

Nathan

Script de nettoyage

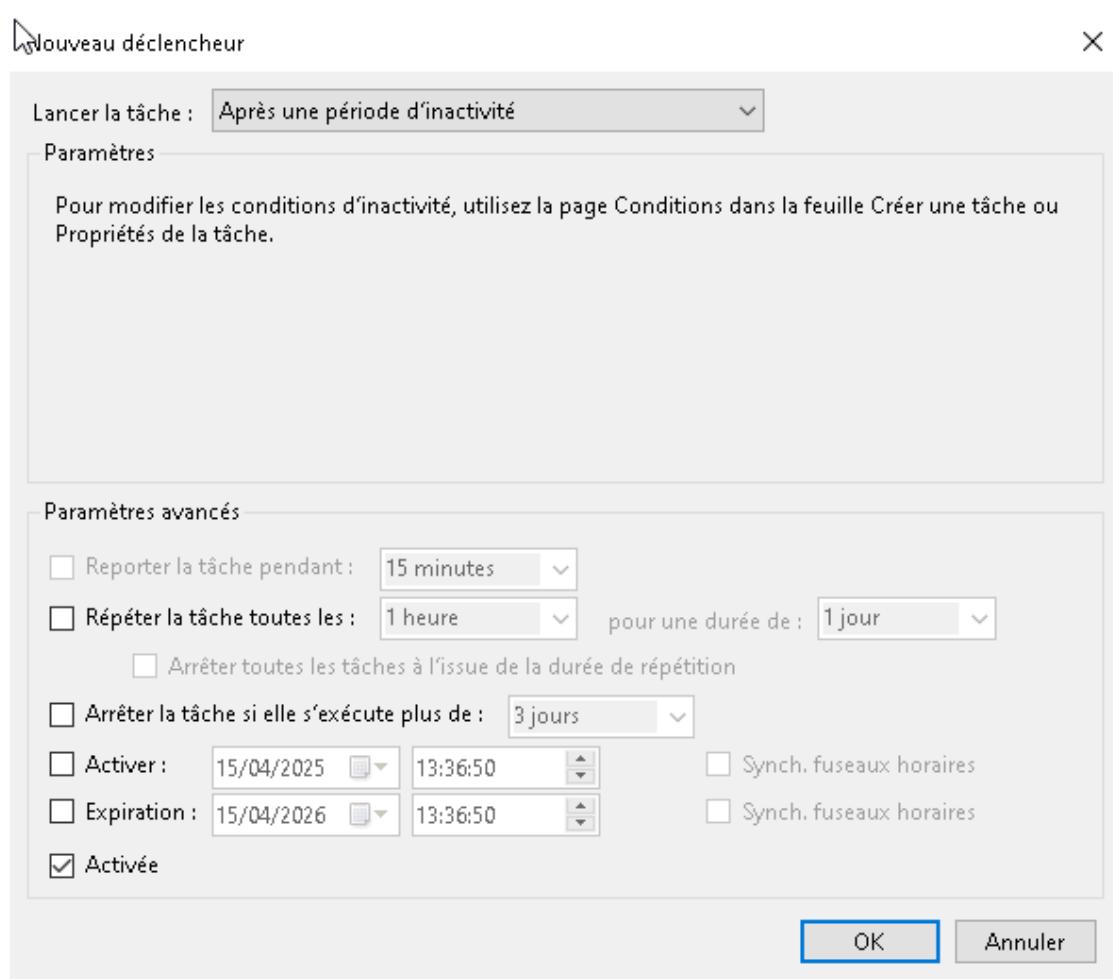
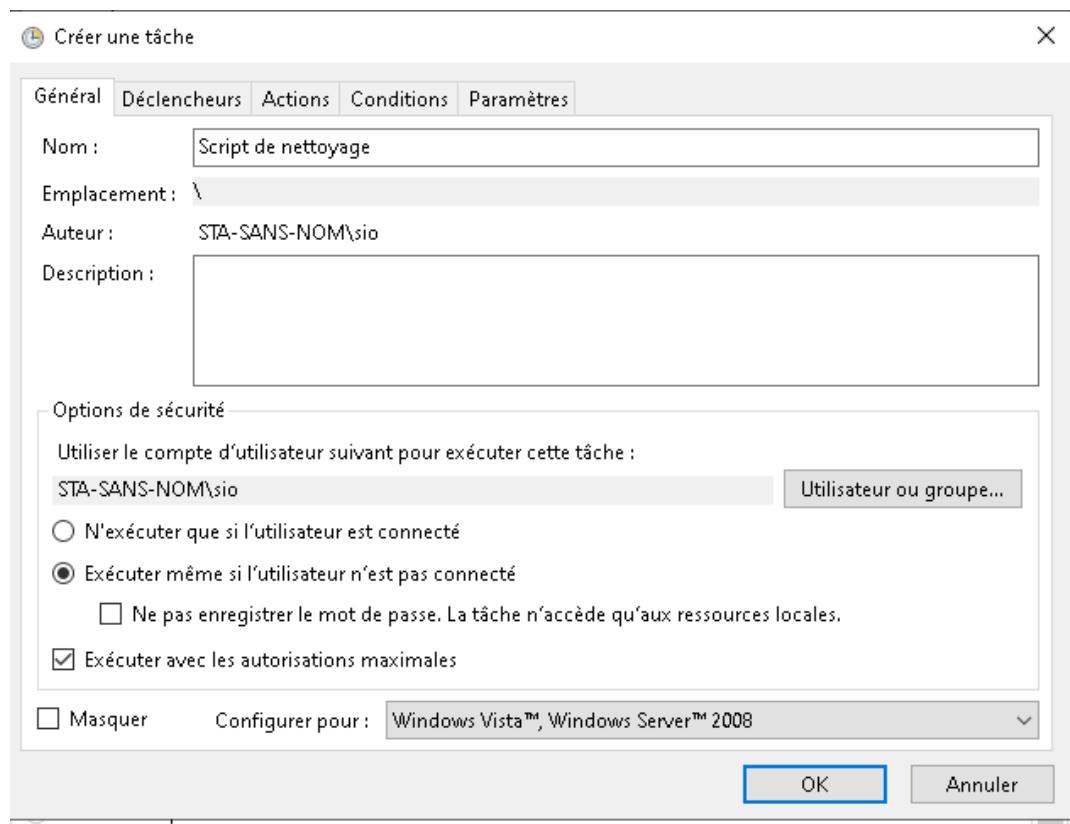
Pour commencer, on écrit le script :

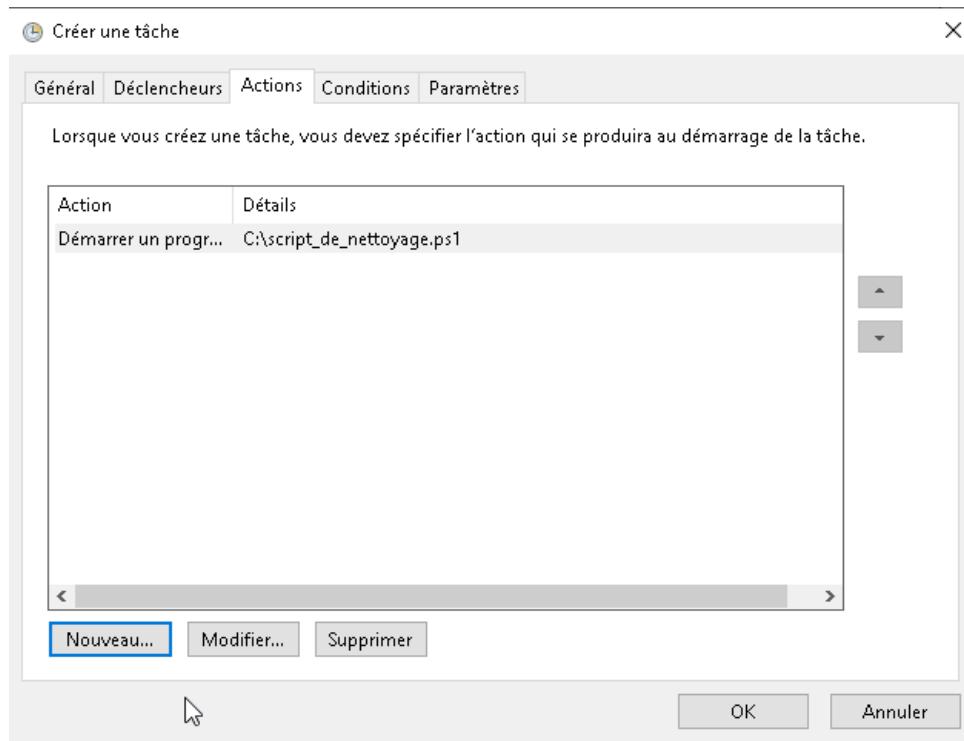
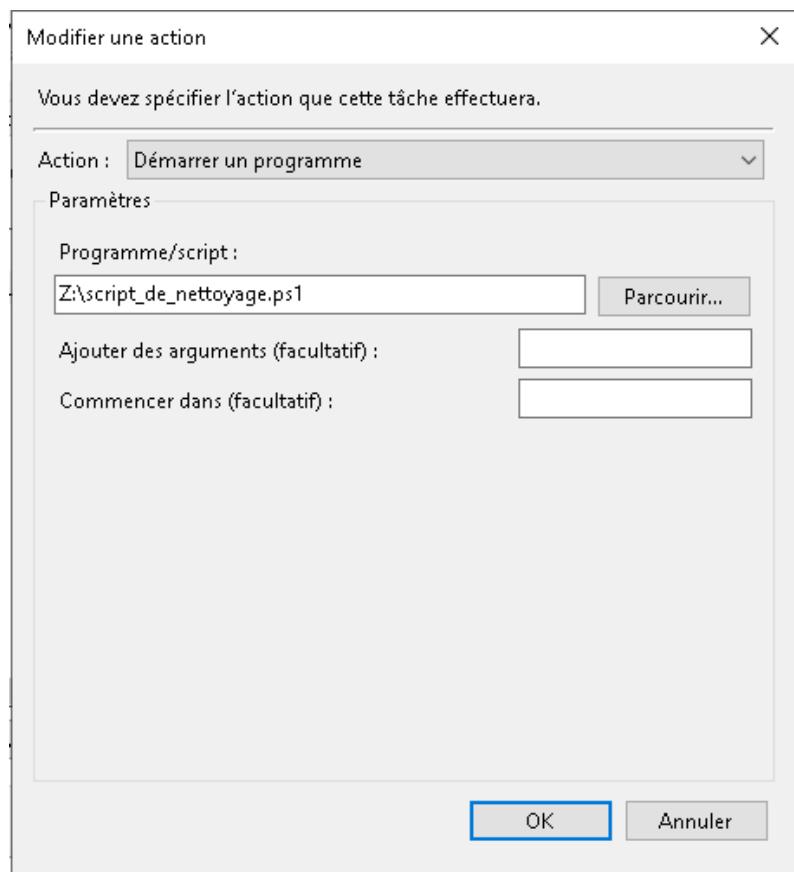
```
script_de_nettoyage.ps1 X
1 # Liste des utilisateurs à vérifier
2 $utilisateurs = @("barbirato", "demeec", "mgaillat")
3
4 # Boucle pour chaque utilisateur
5 foreach ($utilisateur in $utilisateurs) {
6     # Dossier Documents de l'utilisateur sous Linux
7     $dossier = "C:/Users/$utilisateur"
