

TP 3 - PowerShell

Objets Active Directory

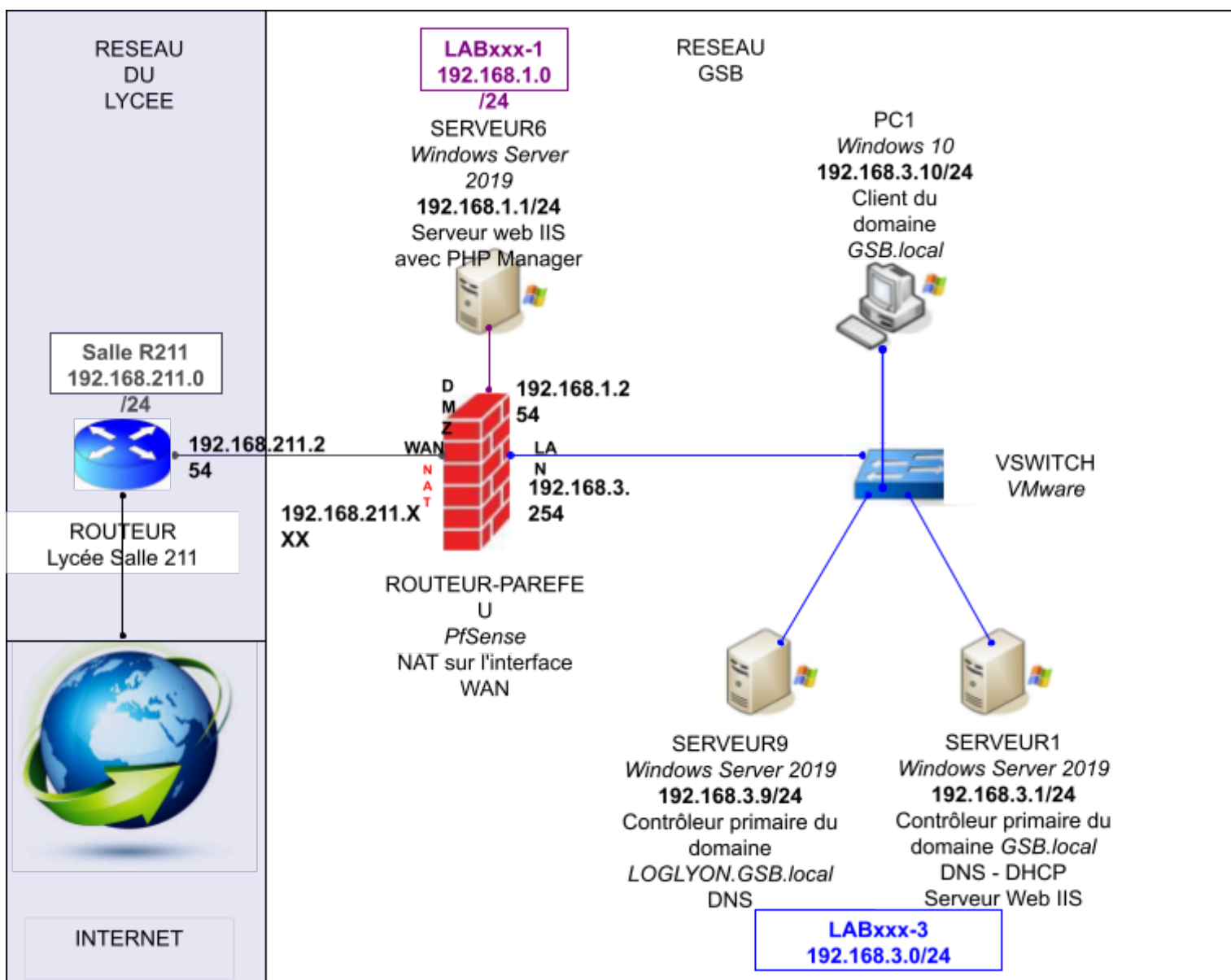
Objectifs

Ecrire des scripts PowerShell utilisant les notions suivantes :

- Test de l'existence d'un objet ActiveDirectory (de classe OU, group, user, computer, ...)
- Création d'un objet ActiveDirectory (de classe OU, group, user, computer, ...)

Schéma du réseau

Les scripts sont à créer sur le serveur SERVEUR1 contrôleur de domaine GSB.local.



Travail à faire : Script *CreationUnEtudiant.ps1*

Dans l'Active Directory *GSB.local*, créer l'utilisateur *Amélie Nothomb* (de login *anothomb* et de mot de passe *secret1A!*) dans l'OU *Etudiants* (OU à créer si inexistante).

