

# Encontrar las cuentas de equipos caducadas en AD DS

## Problemática

Existen a menudo un gran número de cuentas de equipos innecesarias que están presentes en el servicio de directorio Active Directory. El razón es sencilla: en muchas empresas falta un procedimiento de eliminación del material, o si existe para la gestión del material físico no hay nada previsto para eliminar las cuentas de equipos. Así al cabo de unos pocos años, no es raro tener un exceso del 50% de las cuentas de equipos. Por consiguiente, puede llegar a ser difícil para un administrador gestionar su parque de ordenadores. Para tratar de poner un poco de orden en AD DS vamos a desarrollar un script que va a conectarse a un controlador de dominio y va a recuperar la lista de las cuentas de los equipos. Para cada cuenta de equipo observaremos cuál ha sido la fecha del último logon o, dicho de otro modo, la fecha de la última apertura de sesión. Pues sí, ¡incluso las cuentas de equipos abren sesiones! Una cuenta de un equipo abre una sesión en un dominio autenticándose en un controlador de dominio, al igual que un usuario. Con la diferencia de que las contraseñas de cuentas de equipos son autogestionadas. La contraseña se genera aleatoriamente la primera vez que un ordenador se adhiere a un dominio, y luego se cambia automáticamente cada treinta días.

## Algunas dificultades a superar

Las propiedades de las cuentas de equipos están disponibles en los controladores de dominio, ¡aunque no todas! En efecto, la información del último logon no está replicada; por lo que reside localmente en cada controlador de dominio. Para tener la información actualizada se deberá consultar a todos los controladores y únicamente guardar la información más actualizada. Esto constituirá la primera dificultad, ¡si no hubiese sido demasiado sencillo!

La buena noticia, es que en un dominio Windows Server 2003 o 2008 tenemos a nuestra disposición un atributo llamado «LastLogonTimeStamp». Éste se replica entre todos los controladores de dominio, pero (como no, ¡tiene que haber un pero!) esta replicación sólo tiene lugar cada 14 días. Para responder a nuestra problemática no nos vendrá de dos semanas; por lo que aprobaremos esta técnica.

La segunda dificultad proviene del valor del atributo "LastLogonTimeStamp" ya que este valor es un entero codificado en 64 bits y no una simple fecha de tipo «10/02/2010». Hablamos precisamente de este tema en el capítulo Control del Shell, el valor devuelto es el número de intervalos de 10 millonésimas de segundos desde el 1º de Enero 1601, es decir... tendremos que convertir este valor en fecha, pero esta conversión ya sabemos cómo hacerla.

➤ Para obtener un valor óptimo de «LastLogonTimeStamp», es recomendable tener los controladores de dominio en versión Windows Server 2003 SP1 como mínimo. Además, el nivel funcional del dominio debe ser en modo nativo Windows Server 2003. La frecuencia de actualización de este atributo puede regularse modificando el atributo siguiente en el esquema: Object: DC=DomainName, Attribute: msDS-LogonTimeSyncInterval, Default value: 14 days.

## La solución:

```
# Get-MachineAccounts.ps1 - v1.0
[datetime]$date = '01/01/1601'

$objDominio = [ADSI]''
$objBusqueda = New-Object System.DirectoryServices.DirectorySearcher
($objDominio)
$consulta = '(&(sAMAccountType=805306369)(name=*))'
$objBusqueda.Filter=$consulta
$cuentas = $objBusqueda.FindAll()

$cuentas | select-object @{e={$_.properties.cn};n='Nombre común'},
@{e={$date.AddTicks($_.properties.lastlogontimestamp)};n='
Última conexión'},
@{e={$_.properties.operatingsystem};n='OS'},
@{e={$_.properties.operatingsystemversion};n='Versión'},
```

### Resultado:

```
PS > ./Get-MachineAccounts.ps1 | Format-Table -Autosize
```

Nombre común	Última conexión	OS	Versión	Service Pack
POWERSERVER	29/10/2007 21:02:38	Windows Server 2003	5.2 (3790)	Service Pack 2
PCXPBIS	05/11/2007 23:28:33	Windows XP Professional	5.1 (2600)	Service Pack 2
PCXP1	29/05/2007 22:51:31	Windows XP Professional	5.1 (2600)	Service Pack 2
PCVISTA	03/11/2007 13:37:16	Windows VistaT Editio...	6.0 (6000)	
SERVIDOR2008	03/11/2007 11:54:49	Windows Server® 2008	6.0 (6001)	Service Pack 1

