

Configuración de usuarios mediante .txt SCRIPT

miércoles, 4 de octubre de 2023 10:57

SCRIPT COMPLETO, SIN CONFIGURACIÓN DE PERMISOS DE JEFE EN CADA DPTO

```
#Creacion y configuracion de grupos de trabajo.

# Especifica la ruta de la OU donde quieres crear los grupos
$OU = "OU=Usuarios,DC=cr,DC=loc"

# Ruta completa al archivo de texto que contiene los nombres de los grupos
$archivoGrupos = "C:\Users\Administrator\Documents\grupos.txt"

# Leer el archivo de texto
$lineas = Get-Content -Path $archivoGrupos

# Crear grupos en la OU especificada
foreach ($linea in $lineas) {
    # Verificar si el grupo ya existe en la OU
    if (-not (Get-ADGroup -Filter {Name -eq $linea} -SearchBase $OU -ErrorAction SilentlyContinue)) {
        # Crear el grupo en la OU
        New-ADGroup -Name $linea -Path $OU -GroupScope Global -GroupCategory Security
        Write-Host " "
        Write-Host "Grupo $linea creado en la OU $OU."
        # Crear carpeta para los grupos
        New-Item "E:\Export\$linea" -ItemType Directory
        # Ruta de la carpeta donde vamos añadir al grupo de trabajo
        $rutaTrabajo = "E:\Export\$linea"
        # Dar permisos a los grupos de trabajo dentro de Dpto1 y Dpto2
        icacls.exe $rutaTrabajo /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "${linea}:(OI)(CI)RX"
        # Crear carpeta comun de cada departamento
        New-Item "E:\Export\$linea\Comun" -ItemType Directory
        icacls.exe $rutaTrabajo\Comun /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "${linea}:(OI)(CI)F"
        Write-Host " "
        Write-Host " "
        Write-Host " "
    }
}
#-----

#Creacion usuarios mediante fichero

#Ruta del archivo de texto que contiene los datos de usuario
$archivoUsuarios = "C:\Users\Administrator\Documents\usuarios.txt"

#Verificar si el archivo existe
if (Test-Path $archivoUsuarios) {
    #Leer el contenido del archivo
    $lineasusuarios = Get-Content $archivoUsuarios

    foreach ($lineausuario in $lineasusuarios) {
        #Separamos campos por espacios
        $datosUsuario = $lineausuario -split " "
        Write-Host " "
        if ($datosUsuario.Count -eq 6) {
            $nombreUsuario = $datosUsuario[0].Trim() #Para espacios en blanco
            $apellido1 = $datosUsuario[1].Trim()
            $apellido2 = $datosUsuario[2].Trim()
            $inicialesUsuario = $datosUsuario[3].Trim()
            $contrasena = $datosUsuario[4].Trim()
            $grupos = $datosUsuario[5].Trim()

            Write-Host " "
            #Comprobar si el usuario ya existe
            if (-not (Test-Path "C:\Users\$nombreUsuario")) {
                #Crear el usuario
                Write-Host " "
                New-ADUser -Name $nombreUsuario -GivenName $nombreUsuario -Surname "$apellido1 $apellido2" -Initials $inicialesUsuario -AccountPassword (ConvertTo-SecureString $contrasena -AsPlainText -Force) -PasswordNeverExpires $true -Enabled $true -Path $OU

                Write-Host "Usuario $nombreUsuario $apellido1 $apellido2 se ha creado correctamente."

                #Añadir usuarios a grupos de trabajo
                Add-ADGroupMember -Identity "$grupos" -Members "$nombreUsuario"

                #Crear carpetas de usuario

                #Ruta del directorio en el que deseas crear las carpetas
                $rutaCarpeta = "E:\Export\$grupos"

                #Crear las carpetas en el directorio de destino
                New-Item $rutaCarpeta\$nombreUsuario -ItemType Directory
                Write-Host "Carpeta de usuario creada correctamente dentro de su departamento."

                #Añadimos permisos a los usuarios en sus carpetas
                $rutaPermisos = "E:\Export\$grupos\$nombreUsuario"
                icacls.exe $rutaPermisos /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "${nombreUsuario}:(OI)(CI)F"
                Write-Host " "
            }
        }
    }
}
else {
    Write-Host "El archivo $archivoUsuarios no se encontró."
}
```