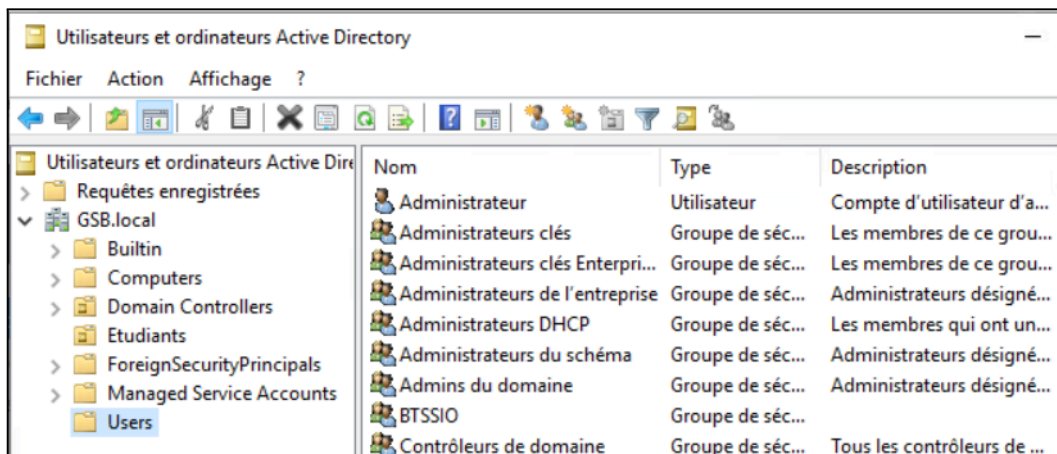
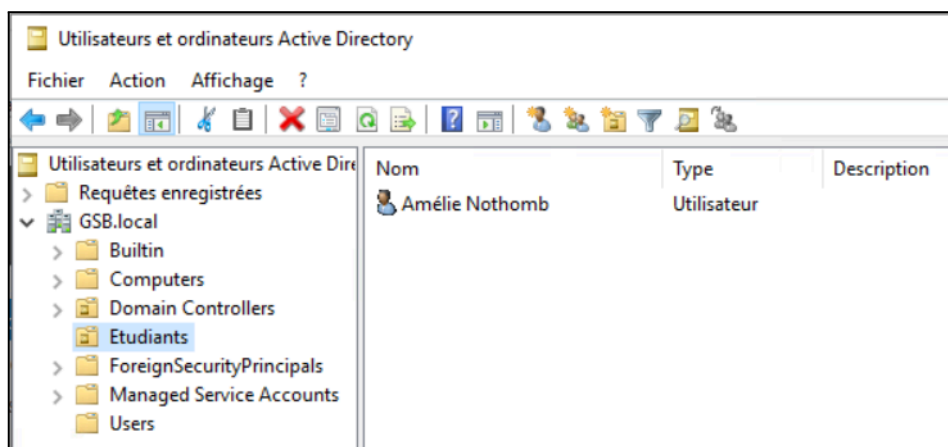
Cet utilisateur doit être membre du groupe *BTSSIO* (groupe à créer dans le conteneur *Users* si inexistant).

Indications :

- on testera l'existence de l'OU *Etudiants* dans l'Active Directory *GSB.local*; si celle-ci n'existe pas, on la créera.

- on testera l'existence du groupe *BTSSIO* dans le conteneur *Users*; si celui-ci n'existe pas, on le créera.

- on testera l'existence de l'utilisateur *Amélie Nothomb* dans l'Active Directory *GSB.local*; si celle-ci n'existe pas, on la créera dans l'OU *Etudiants* et dans le groupe *BTSSIO*.



Travail à faire : Script *CreationUnEtudiant.ps1*

Dans le cadre du module d'administration des systèmes et des réseaux , l'objectif de ce TP est de manipuler des objets Active Directory via PowerShell.

