

7. Powershell

a. Wat is Powershell?

PowerShell is een krachtige tool voor het automatiseren van taken in Windows, waaronder het beheer van Active Directory. Met PowerShell kunnen we snel en efficiënt gebruikersaccounts aanmaken, beheren en verwijderen in Active Directory. PowerShell-cmdlets maken gebruik van het Active Directory Service Interface (ADSI) om taken uit te voeren in Active Directory.

b. Powershell commando's voor aanmaken van mappen en bestanden:

- Maak een nieuwe map:

```
New-Item -ItemType Directory -Path "C:\NieuweMap"
```

- Maak een nieuwe bestand:

```
New-Item -ItemType File -Path "C:\NieuweMap\NieuwBestand.txt"
```

- Kopieer een bestand naar een andere locatie:

```
Copy-Item -Path "C:\Bestand.txt" -Destination "D:\Backup"
```

- Hernoem een bestand:

```
Rename-Item -Path "C:\OudBestand.txt" -NewName "NieuwBestand.txt"
```

- Verplaats een bestand naar een andere map:

```
Move-Item -Path "C:\OudBestand.txt" -Destination "C:\NieuweMap"
```

- Maak een nieuwe map en een bestand in die map:

```
New-Item -ItemType Directory -Path "C:\NieuweMap"
```

```
New-Item -ItemType File -Path "C:\NieuweMap\NieuwBestand.txt"
```

- Maak een nieuwe map en verplaats een bestand naar die map:

```
New-Item -ItemType Directory -Path "C:\NieuweMap"  
Move-Item -Path "C:\OudBestand.txt" -Destination "C:\NieuweMap"
```

- Voeg inhoud toe aan een bestand:

```
Add-Content -Path "C:\Bestand.txt" -Value "Dit is wat extra inhoud."
```

- Verwijder een bestand:

```
Remove-Item -Path "C:\OudBestand.txt"
```

- Verwijder een map en alle inhoud ervan:

```
Remove-Item -Path "C:\OudeMap" -Recurse
```

Opdracht:

Maak een map aan genaamd PXL, maak er een bestand in aan genaamd pxl.txt. Voeg je naam en voornaam toe aan het tekstbestand. Maak een kopie van dit bestand naar een nieuwe map PXL2 en verwijder de oude map PXL inclusief het oude bestand pxl.txt

c. Lokale users aanmaken in powershell

- Maak een lokale user aan met een versleuteld wachtwoord

```
$Userpassword = Read-Host -AsSecureString  
New-LocalUser "PXLuser" -Password $Userpassword -FullName "PXLUser" -Description  
"deze user is aangemaakt via Powershell"
```

- Aanpassen settings voor wachtwoord

```
Set-Localuser -name PXLUSER -Passwordneverexpires $True -password $Userpassword
```

- Alle lokale users tonen

```
Get-LocalUser
```

- Alle lokale users verwijderen

```
Remove-Localuser -name "PXLuser"
```

- nieuwe lokale usergroup maken

new-localgroup -name 'studenten'

- Gebruiker toevoegen aan lokale groep

add-localgroupmember -group 'studenten' -member 'pxlstudent'

- Gebruiker uit groep gooien

remove-localgroupmember -group 'studenten' -member 'pxlstudent'

- Groep verwijderen

Remove-localgroup -name 'studenten'

Opdracht:

Maak een nieuwe user PXLuser met wachtwoord PXLRocks

Maak een nieuwe user PXLuser met wachtwoord PXLRocks en plaats deze in de lokale administratorgroep

verander het wachtwoord van PXLuser naar PXLWins

d. Active Directory-cmdlets in PowerShell

Lezen users in AD

- Hoe alle users tonen in AD:

Get-ADUser -Filter *

- Geef me een bepaalde user met Voornaam

Get-ADUser -Filter "GivenName -eq 'Ethan'"

- Geef me een bepaalde user die start met 'am'

Get-Aduser -Filter "Name -like '*am*'"

- Geef me enkel de users die enabled zijn
Get-ADUser -Filter "Enabled -eq 'True'"

- Geef me enkel de users uit een bepaalde OU
Get-ADUser -Filter * -SearchBase "OU=Users,OU=PXLDigital,DC=pxldigital,DC=local" ==>
focus op volgorde van OU

Opdracht:

Toon alle users in de OU IT.

