

TP IMPORT CSV SUR POWERSHELL

Compte rendu sur l'utilisation de l'import CSV sur PowerShell

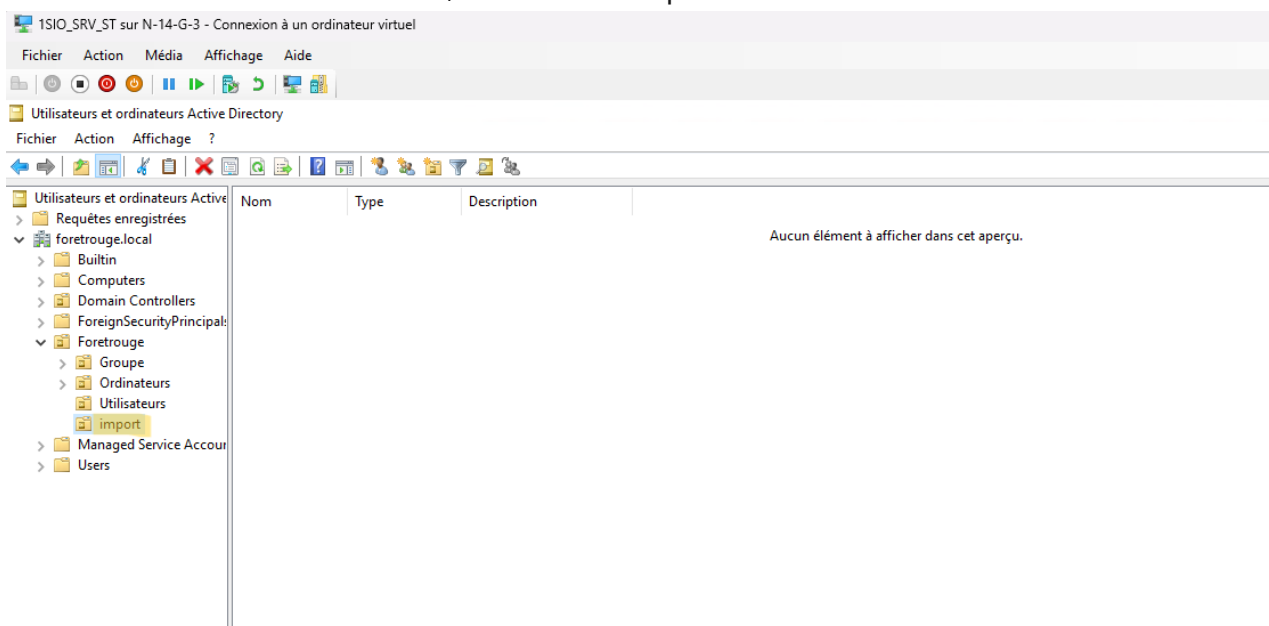
TEDE SACHA
MR ROTH

PowerShell est un interpréteur de commandes moderne qui comprend les meilleures fonctionnalités d'autres interpréteurs de commandes populaires. Contrairement à la plupart d'entre eux, qui acceptent et retournent uniquement du texte, PowerShell accepte et retourne des objets.

En effet, dans ce compte rendu nous allons voir comment l'utiliser dans un premier temps, puis par la suite on va importer un fichier CSV sur PowerShell ISE qu'on va adapter.

En tant que langage de script, PowerShell est souvent utilisé pour automatiser la gestion des systèmes. Il sert également à créer, à tester et à déployer des solutions, souvent dans des environnements CI/CD. Il agit sur des « objets »

Première étape, créer une unité d'organisation que l'on va appeler « import ». Pour l'instant on le met de côté, il nous servira plus tard.



Une fois cela fait, on se dirige vers le dossier « script » qu'on vient de créer directement dans notre Disque local. On y ajoute notre fichier CSV, et un fichier PS1 qu'on a renommé Import Utilisateurs.