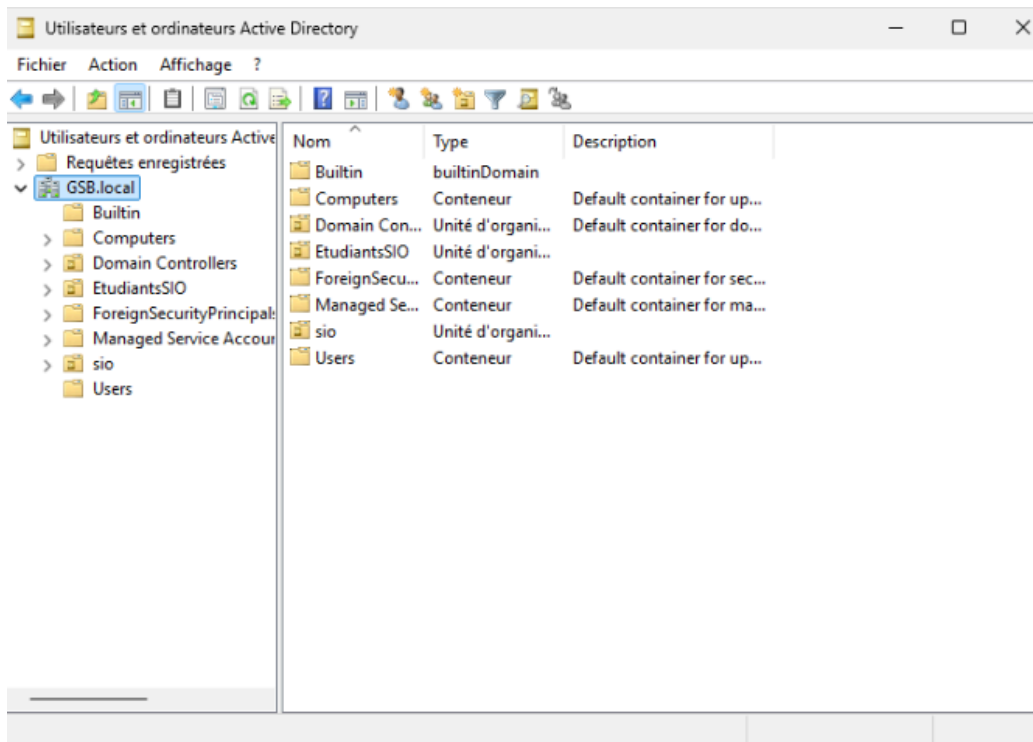
Le but est d'écrire un script permettant de tester l'existence d'objets (OU, groupe, utilisateur) et de les créer s'ils sont absents.

Les manipulations s'effectuent sur le contrôleur de domaine principal (SERVEUR1) du domaine GSB.local.

État initial de l'environnement

Avant le lancement du script, une vérification de la console **Utilisateurs et ordinateurs Active Directory** permet de constater l'état d'origine de la structure du domaine **GSB.local**.

- La racine contient les conteneurs et **OUs** par défaut tels que **Builtin**, **Computers**, **Domain Controllers**, **ForeignSecurityPrincipals**, **Managed Service Accounts** et **Users**.
- L'Unité d'Organisation cible **Etudiants** n'existe pas encore.



Analyse du script PowerShell

Le script est développé pour automatiser la création tout en évitant les erreurs liées aux doublons. Il utilise les commandes `dsquery` pour tester l'existence des objets et `dsadd` pour les créer.

- **Initialisation des variables** : Les variables `$myOU = "Etudiants"` et `$myGroup = "BTSSIO"` sont définies pour cibler les objets à traiter.
- **Gestion de l'Unité d'Organisation** : Une condition `if` teste l'existence de l'OU. Si elle est introuvable, la commande `dsadd ou` est exécutée pour la générer. Dans le cas contraire, un message d'attention est affiché pour signaler que l'OU existe déjà.
- **Gestion du Groupe** : Une logique similaire est appliquée avec `dsquery group` pour vérifier la présence du groupe dans l'Active Directory. S'il n'existe pas, `dsadd group` procède à sa création dans le conteneur cible.
- **Création de l'Utilisateur** : Enfin, l'utilisateur **Amelie Nothomb** est créé avec `dsadd user`. La commande intègre directement les paramètres requis : l'UPN `anotherb@GSB.local`, l'activation du compte avec `-disabled no`, l'attribution du mot de passe `secret1A!` et l'ajout au groupe via le paramètre `-memberof`

```
$myDomainSLD="GSB" # Second Level Domain
$myOU="Etudiants"
$myGroup="BTSSIO"

if (!(Dsquery ou -name $myOU "dc=$myDomainSLD,dc=$myDomainTLD"))
{
    dsadd ou "ou=$myOU,dc=$myDomainSLD,dc=$myDomainTLD"
}
else
{
    write-host "Attention : l'OU $myOU existe déjà dans dc=$myDomainSLD,dc=$myDomainTLD !"
}

if (!(Dsquery group -samid "$myGroup"))
{
    dsadd group -scope 1 "cn=$myGroup,cn=Users,dc=$myDomainSLD,dc=$myDomainTLD"
}
else
{
    write-host "Attention : le groupe $myGroup existe déjà !"
}

dsadd user cn="Amelie Nothomb,ou=$myOU,dc=$myDomainSLD,dc=$myDomainTLD" -upn "anotherb@GSB.local" -pwd "secret1A!" -disabled no -memberof "cn=BTSSIO,cn=Users,dc=$myDomainSLD,dc=$myDomainTLD"
```

Analyse du script PowerShell

Le script a été lancé depuis la console PowerShell en tant qu'Administrateur, via le chemin `C:\Users\Administrateur\Documents\CreationUnEtudiant.ps1`. Les retours de la console confirment le bon déroulement des opérations avec trois messages de succès consécutifs :

- `dsadd` réussite pour la création de l'Unité d'organisation `ou=Etudiants, dc=GSB, dc=local`.
- `dsadd` réussite pour la création du groupe `cn=BTSSIO` dans le conteneur `cn=Users`.
- `dsadd` réussite pour l'utilisateur `cn=Amelie Nothomb` au sein de l'OU `Etudiants`.

