

RAPPORT DE STAGE 1ERE ANNEE SIO

Rapport de stage - Semaine 1 :

Durant ma première semaine de stage, j'ai travaillé sur la création d'un programme en PowerShell. Ce programme a pour objectif de récupérer les autorisations des utilisateurs et des groupes pour un dossier spécifique, puis de les présenter dans un tableau Excel. Afin de faciliter la lecture et de préparer une future réorganisation de la structure de l'entreprise, j'ai transféré ces informations dans un autre tableau Excel plus clair. Avant d'appliquer ce programme aux fichiers de l'entreprise, j'ai effectué de nombreux tests et améliorations en utilisant mes propres documents pour affiner le script.

J'avoue j'ai eu besoin d'aide de ChatGPT, car je n'avais pas beaucoup de connaissances en PowerShell. Cependant, j'ai dû faire plusieurs ajustements aux extraits de code suggérés par l'IA et effectuer des recherches sur Internet pour obtenir le résultat que je souhaitais. J'ai demandé à ChatGPT de rajouter des commentaires dans le code pour mieux comprendre ce que le script fait.

```
1 # Chemin du répertoire parent dont vous souhaitez extraire les droits ACL, y compris un sous-dossier
2 $cheminRepertoireParent = "A:\32 GC2M"
3
4 # Récupérer les sous-répertoires immédiats du répertoire parent
5 $sousRepertoires = Get-Childitem -Path $cheminRepertoireParent -Directory
6
7 # Créer une ArrayList pour stocker les données des droits ACL
8 $aclData = New-Object System.Collections.ArrayList
9
10 # Fonction récursive pour extraire les informations ACL
11 function Extract-ACL {
12     param (
13         [string]$Repertoire,
14         [int]$Profondeur
15     )
16
17     # Récupérer les informations ACL pour le répertoire actuel
18     $acl = Get-Acl -Path $Repertoire
19
20     # Parcourir les entrées ACL
21     foreach ($entry in $acl.Access) {
22         $aclEntry = New-Object -TypeName PSObject
23         $aclEntry | Add-Member -MemberType NoteProperty -Name "Repertoire" -Value $Repertoire
24         $aclEntry | Add-Member -MemberType NoteProperty -Name "utilisateur / groupe" -Value $entry.IdentityReference
25         $aclEntry | Add-Member -MemberType NoteProperty -Name "Type d'accès" -Value $entry.FileSystemRights
26         $aclEntry | Add-Member -MemberType NoteProperty -Name "Autorisations" -Value $entry.AccessControlType
27         $aclEntry | Add-Member -MemberType NoteProperty -Name "Herite" -Value $entry.IsInherited
28         $aclData.Add($aclEntry) | Out-Null
29     }
30
31     # Vérifier la profondeur pour éviter une récursion excessive
32     if ($Profondeur -gt 0) {
33         $subRepertoires = Get-Childitem -Path $Repertoire -Directory
34         foreach ($subRepertoire in $subRepertoires) {
35             Extract-ACL -Repertoire $subRepertoire.FullName -Profondeur ($Profondeur - 1)
36         }
37     }
38 }
39
40 # Appeler la fonction Extract-ACL pour le répertoire parent spécifié
41 foreach ($sousRepertoire in $sousRepertoires) {
42     Extract-ACL -Repertoire $sousRepertoire.FullName -Profondeur 0
43 }
44
45 # Chemin du fichier CSV pour enregistrer les résultats
46 $cheminFichierCSV = "C:\Users\s bendjelida\Desktop\arborescence csv\32 GC2Mfinalfinal.csv"
47
48 # Convertir l'objet aclData en format CSV et l'enregistrer dans le fichier CSV
49 $aclData | Export-Csv -Path $cheminFichierCSV -NotypeInformation
50
51 Write-Host "Les droits ACL ont été extraits, y compris les sous-répertoires, et enregistrés dans le fichier CSV : $cheminFichierCSV"
52
```

Rapport de stage - Semaine 2 :

