Comunicaciones remotas WSMAN/WinRM con WMI

La tecnología WMI se basa en las tecnologías COM y DCOM, hecho de hace que ésta no sea muy *firewall-friendly*. Para tratar de ser más amistosa ante los cortafuegos, el protocolo WinRM se encarga del enrutamiento de las peticiones WMI encapsulándolas en tramas que transitarán por los puertos 5985/ 5986 y 80/443 si se utiliza la conexión vía una dirección URI (véase más adelante).

Los ejemplos que proporcionamos se realizan en una máquina con Windows 7 x64 que sirve de cliente y otra máquina con Windows Server 2008 R2 que sirve de máquina controlada, pero podríamos haberlo hecho a la inversa. Estas dos máquinas se encuentran en un mismo dominio y el firewall está activado en cada una de ellas. No hemos configurado normas específicas en el firewall, sólo hemos ejecutado el comando `Enable-PSRemoting` del que hemos debatido en el apartado anterior de este capítulo.

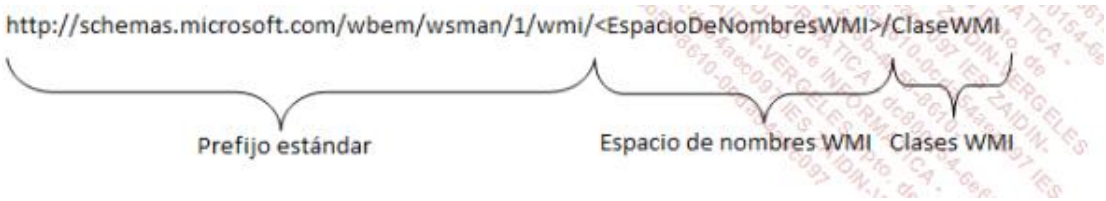
WinRM responde a los estándares de la Web, y por lo tanto todos los recursos a los que se dirige deben ajustarse a un cierto formalismo. Cada recurso WMI en el mundo WinRM está representado por un URI (*Uniform Resource Identifier*).

1. Identificar un recurso WMI con los URIs

WinRM soporta la mayoría de las clases WMI y operaciones sobre éstas. WinRM por lo tanto necesita de un mecanismo que le permita identificar todos los recursos del sistema para poder actuar sobre éstos a través de WMI. En otras palabras, esto significa que vamos a poder obtener información o actuar sobre los objetos tales como los discos, los procesos, los servicios o las tarjetas de redes a través de las clases WMI que ya conocemos.

Son los URIs quienes van a permitirnos establecer el vínculo entre el protocolo WS-Management y las clases WMI. Los URIs WMI están definidos directamente en el esquema WS-Management.

Un URI es la concatenación de un prefijo y del espacio de nombre WMI, como se muestra a continuación:



Lo que necesitamos saber para construir un URI:

Un URI WMI empieza siempre por: http://schemas.microsoft.com/wbem/wsman/1

El espacio de nombres WMI tiene el formato `wmi/root`, `wmi/root/cimv2` (el más común), `wmi/root/microsoft`, `wmi/root/directory`, etc.

Finalmente será necesario añadir al final del URI el nombre de una clase WMI.

Ejemplos de URIs:

```
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_CurrentTime
http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Processor
```

El 1 del prefijo estándar de la URI está para indicar que se trata de la versión 1 del protocolo WS-Management.

2. El conjunto de comandos PowerShell

PowerShell v2 está dotado de un conjunto de comandos específicos que simplifican el acceso a la administración con

WSMAN. Se puede dividir en dos categorías: una para realizar operaciones y otra para la configuración de las sesiones WSMAN.

Comandos WSMan orientados a operación

Comando	Descripción
`Test-WSMan`	Verifica si el servicio WinRM está bien iniciado.
`Get-WSManInstance`	Muestra información de administración para una instancia de recurso especificada por un URI de recurso.
`Set-WSManInstance`	Modifica la información de administración relacionada con un recurso.
`New-WSManInstance`	Crea una nueva instancia de un recurso de administración.
`Remove-WSManInstance`	Elimina una instancia de un recurso de administración.
`Invoke-WSManAction`	Invoca una acción en el objeto especificado por el URI de recurso y por los selectores.

Comandos WSMan orientados a configuración

Comando	Descripción
`Connect-WSMan`	Produce la conexión con el servicio WinRM en un equipo remoto.
`Disconnect-WSMan`	Desconecta el cliente del servicio WinRM en un equipo remoto.
`New-WSManSessionOption`	Crea la tabla hash de opciones de sesión WSMan que se van a utilizar como parámetros de entrada en los siguientes cmdlets WSMan: Connect-WSMan, Get-WSManInstance, Invoke-WSManAction, Set-WSManInstance.
`Set-WSManQuickConfig`	Configura el equipo local para la administración remota.
`Get-WSManCredSSP`	Obtiene la configuración relacionada con el proveedor de servicios de seguridad de credenciales CredSSP (*Credential Security Service Provider*) para el cliente.
`Enable-WSManCredSSP`	Habilita la autenticación CredSSP (proveedor de servicios de seguridad de credenciales) en un equipo cliente.
`Disable-WSManCredSSP`	Deshabilita la autenticación CredSSP (proveedor de servicios de seguridad de credenciales) en un equipo cliente.