Lezen groepen in AD

- Opvragen van alle groepen opvragen in Active Directory
Get-ADGroup - Filter *
of focus op Name Focus op GroupScope

- Opvragen van groep die start met de naam IT
Get-ADGroup -Filter "Name like 'it*'"

- Opvragen van domain local groepen in Active Directory
Get-ADGroup -Filter "GroupScope -eq 'domainlocal'"

- Opvragen van groepen die in de OU Groups staat
Get-ADGroup -Filter * -SearchBase "OU=Groups, OU=PXLDigital,DC=pxldigital,DC=local"

- Verkennen Get-ADGroupMember
Get-ADGroupMember Dan zie je dat het commando een Identity verwacht. Identity is een naam van een groep

- Opvragen van alle Groepen
Get-ADGroupMember -Identity IT_DL

Opdracht:

geef alle groepen in de OU 'groups'

Aanmaken users in AD

- Maak een nieuwe gebruiker aan met de standaardinstellingen:

```
New-ADUser -Name "Gebruikersnaam" -GivenName "Voornaam" -Surname "Achternaam"
-SamAccountName "Gebruikersnaam" -AccountPassword (ConvertTo-SecureString
"Wachtwoord" -AsPlainText -Force) -Enabled $true
```

- Maak een nieuwe gebruiker aan met extra eigenschappen:

```
New-ADUser -Name "Gebruikersnaam" -GivenName "Voornaam" -Surname "Achternaam"
-SamAccountName "Gebruikersnaam" -AccountPassword (ConvertTo-SecureString
"Wachtwoord" -AsPlainText -Force) -Enabled $true -Department "Afdeling" -Office "Kantoor"
-Title "Functie"
```

- Voeg een gebruiker toe aan een specifieke OU:

```
New-ADUser -Name "Gebruikersnaam" -GivenName "Voornaam" -Surname "Achternaam"
-SamAccountName "Gebruikersnaam" -AccountPassword (ConvertTo-SecureString
"Wachtwoord" -AsPlainText -Force) -Enabled $true -Path
"OU=Gebruikers,DC=Contoso,DC=com"
```

- Voeg een gebruiker toe aan meerdere groepen:

```
New-ADUser -Name "Gebruikersnaam" -GivenName "Voornaam" -Surname "Achternaam"
-SamAccountName "Gebruikersnaam" -AccountPassword (ConvertTo-SecureString
"Wachtwoord" -AsPlainText -Force) -Enabled $true | Add-ADGroupMember -Identity
"Groep1","Groep2"
```

- Maak meerdere gebruikers tegelijk aan vanuit een CSV-bestand:

```
Import-Csv C:\Users.csv | ForEach-Object {New-ADUser -Name $_.Name -GivenName
$_.FirstName -Surname $_.LastName -SamAccountName $_.Username -AccountPassword
(ConvertTo-SecureString $_.Password -AsPlainText -Force) -Enabled $true}
```

Opdracht:

Maak een user IT_5 aan binnen je OU IT met voornaam IT_5 en familienaam IT_5, gebruik wachtwoord 'Pass1234', voeg deze toe aan de GL-groep IT_GL en enable de account.

Nieuwe AD groepen aanmaken

Aanmaken group

```
New-ADGroup -Name "Powershell" -GroupScope Global -Path
"OU=Groups,OU=PXLDigital,DC=pxldigital,DC=local"
```

Aanmaken OU via commandline Get-help *unit*

```
New-ADOrganizationalUnit -Name "Import" -Path "OU=PXLDigital,dc=pxldigital,dC=local"
```

Jan toevoegen aan de groep Powershell

```
Add-ADGroupMember -Identity Powershell -Members jan.peeters
```

Opvragen van gebruikers uit een groep

```
Get-ADGroupMember -Identity Powershell
```

Gebruiker verwijderen uit een groep

```
Remove-ADGroupMember -Identity Powershell - Members jan.peeters
```

Gebruiker verwijderen

```
Get-ADUser -Filter "Name -like '*jan*'" Get-ADUser -Filter "Name -like '*jan*'" |
```

```
Remove-ADUser
```

Groep verwijderen

```
Get-ADGroup -Filter "Name -like '*import*'" Get-ADGroup -Filter "Name -like '*import*'" |
```

```
Remove-ADGroup
```

Toepassingen

Maak een excelbestand aan met 5 kolommen

CSV bestand aanmaken ○ Voornaam ○ Achternaam ○ Gebruikersnaam ○ Wachtwoord ○ Afdeling

Plaats hierin 3 fictieve users

Sla het bestand op als users.csv

Verplaats het csv naar de C schijf van de server

Schrijf in Powershell ISE het volgende script stapsgewijs op.

```
$users = Import-Csv -Delimiter ";" -path C:\users.csv
ForEach ($user in $users) {
    $firstname = $user.voornaam
    $lastname = $user.achternaam
    $fullname = $user.voornaam + " " + $user.achternaam
    $samaccount = $account
    $ou = $user.afdeling
    $setpassword = ConvertTo-SecureString $user.wachtwoord -AsPlainText -Force
    New-ADUser -Name $fullname -Displayname $fullname -GivenName $firstname -Surname $lastname -SamAccountName $samaccount -Path "OU=$ou,OU=Users,OU=pxldigital,DC=pxldigital,DC=local" -AccountPassword $password -Enabled $true -PassThru
}
```