Module 8: Interaction avec Active Directory

PowerShell peut interagir avec Active Directory de façon assez simple - le fonctionnement est très similaire à VBScript.

La première chose à faire est de définir une variable qui pointera vers une unité organisationnelle (UO) de notre Active Directory (AD). C'est par cette variable qu'on accèdera aux objets qu'elle contient. On pourra alors en créer des nouveaux, en détruire ou en modifier à notre guise:

```
$uo=[ADSI]"LDAP://localhost:389/CN=Users,DC=AECreseaux,DC=com"
```

- Le ADSI entre crochets (ça signifie Active Directory Service Interface et ça permet donc d'accéder à des objets d'AD).
- Le chemin donné entre guillemets est un chemin LDAP (*Lightweight Directory Access Protocol* - protocole léger d'accès à l'annuaire). C'est le protocole qu'utilise AD (mais c'est un protocole défini hors-AD, d'autres annuaires peuvent s'en servir).
- Ce qui suit le LDAP://, c'est le nom du serveur suivi du port 389 (standard pour LDAP). Cette partie est facultative mais peut parfois régler des problèmes.
- Juste après, c'est le *distinguished name* (DN - nom distingué). Cela permet d'identifier un objet d'annuaire de façon unique et précise. Un DN est toujours composé de plusieurs parties séparées par des virgules. Chaque partie est elle-même composée d'une abréviation de deux lettres, d'un symbole d'égalité et d'une valeur. On organise les parties en ordre croissant, comme pour adresser une lettre.
- Pour une UO, on devrait normalement utiliser OU, qui signifie *organizational unit* (unité organisationnelle) et qui contient le nom de l'UO voulue. Si elle est imbriquée dans d'autres UO, on devra les nommer toutes, partant de la plus interne. Attention! Les conteneurs déjà créés par AD comme Computers et Users ne sont pas des UO. On y accède avec CN (*common name, voir plus bas).*
- DC signifie *domain controller* (contrôleur de domaine) et contient finalement l'adresse DNS du domaine. Dans notre cas, on suppose que le domaine s'appelle AECreseaux.com. Il faut un DC pour chaque partie de l'adresse, données dans l'ordre.
- Si on nommait un usager, on aurait commencé par CN pour *common name* (nom commun). Le nom commun est le nom de l'objet usager.

Pour créer un usager, on doit partir de l'UO qui le contiendra. On créera d'abord une variable qui représentera l'usager:

```
$usager = $uo.Create("user","cn=NouvelUsager")
```

La méthode Create permet de créer quelque chose dans l'UO. On définit le type comme premier paramètre (ici, on a utilisé "user", mais on aurait pu utiliser "organizationalUnit" ou "group"). Le deuxième paramètre contiendra le nom de l'usager. Ici, le DN est plus court puisqu'on sait qu'il sera créé dans l'UO - uniquement besoin de son propre nom!

Une fois l'objet usager créé (notez qu'il n'est créé que localement, pas dans AD encore), on peut modifier ses attributs avec la méthode put:

```
$usager.put("sAMAccountName", "nouvelu")
$usager.put("givenName", "Nouvel")
$usager.put("sn", "Usager")
$usager.put("telephoneNumber", "5145551212")
$usager.put("profilePath", "\\serveur\profils\nouvelu")
```

Notez que:

- sAMAccountName est le nom de login de l'usager et c'est le seul paramètre indispensable pour un usager.
- givenName est le prénom
- sn est le nom de famille

Une fois que notre objet usager est créé, on doit l'envoyer dans AD en faisant simplement:

```
$usager.SetInfo()
```

Pour que ça fonctionne, il faut exécuter PowerShell en tant qu'un administrateur du domaine (et être logué sur le domaine, bien entendu).

Par défaut, les comptes usagers sont créés désactivés et sans mot de passe. Pour rémédier à la situation, on doit faire (après le SetInfo qui a créé l'objet proprement dit dans AD):

```
$usager.psbase.Invoke("SetPassword","1212patate")
$usager.psbase.InvokeSet('Accountdisabled',$false)
$usager.psbase.CommitChanges()
```

- Psbase vous permet d'accéder à la technologie se trouvant dans AD, qui n'a pas été conçue au départ pour PowerShell.
- Invoke permet d'appeler une méthode de la technologie AD, ici SetPassword. Notez les guillemets.
- InvokeSet permet de modifier une propriété de la technologie AD, ici Accountdisabled. Notez les apostrophes.
- CommitChanges permet de sauvegarder vos changements.

On peut également modifier un usager en faisant pointer une variable sur lui (via son DN) et en faisant des put pour écraser le contenu précédent de certains champs. N'oubliez pas le SetInfo à la fin!

Un exercice cliché

Créez un script qui lit un fichier texte formaté ainsi:

```
nomLogin prénom nomFamille telephone cheminProfil motPasse
```

Chaque ligne correspond à un usager à créer. Votre script doit lire le fichier ligne par ligne et créer les comptes dans AD. Inspirez-vous des exemples sur cette page et sur le programme de bottin - ce script sera simplement un croisement entre les deux.