3. Configuración del sistema

En primer lugar conviene comprobar si el servicio WinRM está en ejecución. Para ello usamos el comando `Test-WSMan`:

```
PS > Test-WSMan -Authentication default

wsmid      : http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd
ProtocolVersion : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
```

```
ProductVendor   : Microsoft Corporation
ProductVersion  : OS: 6.1.7600 SP: 0.0 Stack: 2.0
```

El resultado indica que la versión instalada de WinRM es la versión 2.0.

Para configurar el sistema es necesario utilizar como mínimo el commandlet `Set-WSManQuickConfig`. Decimos como mínimo, porque si ya ha configurado su sistema para utilizar las comunicaciones remotas PowerShell (con el comando `Enable-PSRemoting`) entonces este paso no es necesario.

a. Listar los servicios de una máquina remota

Como el resultado será muy difuso, con la ayuda de `Select-Object` y del parámetro `-first 1`, nos limitaremos a la recuperación del primer objeto.

```
PS > $URI =
'http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/Win32_Service'
PS > Get-WSManInstance -ResourceURI $URI -computer w2K8R2VM -Enumerate |
Select-Object -first 1


xsi                     : http://www.w3.org/2001/XMLSchema-instance
p                       : http://schemas.microsoft.com/wbem/wsman/1/wmi/
                          root/cimv2/Win32_Service
cim                     : http://schemas.dmtf.org/wbem/wscim/1/common
type                    : p:Win32_Service_Type
lang                    : es-ES
AcceptPause             : false
AcceptStop              : true
Caption                 : Active Directory Web Services
CheckPoint              : 0
CreationClassName       : Win32_Service
Description             : This service provides a Web Service interface
                          to instances of the directory service (AD DS
                          and AD LDS) that are running locally on this
                          server. If this service is stopped or disabled,
                          client applications, such as Active Directory
                          PowerShell, will not be able to access or
                          manage any directory service instances that are
                          running locally on this server.
DesktopInteract         : false
DisplayName             : Active Directory Web Services
ErrorControl            : Normal
ExitCode                : 0
InstallDate             : InstallDate
Name                    : ADWS
PathName                : C:\Windows\ADWS\Microsoft.ActiveDirectory.
                          WebServices.exe
ProcessId               : 1372
ServiceSpecificExitCode : 0
ServiceType             : Own Process
Started                 : true
StartMode               : Auto
StartName               : LocalSystem
State                   : Running
Status                  : OK
```

```
SystemCreationClassName : Win32_ComputerSystem
SystemName              : W2K8R2VM
TagId                   : 0
WaitHint                : 0
```

4. Determinación de la fecha de instalación de una máquina remota

```
PS > $u=
'http://schemas.microsoft.com/wbem/wsman/1/wmi/root/cimv2/
Win32_OperatingSystem'
PS > Get-WSManInstance -ResourceURI $u -computer w2K8R2VM -Enumerate
```

Restringiremos voluntariamente la visualización a ciertas propiedades ya que verdaderamente son muy numerosas.

```
...
type                          : p:Win32_OperatingSystem_Type
lang                          : es-ES
BootDevice                    : \Device\HarddiskVolume1
BuildNumber                   : 7600
BuildType                     : Multiprocessor Free
Caption                       : Microsoft Windows Server 2008 R2 Enterprise
CodeSet                       : 1252
CountryCode                   : 33
CreationClassName             : Win32_OperatingSystem
CSCreationClassName           : Win32_ComputerSystem
CSDVersion                    : CSDVersion
CSName                        : W2K8R2VM
CurrentTimeZone               : 60
EncryptionLevel               : 256
FreePhysicalMemory            : 86584
FreeSpaceInPagingFiles        : 781408
FreeVirtualMemory             : 709956
InstallDate                   : InstallDate
LastBootUpTime                : LastBootUpTime
LocalDateTime                 : LocalDateTime
Locale                        : 040c
Manufacturer                  : Microsoft Corporation
MaxNumberOfProcesses          : 4294967295
MaxProcessMemorySize          : 8589934464
MUILanguages                  : en-US
SerialNumber                  : 55041-507-0304761-12345
ServicePackMajorVersion       : 0
ServicePackMinorVersion       : 0
SizeStoredInPagingFiles       : 1048576
Status                        : OK
Version                       : 6.1.7600
WindowsDirectory              : C:\Windows
...
```

La propiedad `InstallDate` todavía no está accesible, vamos a conseguirla de la manera siguiente:

```
PS > $result = Get-WSManInstance -ResourceURI $u -computer w2K8R2VM
-Enumerate
PS > $result.InstallDate
```

```
Datetime
--------
2009-10-11T16:38:18+02:00
```

Este resultado, aunque legible, no es necesariamente el más adaptado. Pero podemos fácilmente transformarlo en la medida en que se trate de un formato reconocido por el tipo **`DateTime`**. Forzamos entonces la conversión de este valor en tipo **`DateTime`** y observemos el resultado:

```
PS > [DateTime]$result.InstallDate.DateTime

domingo 11 octubre 2009 16:38:18
```

Perfecto, hemos logrado enviar el equivalente de una solicitud WMI a una máquina remota, pasando a través de los firewalls por el puerto HTTP y hemos recuperado el contenido. HTTP, como le dijimos anteriormente, no es seguro pues los intercambios pasan en claro por la red (excepto la autentificación). Le recomendamos encarecidamente si tiene que utilizarWinRM en un medio hostil, que active el puerto de escucha por el protocolo de transporte HTTPS y que despliegue un certificado servidor.