### Algunas aclaraciones:

Empezamos por crear una variable de tipo fecha a la que asignaremos el valor de 1º Enero 1601. A continuación le añadimos el valor de «LastLogonTimeStamp» para obtener una fecha en un formato comprensible. Creamos posteriormente una consulta de búsqueda en AD DS. El valor dado a «sAMAccountType» corresponde al valor hexadecimal «0x30000001»; el que representa una cuenta de equipo. El resultado de la consulta se envía a la variable \$cuentas. Después, gracias a select-object creamos un nuevo objeto a partir del contenido recuperado vía la tubería, objeto que contendrá las propiedades «Nombre común», «Última conexión», «OS», «Versión» y «Service Pack». Por último, llamaremos al script indicándole que muestre su contenido en forma de tabla.

➤ Para demostrar que existe un gran número de información interesante además del «LastLogonTimeStamp» hemos aprovechado para también devolverla al mismo tiempo. Esta información puede resultar útil para, por ejemplo, hacer inventarios rápidos de los sistemas operativos instalados, así como sus versiones de Service Pack asociado.

➤ Para que este script pueda funcionar en un máximo de plataformas hemos realizado la búsqueda en Active Directory Domain Services basándonos únicamente en ADSI. Este script funciona también con la versión 1 de PowerShell.

He aquí lo que podría ser el script si se utilizan los nuevos commandlets proporcionados en el módulo Active Directory de Windows Server 2008 R2 :

```
# Get-MachineAccounts.ps1 - v2.0
[datetime]$date = '01/01/1601'

$cuentas = Get-ADComputer -filter * `
    -properties name,LastLogonTimeStamp,OperatingSystem,
        OperatingSystemVersion,OperatingSystemServicePack

$cuentas | select-object @{e={$_.name};n='Nombre común'},
    @{e={$date.AddTicks($_.lastlogontimestamp)};n='Última conexión'},
    @{e={$_.operatingsystem};n='OS'},
    @{e={$_.operatingsystemversion};n='Versión'},
    @{e={$_.operatingsystemservicepack};n='Service Pack'}
```

### Posible mejora:

Mejoraremos nuestro script para que podamos pasarle un parámetro de caducidad de la cuenta. Así, gracias a este parámetro, nuestro script sólo nos devolverá las cuentas de equipos que no se hayan conectado al dominio desde hace más de n días.

Para ello, será necesario definir un filtro con la cláusula `Where-object`. Si la expresión contenida en el bloque de script es verdadera, entonces el objeto tratado pasa a lo largo de la tubería. En otras palabras, si la evaluación es verdadera, entonces es que hemos obtenido un resultado.

He aquí el script modificado:

```
# Get-MachineAccounts.ps1 - v1.1
param ($NoConectadoDespuesNDias=0)

[datetime]$date = '01/01/1601'

$ObjDominio = [ADSI]''
$ObjBusqueda =
    New-Object System.DirectoryServices.DirectorySearcher($ObjDominio)
$consulta = '(&(sAMAccountType=805306369)(name=*))'
$ObjBusqueda.Filter=$consulta
$cuentas = $ObjBusqueda.FindAll()


$cuentas | select-object @{e={$_.properties.cn};n='Nombre común'},
    @{e={$date.AddTicks($_.properties.lastlogontimestamp)};
    n='Última conexión'},
    @{e={$_.properties.operatingsystem};n='OS'},
    @{e={$_.properties.operatingsystemversion};n='Versión'},
    @{e={$_.properties.operatingsystemservicepack};n='Service Pack'} |
Where-object {(new-timespan $_."Última conexión"
$(get-date)).days -ge $NoConectadoDespuesNDias
}
```

### Resultado:

```
PS > ./Get-MachineAccounts.ps1 60 | Format-Table -AutoSize

Nombre común  Última conexión      OS              Versión          Service Pack
-----
PCXP1         29/05/2007 22:51:31  Windows XP      5.1 (2600)       Service Pack 2
Professional
```

¡Obtenemos un resultado! En efecto con relación a la fecha del día, la cuenta del equipo «PCXP1» no se ha conectado desde al menos sesenta días. Pero dado que existe un delta de actualización de 14 días, en realidad podría ser que hiciese más de setenta días que la cuenta está inactiva.

 También hubiéramos podido, para abordar este problema, basarnos en la fecha del último cambio de contraseña, pero el delta hubiera sido esta vez de treinta días en lugar de catorce. En efecto, las contraseñas de cuentas de equipos cambian automáticamente cada treinta días.