destinatario,
          $asunto, $cuerpo
    $cliente = new-object System.Net.Mail.SmtpClient $servidor
    $cliente.Credentials =
        [System.Net.CredentialCache]::DefaultNetworkCredentials
    $cliente.Send($mensaje)
  }
}
```

Veamos cual será el contenido de un mensaje enviado cuando se detecta un evento 632 en el registro de seguridad que contenga la palabra clave «Admins del dominio»:

```
PS-SCRIPTING\Administrador
20071112011950.000000+060
Miembro del grupo global de seguridad activada añadido:

  Nombre del miembro: CN=Jose Bar,OU=Usuarios,DC=ps-scripting,DC=com

  Id. del miembro: PS-SCRIPTING\JoseBar

  Nombre de la cuenta objetivo: Admins del dominio

  Dominio objetivo: PS-SCRIPTING

  Id. de la cuenta objetivo: PS-SCRIPTING\Admins del dominio

  Usuario consultado: administrador

  Dominio consultado: PS-SCRIPTING

  Id. de sesión de la consulta: (0x0,0x28655)

  Privilegios: -
```

Mejora del resultado

Podemos observar el carácter especial de la fecha devuelta en la segunda línea del resultado. Ésta tiene un formato fecha WMI que convendría reformatear, por ejemplo, como se muestra a continuación:

```
PS > $dateWMI = '20071112011950.000000+060'
PS > $OFS = ''
PS > $date = New-Object system.datetime (
                [string]$dateWMI[0..3],[string]$dateWMI[4..5],
                [string]$dateWMI[6..7],[string]$dateWMI[8..9],
                [string]$dateWMI[10..11],[string]$dateWMI[12..13])
PS > $date

lunes 12 noviembre 2007 01:19:50
```

Hemos creado un objeto de tipo **`system.datetime`** en el que hemos pasado al constructor esta clase de subcadenas que corresponde a cada parte del contenido de **`$dateWMI`**. Sin embargo, para que eso funcione correctamente hay en realidad dos trucos:

- cuando extraemos una subcadena de la variable **`$dateWMI`** con **`$dateWMI[0..3]`**; esto equivale a pedir sucesivamente el valor del índice 0, después el del índice 1 hasta el índice 3. Esta operación nos devolverá los valores siguientes:

```
PS > $dateWMI[0..3]
2
0
0
7
```

- Después debemos convertir este resultado en tipo **string** con el fin de pasar este valor al constructor del objeto **system.datetime**. Lo hacemos forzando el resultado en una cadena, como a continuación: **[string]$dateWMI[0..3]**

Sólo que este comando no nos devolverá el resultado esperado. Probémoslo para ver:

```
PS > [string]$dateWMI[0..3]
2 0 0 7
```

Tocamos nuestro objetivo, pero ¿por qué tenemos un espacio entre cada carácter? Pues sencillamente porque existe en PowerShell, la variable especial **$OFS**. Ésta entra en juego cuando se trata de convertir una tabla en una cadena. Y por defecto esta variable contiene el carácter blanco o espacio. Vamos a sustituir por lo tanto su valor por "" lo que tendrá por efecto eliminar los blancos. Consideremos esto más detalladamente:

```
PS > $OFS = ''
PS > [string]$dateWMI[0..3]
2007
```

Verá que puede entretenerse introduciendo cualquier valor en **$OFS**, por ejemplo:

```
PS > $OFS = ' '
PS > [string]$dateWMI[0..3]
2 0 0 7
```

Mejora del resultado (bis)

Hemos descubierto hace poco tiempo, que podíamos simplificarnos la vida en lo que respecta al formateo de las fechas WMI. En efecto, existe un método estático del Framework .Net llamado «ToDateTime» perteneciente a la clase «System.Management.ManagementDateTimeConverter». Así podemos ahora simplificar nuestro script escribiendo lo siguiente:

```
PS > $dateWMI = '20071112011950.000000+060'
PS > [System.Management.ManagementDateTimeConverter]::ToDateTime($dateWMI)

lunes 12 noviembre 2007 01:19:50
```

El resultado devuelto es de tipo DateTime, podremos formatearlo a nuestra voluntad (véase el capítulo Control del Shell, sección Las fechas - Los formatos).