

Script automatizacion

ASO

Yeray Gutiérrez Mullor

Contenido

Información acerca del script.....	2
Definición de parámetros.....	2
Importar módulo de Active Directory	2
Configuración de Logs	2
Parámetros para crear historial de eventos.....	2
Crear Unidades Organizativas	3
Crear Grupos de Seguridad	3
Crear Usuarios	3
Información en los Logs.....	4

Información acerca del script

Detalles acerca del script, que hace, el autor, la versión y la fecha de creación/ultima modificación

```
1 <#  
2 .SYNOPSIS  
3 Crea automaticamente 3 UOs, 2 Grupos de Seguridad y 6 Usuarios utilizando Active Directory  
4  
5 .AUTHOR  
6 Yeray Gutierrez Mullor  
7  
8 .VERSION  
9 1.4  
10  
11 .DATE  
12 2026-01-11  
13 #>
```

Definición de parámetros

Hace que el script solicite el parámetro de la ruta del .csv donde esta la información necesaria para el script

```
15 param (  
16     [Parameter(Mandatory)]  
17     [string]$CsvPath  
18 )
```

Importar módulo de Active Directory

Añade las funciones de Active Directory para que PowerShell las reconozca y poder usarlas en el script

```
20 Import-Module ActiveDirectory
```

Configuración de Logs

Configura la ruta donde se crea el Log y el formato del nombre del archivo; Si la ruta no existe, la crea

```
22 $LogFile = "C:\Logs\AD_Deployment_$(Get-Date -Format 'yyyyMMdd_HH:mm').log"  
23 New-Item -ItemType Directory -Path (Split-Path $LogFile) -ErrorAction SilentlyContinue | Out-Null
```

Parámetros para crear historial de eventos

Establece los parámetros y el formato de la información que se mostrará en el Log por defecto, como la fecha y la hora

```

24 function Write-Log {
25     param ($Message)
26     $Time = Get-Date -Format "yyyy-MM-dd HH:mm:ss"
27     "$Time - $Message" | Out-File -FilePath $LogFile -Append
28 }

```

Crear Unidades Organizativas

Esta función crea las unidades organizativas mediante los datos en el .csv y si ya existen, no las crea

```

30 function Create-OU {
31     param ($Name, $Path)
32
33     if (-not (Get-ADOrganizationalUnit -Filter "Name -eq '$Name'" -SearchBase $Path -ErrorAction SilentlyContinue)) {
34         New-ADOrganizationalUnit -Name $Name -Path $Path
35         Write-Log "OU creada: $Name"
36     } else {
37         Write-Log "OU ya existe: $Name"
38     }
39 }
40

```

Crear Grupos de Seguridad

Esta función crea los Grupos de Seguridad mediante los datos en el .csv y si ya existen, no los crea

```

41 function Create-Group {
42     param ($Name, $Path, $Scope)
43
44     if (-not (Get-ADGroup -Filter "Name -eq '$Name'" -ErrorAction SilentlyContinue)) {
45         New-ADGroup -Name $Name -Path $Path -GroupScope $Scope -GroupCategory Security
46         Write-Log "Grupo creado: $Name"
47     } else {
48         Write-Log "Grupo ya existe: $Name"
49     }
50 }
51

```

Crear Usuarios

Esta función crea los Usuarios mediante los datos en el .csv y si ya existen, no los crea

```

52 function Create-User {
53     param ($User)
54
55     if (-not (Get-ADUser -Filter "SamAccountName -eq '$($User.SamAccountName)'" -ErrorAction SilentlyContinue)) {
56         New-ADUser `
57             -Name $User.Name `
58             -SamAccountName $User.SamAccountName `
59             -AccountPassword (ConvertTo-SecureString $User.Password -AsPlainText -Force) `
60             -Enabled $true `
61             -Path $User.Path
62
63         Write-Log "Usuario creado: $($User.SamAccountName)"
64     } else {
65         Write-Log "Usuario ya existe: $($User.SamAccountName)"
66     }
67
68     if ($User.Group) {
69         Add-ADGroupMember -Identity $User.Group -Members $User.SamAccountName -ErrorAction SilentlyContinue
70         Write-Log "Usuario $($User.SamAccountName) añadido al grupo $($User.Group)"
71     }
72 }

```

Información en los Logs

El script primero registra el inicio del proceso y carga los datos del archivo externo mediante `$Data = Import-Csv $CsvPath`. A continuación, procesa la información de forma jerárquica y ordenada mediante filtros: primero crea las Unidades Organizativas (carpetas), luego los Grupos y finalmente los Usuarios. Para cada elemento, el script utiliza bucles `ForEach-Object` que llaman a las funciones de creación definidas previamente y termina registrando la finalización del despliegue en el archivo de log.

```

78 Write-Log "==== Inicio del despliegue AD ====="
79
80 $Data = Import-Csv $CsvPath
81
82 # OUs
83 $Data | Where-Object { $_.Type -eq "OU" } | ForEach-Object {
84     Create-OU -Name $_.Name -Path $_.Path
85 }
86
87 # Grupos
88 $Data | Where-Object { $_.Type -eq "GROUP" } | ForEach-Object {
89     Create-Group -Name $_.Name -Path $_.Path -Scope $_.Scope
90 }
91
92 # Usuarios
93 $Data | Where-Object { $_.Type -eq "USER" } | ForEach-Object {
94     Create-User -User $_
95 }
96
97 Write-Log "==== Fin del despliegue AD ====="

```