Ce PC > Disque local (C:) > Script					Rechercher dans : Script
	Nom	Modifié le	Type	Taille	
	IMport Utilisateurs.ps1	03/04/2024 09:18	Document texte	0 Ko	
	Utilisateurs.csv	03/04/2024 09:22	Fichier CSV	1 Ko	

Dans notre fichier CSV, on dresse la liste des utilisateurs que l'on veut ajouter. ATTENTION à bien l'écrire de manière a ce que PowerShell comprenne ce que l'on veut, par exemple (FirstName est bien écrit alors que firstname est mal écrit)

Utilisateurs.csv - Bloc-notes

```
Fichier  Edition  Format  Affichage  Aide
FirstName,LastName,UserName,Password
Thomas,Boire,thomas.boire,Azerty123
Mouss,Manger,mouss.manger,Azerty123
Miguel,Artip,miguel.artip,Azerty123
Theo,Tobi,theo.tobi,Azerty123
Arnaud,Tovic,arnaud.tovic,Azerty123
Leo,Duroi,leo.duroi,Azerty123|
```

On réalise le script, on l'adapte et après cela on va exécuter le script afin de voir s'il fonctionne ou pas.

```

Sans titre2.ps1* X
1  # Variables
2  $csvPath = "C:\Script\Utilisateurs.csv"
3  $ouPath = "OU=import,Ou=Foretrouge,DC=foretrouge,DC=local"
4
5  # Importer le module Active Directory
6  Import-Module ActiveDirectory
7
8  # Lire le fichier CSV et ajouter les utilisateurs
9  $users = Import-Csv -Path $csvPath
10
11 foreach ($user in $users) {
12     $password = ConvertTo-SecureString $user.password -AsPlainText -Force
13     $userProps = @{
14         'Name' = $user.FirstName + " " + $user.LastName
15         'GivenName' = $user.FirstName
16         'Surname' = $user.LastName
17         'SamAccountName' = $user.UserName
18         'UserPrincipalName' = $user.UserName + " foretrouge.local "
19         'DisplayName' = $user.FirstName + " " + $user.LastName
20         'Path' = $ouPath
21         'AccountPassword' = $password
22         'Enabled' = $false
23         'ChangePasswordAtLogon' = $true
24     }
25     New-ADUser @userProps
26 }
27
28 Write-Host " Importation des utilisateurs terminée. "
29

