Event Manager (M159)

Tim Bühler
ST13a

Inhaltsverzeichnis

Einleitung

Die Aufgabe war es, Ein Powershell Script für die Verwaltung von Usern an einem Event zu erstellen. Dieses habe ich Event Manager gennant, da es so konzipiert wurde, dass es universell, also nicht nur für dieses eine Event, einsetzbar ist. Das Tool ist also nicht auf die Messe Basel spezifiziert, sondern kann nach minimalen Anpassungen auf anderen Systemen (Andere Domäne) eingesetzt werden.

Dokumentation Powershell Script

Das Script verfügt über Inline Comments. Sie Beschreiben das wichtigste knapp. Hier wird es etwas ausführlicher dokumentiert.

Meta, Definitionen

<# Meta Info Name: Event Manager Version: 0.1 Date: 2016.04.10 By: Tim Buehler Module : 159 School/Class: TBZ/St13a #>	Meta Informationen. Darin werden Name, Version und Datum sowie Urheber definiert.
clear-host	Damit die Konsole vor dem Ausführen gesäubert wird.
Import-Module ActiveDirectory	Für das Administrieren einer AD wird ein Powershell AD Modul importiert. Mit diesem Modul können Active Directory Pfade wie normale Verzeichnisse angesprochen warden.
#Definitions $Domain = 'schoollab' $TLD = 'local'	Möchte man das Script in einer anderen Domäne verwenden, muss man jediglich $Domain und $TLD anders setzen.

Menü

Set-Location $ADRootDirectory Create-OU -ou Event-Users Set-Location OU=Event-Users	Nun beginnt das eigentliche Script. Wir erstellen die OU Event-Users und wechseln in dieses Verzeichnis. In dieser OU werden dann die OUs für die Events angelegt.
Write-host 'Welcome to the Event Manager' Write-Host '1 = Get Events' Write-Host '2 = New Event' Write-Host '3 = Delete or Backup Event' Write-Host 'x = Exit' $action = Read-Host -Prompt 'Please Enter a Number'	Das Menü fragt den User nach der gewünschten Aktion. Über dieses und das folgende if statement ist eine DO UNTIL Schlafe gemacht, damit sich das Menü nach einer getätigten Aktion wiederholt. Bis „x“ eingegeben wird.

Events anzeigen

Set-Location $AdRootDirectory Set-Location 'OU=Event-Users' Get-ChildItem | Write-Host	Bei einer „1” warden die Unterverzeichnisse der OU Event-Units angezeigt. Also die aktuellen Events, welche als OU da definiert sind.

Events erstellen

`$EventName = Read-Host -Prompt 'Enter event name'`	Bei einer „2“ Wird ein neues Event erstellt. Erst werden Infos zum Event vom Benutzer verlangt. (Nur ein Bsp. in der Linken Box)
`Create-OU -ou $EventName` `Set-Location OU=$EventName`	Danach wird Das Event (OU) erstellt.
`[array]$users = Get-Content $InputFile`	Das Userfile in eine Variable abgespeichert
`foreach ($item in $users)`	Folgende Änderungen werden nun für jeden user (welcher vom Userfile importiert wurde) vorgenommen.
`$TempPassword = GET-Temppassword -length 8 -sourcedata $ascii`	Es wird ein Passwort generiert mit 8 Zeichen und dem ASCII Zeichensatz.
`$SecureTempPassword = ConvertTo-SecureString $TempPassword -AsPlainText -Force`	Das Passwort wird als SecureString in einer anderen Variable abgespeichert.
`New-ADUser ` `` ` `-Name $item ` `` ` `-AccountPassword $currentPassword ` `` ` `-CannotChangePassword $true ` `` ` `-PasswordNeverExpires $true ` `` ` `-AccountExpirationDate $EventEnd ` `` ` `-ChangePasswordAtLogon $false ` `` ` `-Enabled $true ` `` ` `-KerberosEncryptionType AES256`	Danach wird ein Benutzer in der AD angelegt. Die Linksstehenden Parameter werden verwendet.
`[array]$UserLoginData = $item,$temppassword` `$UserLoginData | Out-file $OutputFiles\$item.txt`	Danach wird für den User ein Textfile mit seinem Usernamen und Passwort generiert.

Events löschen und Backupen

`$DeleteEvent = Read-Host -Prompt 'Enter the OU you want to Delete'`
`$OutputFile = Read-Host -Prompt 'Where do you want to Save you backup?'`
 	
`$OutputFile = $OutputFiles + '\' + $deleteEvent + '.txt'`
	Bei einer „3“ wird eine OU gelöscht und die Namen der OU und User gespeichert. Dafür werden zuerst wieder ein paar Variablen des Benutzers gebraucht.

`$users = get-aduser $DeleteEvent`
 	
`$users | Out-File $OutputFile`
	Die User auslesen und als Output speichern.

`Remove-ADOrganizationalUnit $DeleteEvent -Recursive`	Die OU mitsamt Unterobjekten löschen

Error

`Write-Host 'Ungültige Eingabe'`	Wird keine Option angegeben, wird Ungültige Eingabe in die Konsole geschrieben und durch die DO UNTIL Funktion wieder das Menü aufgerufen.

Anhang

250 Benutzer generieren.

Zum testen des Scripts brauchte ich 250 Users in einem File. Dieses Habe ich mit folgendem Script generiert.

```
#Creates 250 named user in ascii encoded text file

#definitions
$path = 'C:\Temp'
$file = '250users.txt'
$fullpath = $path + '\' + $file
[array]$users = $null
$x = 0 #reset var X

#counting and adding users to variable users
do
{
     $currentuser = "user$x"
     $users += $currentuser

     $x += 1
}
until ($x -eq 250)

#writes users to file
$users | out-file $fullpath -Encoding ascii
```