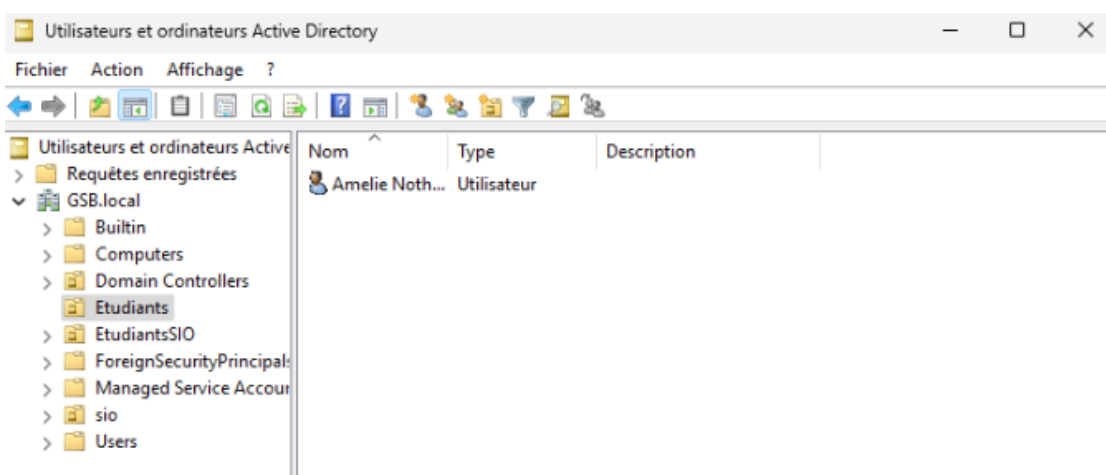
```
PS C:\WINDOWS\system32> C:\Users\Administrateur\Documents\CreationUnEtudiant.ps1
dsadd r,ussite:ou=Etudiants,dc=GSB,dc=local
dsadd r,ussite:cn=BTSSIO,cn=Users,dc=GSB,dc=local
dsadd r,ussite:cn=Amelie Nothomb,ou=Etudiants,dc=GSB,dc=local

PS C:\WINDOWS\system32>
```

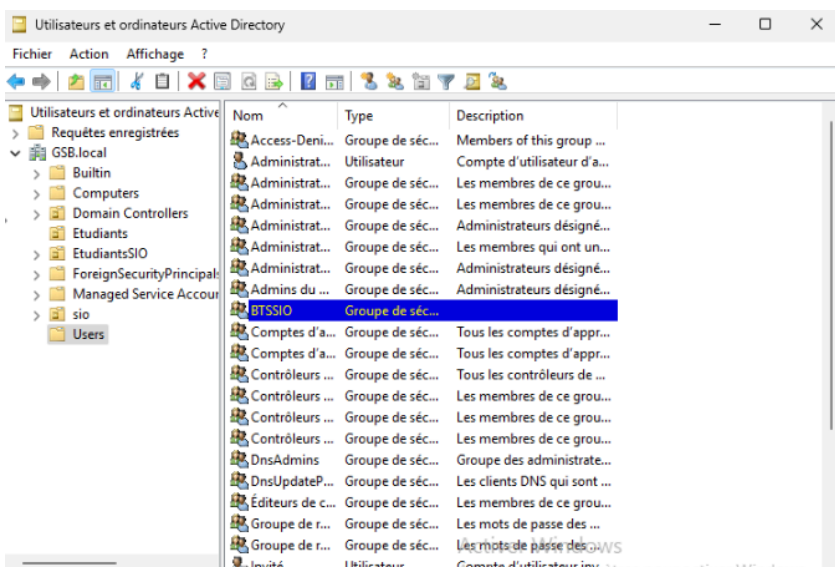
Vérifications Post-Exécution dans l'interface graphique

Afin de valider la bonne application des modifications par le script, des contrôles ont été effectués directement dans la console **Utilisateurs et ordinateurs Active Directory**.

- **Vérification de l'OU et de l'utilisateur** : L'Unité d'Organisation **Etudiants** est désormais visible à la racine du domaine **GSB.local**. À l'intérieur de celle-ci, le compte utilisateur **Amelie Nothomb** est bien présent.



- **Vérification du groupe** : En explorant le conteneur **Users**, le groupe de sécurité **BTSSIO** a bien été généré.



- **Vérification de l'appartenance au groupe** : L'ouverture des propriétés du groupe **BTSSIO**, dans l'onglet **Membres**, confirme que l'utilisateur **Amelie Notho** (Amélie Nothomb) y a été correctement intégré par le script.