```

On s'aperçoit que l'exécution du script fonctionne parfaitement.

```

Importation des utilisateurs terminée.
PS C:\Users\Administrateur> # Variables
$csvPath = "C:\Script\Utilisateurs.csv"
$ouPath = "OU=import,Ou=Foretrouge,DC=foretrouge,DC=local"

# Importer le module Active Directory
Import-Module ActiveDirectory

# Lire le fichier CSV et ajouter les utilisateurs
$users = Import-Csv -Path $csvPath

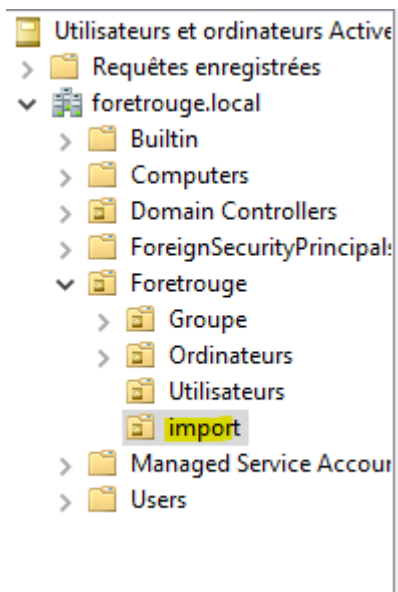
foreach ($user in $users) {
    $password = ConvertTo-SecureString $user.password -AsPlainText -Force
    $userProps = @{
        'Name' = $user.FirstName + " " + $user.LastName
        'GivenName' = $user.FirstName
        'Surname' = $user.LastName
        'SamAccountName' = $user.UserName
        'UserPrincipalName' = $user.UserName + " @geek-advisor.local "
        'DisplayName' = $user.FirstName + " " + $user.LastName
        'Path' = $ouPath
        'AccountPassword' = $password
        'Enabled' = $false
        'ChangePasswordAtLogon' = $true
    }
    New-ADUser @userProps
}

Write-Host " Importation des utilisateurs terminée. "
Importation des utilisateurs terminée.

PS C:\Users\Administrateur>

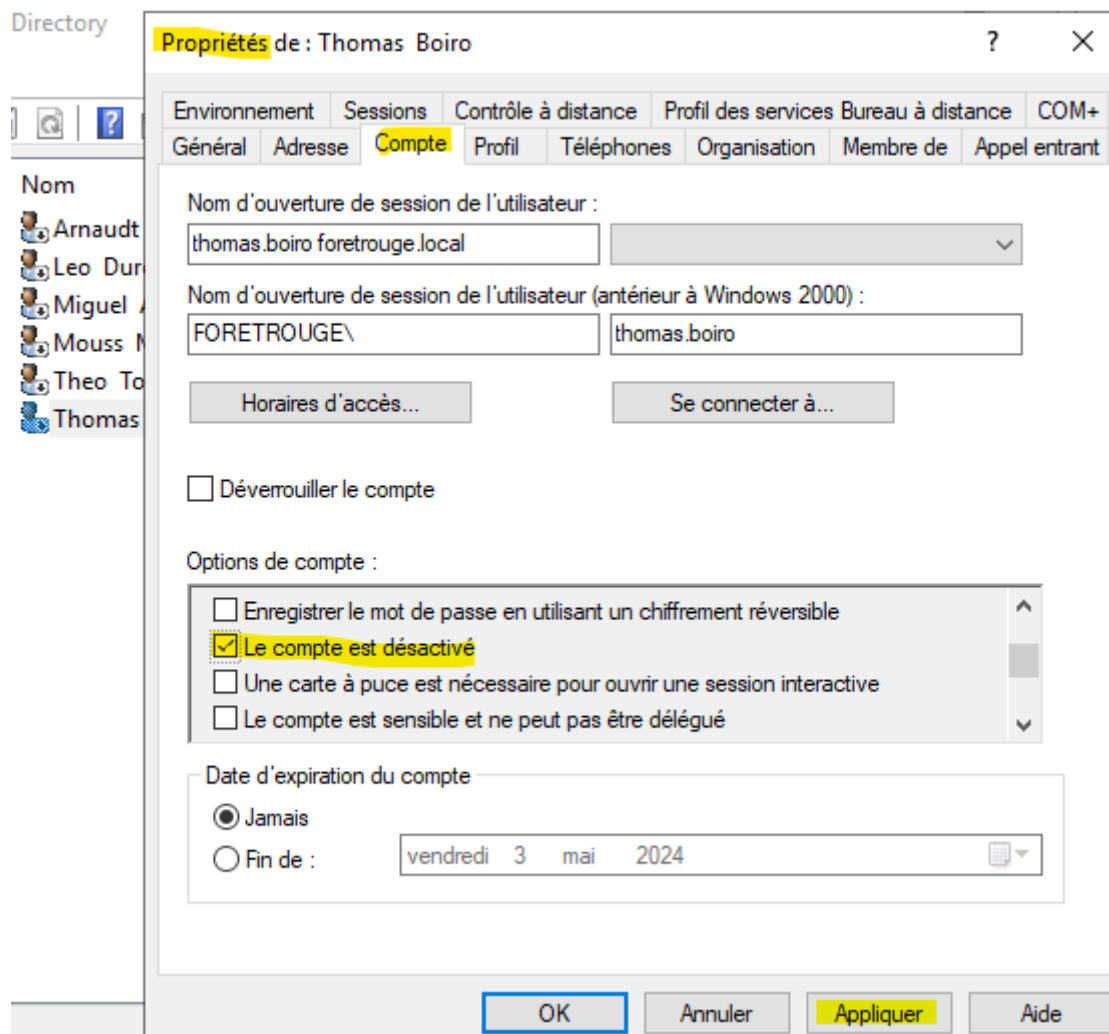
```

Ensuite, nous allons vérifier si on aperçoit bien les utilisateurs dans le dossier « import » que l'on a créé au début et s'ils apparaissent dedans, alors le script a bel et bien fonctionné correctement.