Lors de ma deuxième semaine de stage, j'ai créé un programme en PowerShell pour récupérer les groupes d'utilisateurs du NAS de l'entreprise et les afficher dans un tableau Excel j'ai transféré ces informations vers un autre tableau Excel, optimisé pour une meilleure lisibilité clair. J'ai effectué des tests et des améliorations pour rendre le script plus efficace avant de le déployer.

```
1
2 # Get year and month for CSV export file
3 $DateTime = Get-Date -f "yyyyMMddhhmm"
4
5 # Set CSV file name
6 $CSVFile = "C:\temp\ADGroups_" + $DateTime + ".csv"
7
8 # Set distinguishedName as searchbase, you can use one DN or multiple DNS
9 # Or use the root domain like DC=exoip,DC=local
10 $DNS = @(
11     "DC=exoip,DC=local"
12 )
13
14 # Create empty array for CSV data
15 $CSVOutput = @()
16
17 # Create empty array for AD groups
18 $ADGroups = @()
19
20 # Loop through DNS
21 foreach ($DN in $DNS) {
22     # Add every DN to AD groups
23     $ADGroups += Get-ADGroup -Filter * -SearchBase $DN
24 }
25
26 # Set progress bar variables
27 $i = 0
28 $tot = $ADGroups.count
29
30 foreach ($ADGroup in $ADGroups) {
31     # Set up progress bar
32     $i++
33     $status = "{0:N0}" -f ($i / $tot * 100)
34     Write-Progress -Activity "Exporting AD Groups" -status "Processing Group $i of $tot : $status% Completed" -PercentComplete ($i / $tot * 100)
35
36     # Ensure Members variable is empty
37     $Members = ""
38
39     # Get group members which are also groups and add to string
40     $MembersArr = (Get-ADGroup -filter { Name -eq $ADGroup.Name } -Properties Members).Members | Get-ADObject | select Name, objectClass, distinguishedName
41     if ($MembersArr) {
42         foreach ($Member in $MembersArr) {
43             $MemDN = $Member.distinguishedName
44             $UserObj = Get-ADUser -filter { DistinguishedName -eq $MemDN }
45             if ($UserObj.Enabled -eq $False) {
46                 continue
47             }
48             if ($Member.objectClass -eq "user") {
49                 $Members = $Members + ",U-" + $Member.Name
50             }
51             elseif ($Member.objectClass -eq "contact") {
52                 $Members = $Members + ",C-" + $Member.Name
53             }
54             else {
55                 $Members = $Members + ",G-" + $Member.Name
56             }
57         }
58     }
59     # Check for members to avoid error for empty groups
60     if ($Members) {
61         $Members = $Members.Substring(1, ($Members.Length) - 1)
62     }
63 }
64
65 # Set up hash table and add values
66 $HashTab = $null
67 $HashTab = [ordered]@{
68     "Name" = $ADGroup.Name
69     "Category" = $ADGroup.GroupCategory
70     "Scope" = $ADGroup.GroupScope
71     "Members" = $Members
72 }
73
74 # Add hash table to CSV data array
75 $CSVOutput += New-Object PSObject -Property $HashTab
76
77 }
78
79 # Export report to CSV file
80 $CSVOutput | Sort-Object Name | Export-Csv -Encoding UTF8 -Path $CSVFile -NoTypeInformation #-Delimiter ";"
```

Durant cette semaine de stage, j'ai également eu l'occasion d'effectuer des dépannages informatiques au sein de l'entreprise. Par exemple, lorsque quelqu'un rencontrait un problème avec son ordinateur, les personnes se tournaient vers moi afin que je puisse résoudre ces problèmes.

De plus, j'ai également été chargé de remplacer certains postes informatiques. Cela consistait à échanger les ordinateurs obsolètes ou défectueux des employés, à réinitialiser les machines et à les préparer avant leur affectation. Cela impliquait de donner accès à certains éléments de l'entreprise et d'installer les programmes de base nécessaires.

Pendant mon stage, j'ai donc pu mettre en pratique mes compétences en dépannage informatique et en gestion des postes de travail, en veillant à ce que les employés disposent d'ordinateurs fonctionnels et adaptés à leurs besoins

Rapport de stage - Semaine 3 :

Pendant ma troisième semaine de stage, je me suis occupé de la préparation des équipements technologiques pour les salariés. J'ai configuré et préparé les téléphones en installant les applications nécessaires et en garantissant leur bon fonctionnement. J'ai aussi contribué à la mise en place des ordinateurs pour les nouveaux employés en installant le système d'exploitation et les logiciels requis.

De plus, j'ai apporté un soutien technique en résolvant les problèmes informatiques rencontrés par les employés, que ce soit au niveau matériel ou logiciel. J'ai également effectué des tests sur les écrans et les téléphones obsolètes, prenant les mesures nécessaires pour les remplacer, réparer ou recycler.

En parallèle, j'ai mis à jour l'inventaire des équipements informatiques de l'entreprise. Enfin, j'ai contribué à l'amélioration de l'efficacité de l'organisation en fournissant des informations sur les tableaux croisés dynamiques et la recherche V, du moins je me suis renseigné dessus pour avoir des compétences en Excel plus poussées.

Ces différentes responsabilités m'ont permis de développer mes compétences en préparation d'équipements technologiques, en dépannage informatique et en gestion de l'inventaire, tout en soutenant les employés dans l'utilisation efficace de leurs outils technologiques.

Rapport de stage - Semaine 4 :

Cette semaine, j'ai effectué du dépannage informatique en résolvant un bug sur le NAS de l'entreprise. Plusieurs personnes étaient confrontées à un problème où le NAS tournait en boucle sans accès. La solution était simplement de désinstaller un antivirus qui bloquait l'accès. J'ai également réalisé des réinitialisations d'ordinateurs et installé des équipements tels que des écrans et des claviers pour certains employés.

Rapport de stage - Semaine 5 :

Au cours de cette semaine, j'ai principalement effectué le remplacement d'ordinateurs obsolètes par des nouveaux. Je n'ai pas eu beaucoup d'autres tâches à réaliser. J'ai également effectué des réparations sur du matériel informatique, comme des "docs" (j'imagine que vous voulez dire des "docs" ou des stations d'accueil ?), J'ai configuré des téléphones portables d'entreprise pour des salariés. Le problème de moulinage rencontré par certaines personnes lors de la quatrième semaine persistait toujours. J'ai donc résolu à nouveau ce problème pour les personnes concernées.