poweshell

Powershell

Para ejecutar el powershell desde el cuadro buscar→ powershell o desde consola ejecutamos

```
Powershell
```

El prompt cambiará y aparecerá como **PS C:>**

Para saber la versión que tenemos de powershell Abrimos una sesión de powershell y ejecutamos

```
$PSVersionTable
```

El resultado será algo así

```
PS C:\Users\Administrador> $PSVersionTable
Name                           Value
----                           -----
PSVersion                      4.0
WSManStackVersion              3.0
SerializationVersion           1.1.0.1
CLRVersion                     4.0.30319.42000
BuildVersion                   6.3.9600.19170
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0}
PSRemotingProtocolVersion      2.2
```

También podemos usar **$PSVersionTable.PSVersion** para saber la versión y el número de actualización de la misma

En este caso es una versión vieja 4.0 que viene por defecto con un windows server 2012

Para actualizar la versión debemos de descargarla desde la página de Microsoft

Versión 6

Esta nueva versión de Powershell, no podemos actualizarla como con las versiones anteriores (las que van desde la 1.0 a la 5.1). Hay que quitarlas antes de proceder a instalar la versión 6, que hay que descargarla desde https://github.com/PowerShell/PowerShell

Sacar información del directorio activo

Lo primero es carga el módulo del AD desde el powershell con

```
import-module ActiveDirectory
```

y si queremos sacar un listado po pantalla de equipos de nuestro dominio

```
 Get-ADComputer -Filter * -Property * | Format-Table
Name,OperatingSystem,OperatingSystemServicePack,OperatingSystemVersion -Wrap
–Auto
```

que queremos el mismo listado para abrirlo desde un excel

```
 Get-ADComputer -Filter * -Property * | Select-Object
Name,OperatingSystem,OperatingSystemServicePack,OperatingSystemVersion |
Export-CSV AllWindows.csv -NoTypeInformation -Encoding UTF8
```

Equpos que no han cambiado la password en 30 días

```
    $d = [DateTime]::Today.AddDays(-30)
    Get-ADComputer -Filter ‘PasswordLastSet -lt $d’ -Properties
PasswordLastSet | FT Name,PasswordLastSet
```

Usuarios que no han cambiado su contraseña en tres 180 días

```
    $d = [DateTime]::Today.AddDays(-180)
    Get-ADUser -Filter ‘PasswordLastSet -lt $d’ -Properties PasswordLastSet
| FT Name,PasswordLastSet
```

Usuarios que no requieren password

```
Get-ADUser -Filter 'userAccountControl -band 32' | FT Name
```

Otro valores que podemos utilizar en vez de 32



```
Active Directory userAccountControl Values:

Normal Day to Day Values:
============================
512 – Enable Account
514 – Disable account
544 – Account Enabled – Require user to change password at first
logon
4096 – Workstation/server
66048 – Enabled, password never expires
66050 – Disabled, password never expires
262656 – Smart Card Logon Required
532480 – Domain controller

All Other Values:
============================
1 – script
```



```
2 – accountdisable
8 – homedir_required
16 – lockout
32 – passwd_notreqd
64 – passwd_cant_change
128 – encrypted_text_pwd_allowed
256 – temp_duplicate_account
512 – normal_account
2048 – interdomain_trust_account
4096 – workstation_trust_account
8192 – server_trust_account
65536 – dont_expire_password
131072 – mns_logon_account
262144 – smartcard_required
524288 – trusted_for_delegation
1048576 – not_delegated
2097152 – use_des_key_only
4194304 – dont_req_preauth
8388608 – password_expired
16777216 – trusted_to_auth_for_delegation
```

Lanzar la consola de powershell como administrador

```
start-process powershell -verb runas
```

Iniciar sesión en un equipo remoto

PowerShell está bloqueado de forma predeterminada, por lo que deberá habilitar PowerShell Remoting antes de usarlo.

Previamente en el equipo **destino** deberemos de tener habilitado la conexión remota. Para habilitarla :

```
Enable-PSRemoting -Force
```

Para conectarnos al equipo lanzamos el siguiente comando desde el equipo **origen**

```
Enter-PSSession miequipo.dominio.local -Credential USUARIO
```

Para salir

```
Exit-PSSession
```

Reinicio o apagado programado

https://www.sysadmit.com/2019/02/windows-programar-apagado-reinicio-powershell.html

referencias

- http://blogs.itpro.es/rtejero/2014/10/13/algunos-cmdlets-interesantes-de-directorio-activo-gracias-a-un-rap-as-a-service/
- http://blogs.technet.com/b/askds/archive/2010/02/04/inventorying-computers-with-ad-powershell.aspx
- http://www.joseangelfernandez.es/blog/category/programacion/powershell/
- http://blogs.technet.com/b/heyscriptingguy/archive/2013/11/23/using-scheduled-tasks-and-scheduled-jobs-in-powershell.aspx#
- http://blogs.itpro.es/rtejero/2014/04/02/instalar-powershell-4-0-en-nuestros-clientes-y-servidores-windows-management-framework-4-0/
- http://blogs.itpro.es/rtejero/2014/09/15/errores-de-netlogon-no_client_site-parsear-con-powershell/
- Using PowerShell to Remove Virtual Machine Snapshots in VMware ESXi 4.1 http://pipe2text.com/?page_id=1953
- https://respontodo.com/como-ejecutar-comandos-de-powershell-en-equipos-remotos/
- http://woshub.com/pswindowsupdate-module/
- http://woshub.com/using-powershell-behind-a-proxy/
- https://www.sysadmit.com/2021/11/windows-powershell-error-instalar-nuget.html
- https://reparar.info/pswindowsupdate-administracion-de-actualizaciones-de-windows-desde-powershell/
- https://blog.velingeorgiev.com/install-powershell-5-nuget-sharepointpnppowershellonline-behind-proxy

From:
https://wiki.intrusos.info/ - **LCWIKI**

Permanent link:
https://wiki.intrusos.info/doku.php?id=windows:powershell

Last update: **262023/04/ 10:15**