INTENTO DE CONFIGURACIÓN DE PERMISOS DE JEFE DE CADA DPTO (NO FUNCIONA)

```
#Permisos usuario jefe dptos
$jefes = $archivoUsuarios | where-Object
( $datosUsuario[6].Trim() -eq "6" )

foreach ($jefe in $jefes){

    $nombreUsuario=$jefe.
    $datosUsuario[0].Trim()
    $grupos=$jefe.$datosUsuario[5].Trim()

    $ruta="E:\Export"

    #Obtener carpetas de los usuarios
    desde la carpeta del dpto
    $carpetasUsus = Get-ChildItem -Path
    $ruta\$grupos -Directory -Recurse

    #Asignar permisos
    foreach ($carpetasUsu in
    $carpetasUsus){

        if ($carpetasUsu -ne
        $nombreUsuario){
            $rutapermisos = "E:\Export\$grupos
            \$nombreUsuario"
            icacls.exe $rutapermisos
            /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)
            (CI)F", "${nombreUsuario}:(OI)(CI)F"
        }
    }
}
```

CREACIÓN Y CONFIGURACIÓN GRUPOS DE TRABAJO

PASO 1

```
1 #Creacion y configuracion de grupos de trabajo.
2
3 # Especifica la ruta de la OU donde quieres crear los grupos
4 $OU = "OU=Usuarios,DC=cr,DC=loc"
5
6 # Ruta completa al archivo de texto que contiene los nombres de los grupos
7 $archivoGrupos = "C:\Users\Administrator\Documents\grupos.txt"
8
9 # Leer el archivo de texto
10 $lineas = Get-Content -Path $archivoGrupos
```

En primer lugar, hemos especificado la ruta de la OU (Unidad Organizativa) donde se crearán dichos grupos de trabajo, seguido de la ruta de nuestro fichero .txt que tiene almacenado los departamentos que queremos agregar.

Creamos una variable llamada \$lineas que recorrerá el archivo de texto, esto lo indicamos haciendo referencia a la variable que contiene el .txt (\$archivoGrupos).

PASO 2

```
# Crear grupos en la OU especificada
foreach ($linea in $lineas) {
    # Verificar si el grupo ya existe en la OU
    if (-not (Get-ADGroup -Filter {Name -eq $linea} -SearchBase $OU -ErrorAction SilentlyContinue)) {
        # Crear el grupo en la OU
        New-ADGroup -Name $linea -Path $OU -GroupScope Global -GroupCategory Security
        Write-Host " "
        Write-Host "Grupo $linea creado en la OU $OU."
    }
}
```

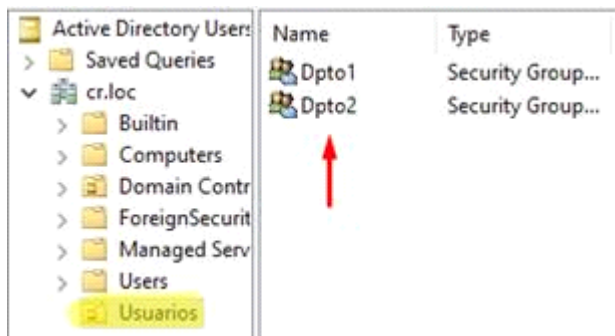
A continuación, hacemos un bucle, en el que su primer paso es verificar si el grupo ya existe en la OU a través de un if, en el que la variable \$linea se encarga de buscar guardar los grupos en la variable \$lineas que hace referencia a nuestro fichero con los grupos.