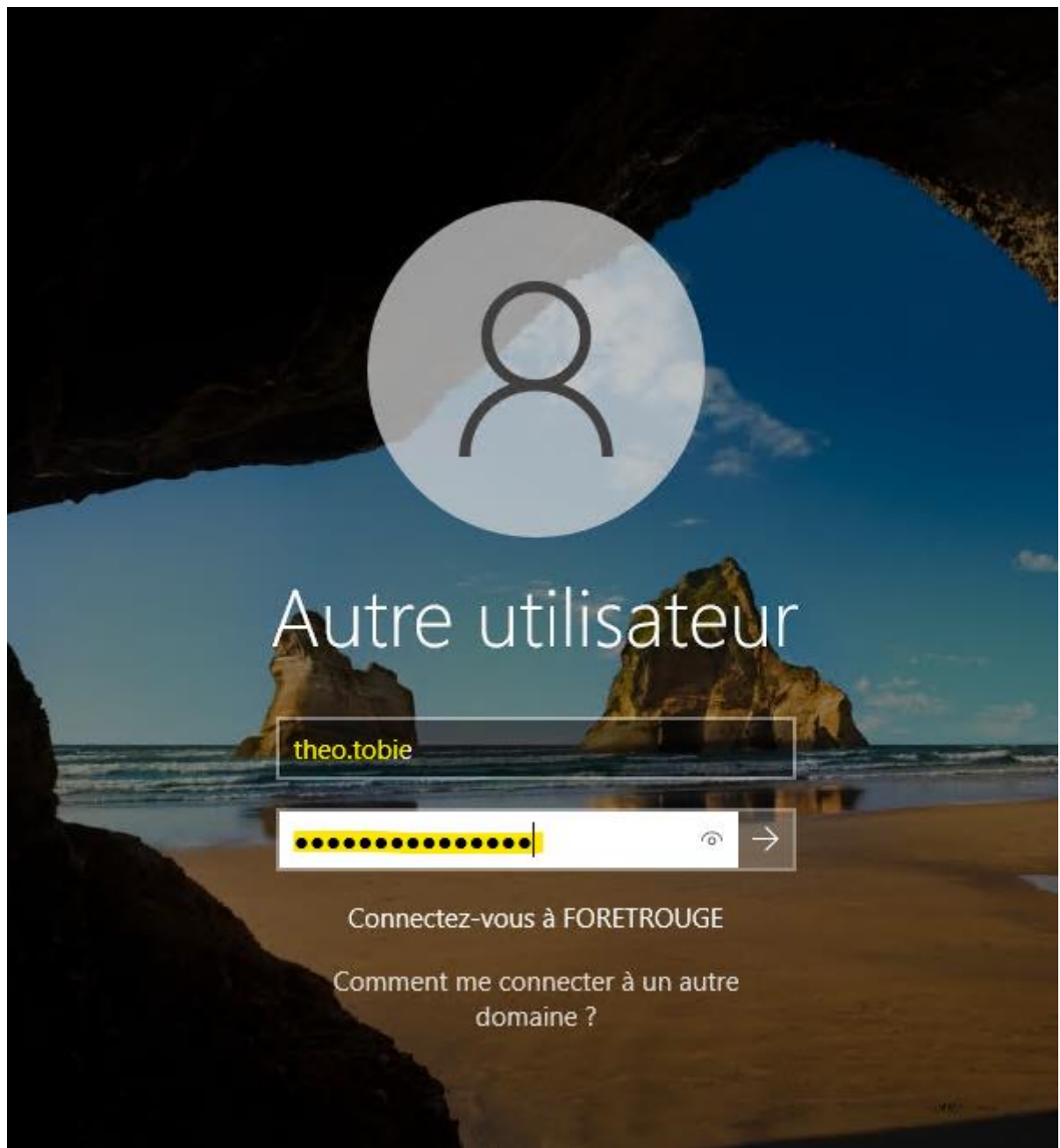


Nom	Type
Arnaudt To...	Utilisateur
Leo Durois	Utilisateur
Miguel Artipe	Utilisateur
Mouss Man...	Utilisateur
Theo Tobie	Utilisateur
Thomas Boi...	Utilisateur

Par rappel de sécurité, pourquoi pas laisser l'utilisateur désactivé jusqu'à sa première connexion, car un mot de passe prédéfini peut être parfois simple à trouver et pour ce faire, vous allez dans « propriété » du compte à activer ou à désactiver, puis dans la rubrique « compte » pour ensuite, cocher ou décocher la case « le compte est désactivé » (voir capture d'écran ci-dessous).



Pour finir, on se connecte sur le compte d'un utilisateur créé.



Puis ça se lance.

Cette opération peut durer plusieurs minutes

N'éteignez pas votre PC.

Pour conclure,

Premièrement, l'import d'un fichier CSV se réalise très simplement avec PowerShell. En seulement quelques minutes j'ai pu réaliser cette manipulation.

Le fichier CSV s'implémente parfaitement avec Active Directory et offre un gain de temps inimaginable, chose qui n'est pas forcément garanti avec d'autres langages.

Quelques consignes de sécurité à suivre pour que la manipulation se passe bien :

- Bien pensé à ne pas se tromper dans la rédaction des catégories sur le fichier CSV sous risque que le script s'accorde mal.
- Ce n'est pas obligatoire mais conseillé de laisser désactiver l'utilisateur jusqu'à ce que celui-ci se connecte pour la première fois.
- Il faut faire attention aux petits détails dans le script par exemple, les majuscules ou encore les caractères spéciaux etc.