Una vez el sistema comprueba que el grupo no existe, crea dicho grupo a través del comando New-ADGroup, especificando con el comando -Path \$OU (variable declarada en la anterior foto) donde se va a crear...Como comprobación hacemos un Write-Host.

Comprobación:

```
Grupo Dpto1 creado en la OU OU=Usuarios,DC=cr,DC=loc.
Grupo Dpto2 creado en la OU OU=Usuarios,DC=cr,DC=loc.

PS C:\Users\Administrator>
```



PASO 3

```
# Crear carpeta para los grupos
New-Item "E:\Export\$linea" -ItemType Directory
# Ruta de la carpeta donde vamos a añadir al grupo de trabajo
$rutaTrabajo = "E:\Export\$linea"
# Dar permisos a los grupos de trabajo dentro de Dpto1 y Dpto2
icacls.exe $rutaTrabajo /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "${linea}:(OI)(CI)RX"
```

Teniendo ya creadas las OU, vamos a generar sus carpetas correspondientes a través del comando New-Item especificando la ruta donde se van a alojar, además le indicamos que es de tipo directorio (-ItemType Directory).

Ahora queremos indicar al script en que ruta vamos a añadir los grupos de trabajo a sus carpetas, creamos una variable (\$rutaTrabajo) en la que dentro se le indica la ruta correspondiente, utilizando la variable \$linea la cual contiene los nombres de los grupos.

Por último, con el ejecutable icacls.exe vamos a asignar los permisos a los grupos de trabajo dentro de su carpeta.

-/inheritance:r: Deshabilitar las herencias del grupo.

-/grant: Comando para especificar que vamos a asignar permisos.

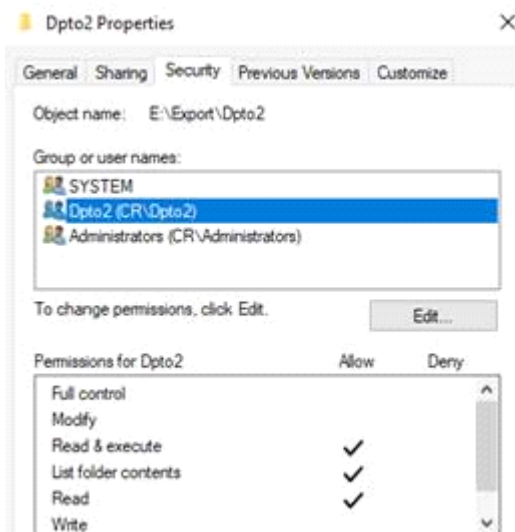
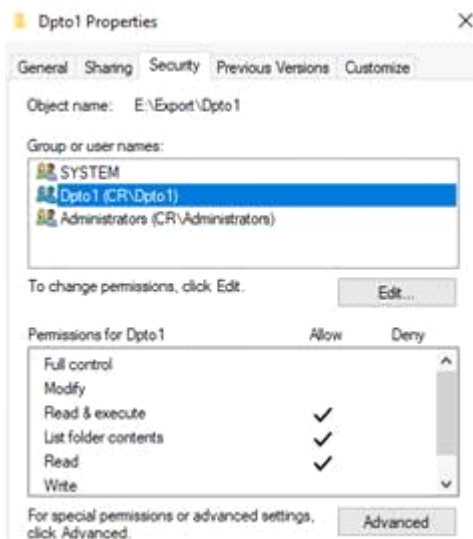
-"Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "\${linea}:(OI)(CI)RX": Al utilizar el ejecutable icacls, nos elimina los grupos que ya están por defecto (Administrators y System), así que nosotros los volvemos a añadir asignando permisos tipo FULL CONTROL(F).

En el caso de la variable \$linea, asignamos permisos de lectura y ejecución, ya que no tiene sentido que tenga permisos completos dentro de su departamento.

-(OI)(CI): Comando para asignar los permisos a su carpeta y subcarpetas.

```
Mode                LastWriteTime         Length Name
----                -
d-----          27/09/2023   9:18           Dpto1
processed file: E:\Export\Dpto1
Successfully processed 1 files; Failed processing 0 files
```

```
d----- 27/09/2023 9:18 Dpto2
processed file: E:\Export\Dpto2
Successfully processed 1 files; Failed processing 0 files
```

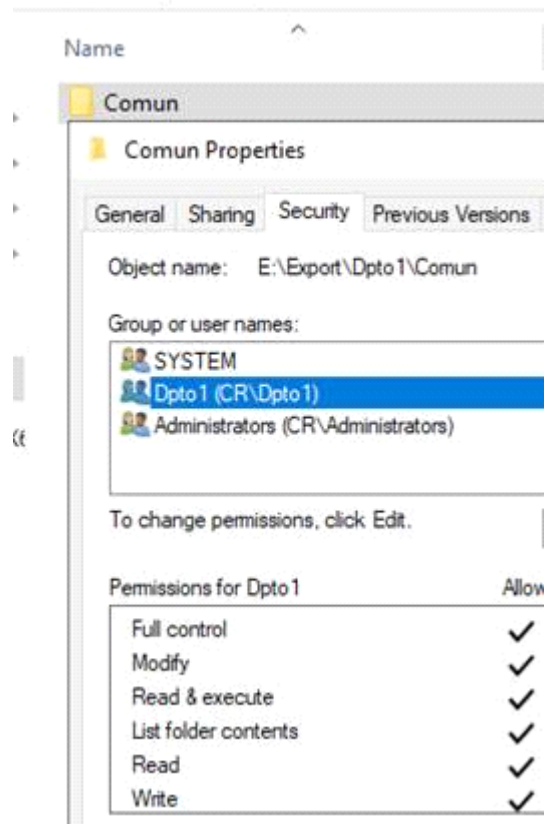


ÚLTIMO PASO

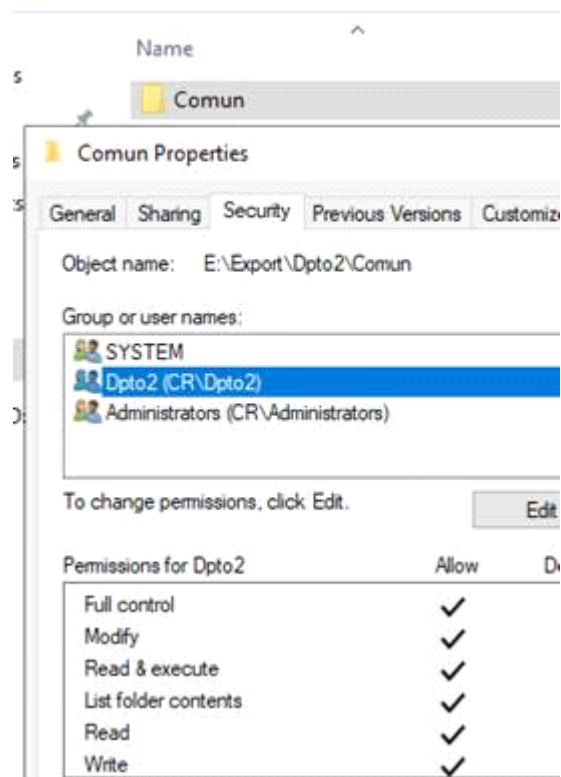
Ahora creamos una carpeta llamada común dentro cada departamento que, en este caso, si tienen que tener permisos tipo FULL CONTROL los grupos de trabajo (Dpto1 y Dpto2), ya que deben tener acceso total a esa carpeta.

```
#Crear carpeta comun de cada departamento
New-Item "E:\Export\${linea}\Comun" -ItemType Directory
icacls.exe $rutaTrabajo\Comun /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "${linea}:(OI)(CI)F"
Write-Host " "
Write-Host " "
Write-Host " "
```

Datos (E:) > Export > Dpto1



Datos (E:) > Export > Dpto2



Configuración de grupos de trabajo y su carpeta común hecha.

CREACIÓN Y CONFIGURACIÓN DE USUARIOS

PASO 1

Creamos la variable \$archivoUsuarios para indicar la ruta que contiene los datos de los usuarios.

A continuación generamos un if para verificar si el archivo usuarios.txt, en caso de que no exista, leerá el contenido del archivo mediante la variable \$lineausuarios, que le indicamos donde esta el archivo (en la variable \$archivoUsuarios).

```
#Ruta del archivo de texto que contiene los datos de usuario
$archivoUsuarios = "C:\Users\Administrator\Documents\usuarios.txt"

#Verificar si el archivo existe
if (Test-Path $archivoUsuarios) {

    #Leer el contenido del archivo
    $lineasusuarios = Get-Content $archivoUsuarios
```

PASO 2

Dentro del if, generamos un foreach, con el objetivo de que la variable \$lineausuario guarde los datos de la variable \$lineausuarios.

Creamos otro if dentro del foreach para que \$datosUsuario, guarde cada posición de nuestro fichero, empezando por el 0 que sería la primera posición (Nombre) y acabando por el 6 que sería la séptima posición (jefes).

.Trim(): Para que no guarde espacios en blanco.

```
foreach ($lineausuario in $lineasusuarios) {
    #Separamos campos por espacios
    $datosUsuario = $lineausuario -split " "
    Write-Host " "
    if ($datosUsuario.Count -eq 6) {
        $nombreUsuario = $datosUsuario[0].Trim() #Para espacios en blanco
        $apellido1 = $datosUsuario[1].Trim()
        $apellido2 = $datosUsuario[2].Trim()
        $inicialesUsuario = $datosUsuario[3].Trim()
        $contrasena = $datosUsuario[4].Trim()
        $grupos = $datosUsuario[5].Trim()
        $jefes = $datosUsuario[6].Trim()
```

PASO 3

Una vez la variable \$datosUsuario tiene guardado cada datos de cada columna con una variable para cada campo, comprobamos si el usuario que se va a crear ya existe, mediante un if, insertando la variable \$nombreUsuario, variable que contiene el nombre de usuario.

Si no existe, creará el usuario con el comando New-AdUser (crear usuarios en Active Directory).

-GivenName: Para indicar el nombre del usuario.

-Surname: Indica los apellidos.

-Initials: Iniciales con las que aparecerá el usuario en sus datos de cuenta.

-AccountPassword: Indicarla contraseña

-ConvertTo-SecureString: Convertir en contraseña segura.

-AsPlainText -Force: Para que no esté en texto plano.

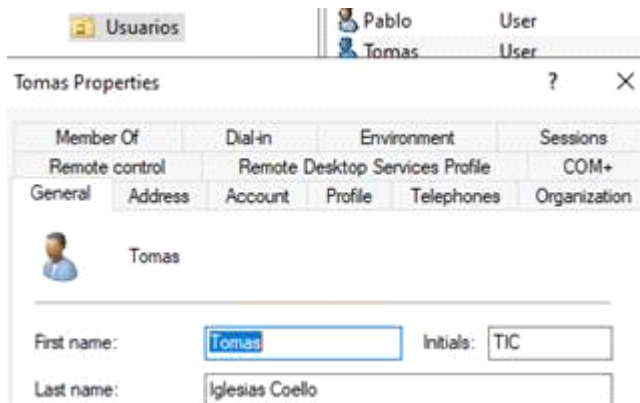
-PasswordNeverExpires \$true: Con la variable \$true indicamos que la contraseña nunca expire.

-Enabled \$true: Activación del usuario.

-Path \$OU: En que unidad organizativa se alojará el usuario.

```
#Comprobar si el usuario ya existe
if (-not (Test-Path "C:\Users\$nombreUsuario")) {
    #Crear el usuario
    Write-Host " "
    New-ADUser -Name $nombreUsuario -GivenName $nombreUsuario -Surname "$apellido1 $apellido2"
    -Initials $inicialesUsuario -AccountPassword (ConvertTo-SecureString $contrasena -AsPlainText -Force)
    -PasswordNeverExpires $true -Enabled $true -Path $OU

    Write-Host "Usuario $nombreUsuario $apellido1 $apellido2 se ha creado correctamente."
}
```



Account options:

☐ User must change password at next logon
☐ User cannot change password
☒ Password never expires
☐ Store password using reversible encryption

Usuario Tomas Iglesias Coello se ha creado correctamente.

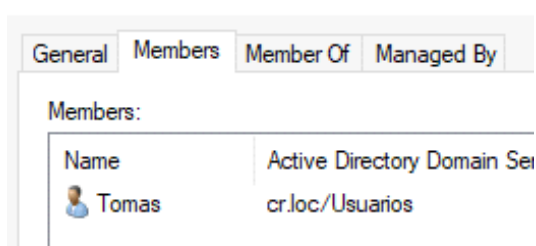
PASO 4

Cuando ya tenemos creado el usuario, lo añadimos a su departamento correspondiente (grupo de trabajo), mediante el comando ADD-ADGroupMember, indicando con la variable \$grupos ya definida, que se alojará en su grupo correspondiente

-Members "\$nombreUsuario": Miembros de usuario que se alojarán en el grupo (la variable \$nombreUsuario indica el nombre del usuario para poder identificarlo.)

```
#Añadir usuarios a grupos de trabajo
Add-ADGroupMember -Identity "$grupos" -Members "$nombreUsuario"
```

Dpto2 Properties



PASO 5

Especificamos la ruta donde se crearán las carpetas de usuario.

Dentro del disco E, en la carpeta Export y con la variable grupos indicamos que se crearán dentro de cada grupo correspondiente, dependiendo del usuario.


```
#Crear carpetas de usuario

#Ruta del directorio en el que deseas crear las carpetas
$rutaCarpeta = "E:\Export\$grupos"
```

PASO 6

A continuación, creamos la carpeta de cada usuario en su grupo de trabajo correspondiente.

Con el comando New-Item especificamos la ruta donde se va a crear la carpeta del usuario.

\$rutaCarpeta\\$nombreUsuario: En la variable rutaCarpeta indicada en el anterior paso, se generará una carpeta con los datos de la variable nombreusuario.

```
#Crear las carpetas en el directorio de destino
New-Item $rutaCarpeta\$nombreUsuario -itemType Directory
Write-Host "Carpeta de usuario creada correctamente dentro de su departamento."
```

```
Carpeta de usuario creada correctamente dentro de su departamento.
processed file: E:\Export\Dpto2\Tomas
```



ÚLTIMO PASO

Por último, asignamos permisos a cada usuario sobre su carpeta correspondiente.

Creamos una variable llamada \$rutapermisos, donde a través de la variable \$grupos y \$nombreUsuario dentro de la ruta, indicamos donde se aplicarán los permisos.

Como estamos dando permisos a los usuarios sobre su carpeta correspondiente, indicamos que tengan permisos de tipo FULL CONTROL.




Para indicar que el usuario tiene permisos sobre su carpeta hemos utilizado la ya declarada variable \$nombreUsuario, que aloja el nombre del usuario.

```
#Añadimos permisos a los usuarios en sus carpetas
$rutapermisos = "E:\Export\$grupos\$nombreUsuario"
icacls.exe $rutapermisos /inheritance:r /grant "Administrators:(OI)(CI)F", "SYSTEM:(OI)(CI)F", "${nombreUsuario}:(OI)(CI)F"
Write-Host " "
1
```

```
Successfully processed 1 files; Failed processing 0 files
```


Object name: E:\Export\Dpto2\Tomas

Group or user names:

 SYSTEM
 Tomas
 Administrators (CR\Administrators)

To change permissions, click Edit.

Edit...

Permissions for Tomas	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

**Configuración de usuarios hecha.
Script terminado.**