8
9     # Vérifier si le dossier existe
10    if (Test-Path $dossier) {
11        # Récupérer tous les fichiers du dossier
12        $fichiers = Get-ChildItem -Path $dossier -File
13
14        # Boucle sur chaque fichier dans le dossier
15        foreach ($fichier in $fichiers) {
16            # Obtenir le propriétaire du fichier
17            $proprietaire = (Get-Acl $fichier.FullName).Owner
18
19            # Récupérer la dernière connexion de l'utilisateur
20            $derniereConnexion = (whois $proprietaire | Select-String "Last logon").Line
21
22            # Extraire la date de la dernière connexion
23            $dateConnexion = [datetime]::ParseExact($derniereConnexion.Substring(14), "yyyy-MM-dd HH:mm", $null)
24
25            # Calculer la différence de jours
26            $diffJours = (Get-Date) - $dateConnexion
27
28            # Si l'utilisateur ne s'est pas connecté depuis 30 jours, supprimer le fichier
29            if ($diffJours.Days -ge 30) {
30                if (Test-Path $fichier.FullName) {
31                    Remove-Item $fichier.FullName -Force
32                    Write-Output "Fichier supprimé : $($fichier.FullName)"
33                }
34            } else {
35                Write-Output "L'utilisateur $utilisateur s'est connecté récemment. Le fichier n'a pas été supprimé : $($fichier.FullName)"
36            }
37        }
38    } else {
39        Write-Output "Le dossier de l'utilisateur $utilisateur n'existe pas : $dossier"
40    }
41 }
```

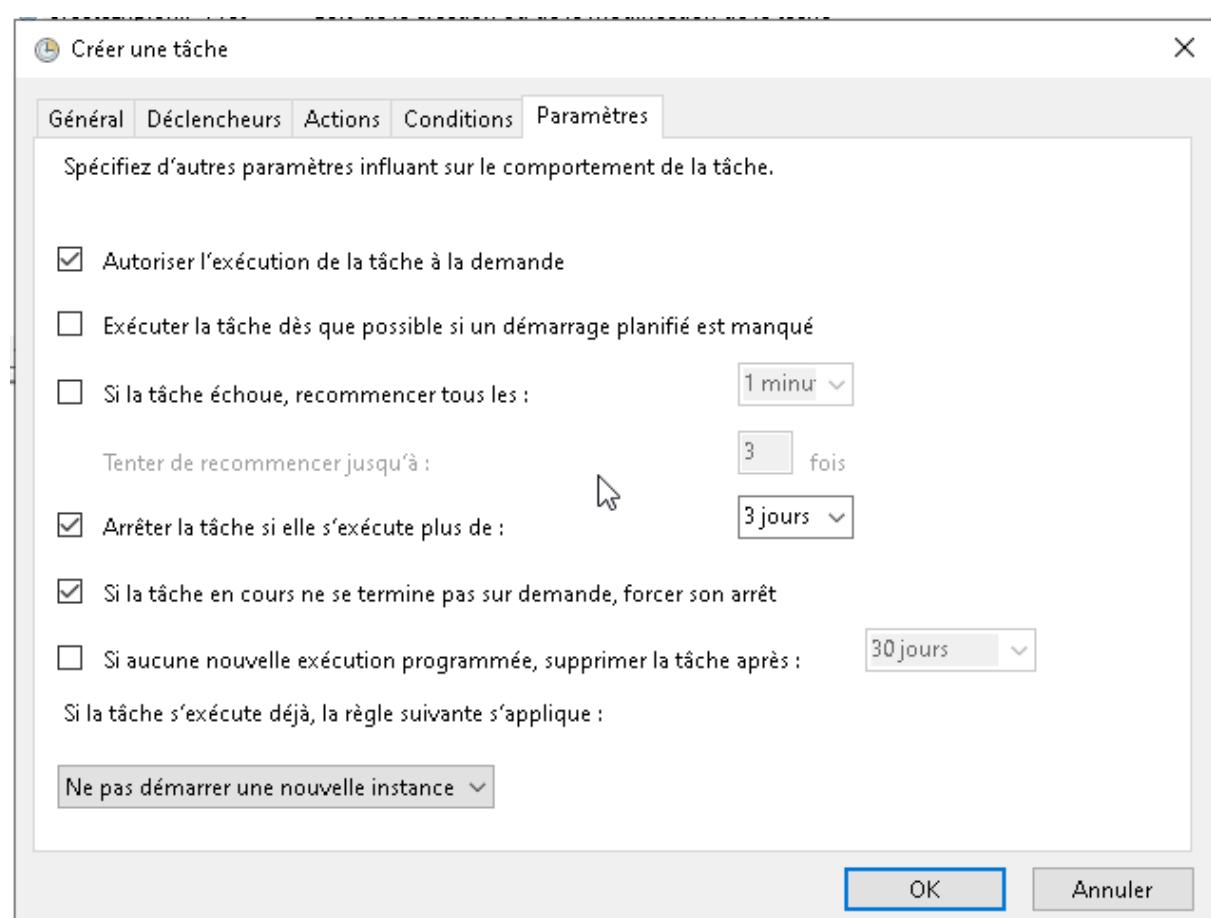
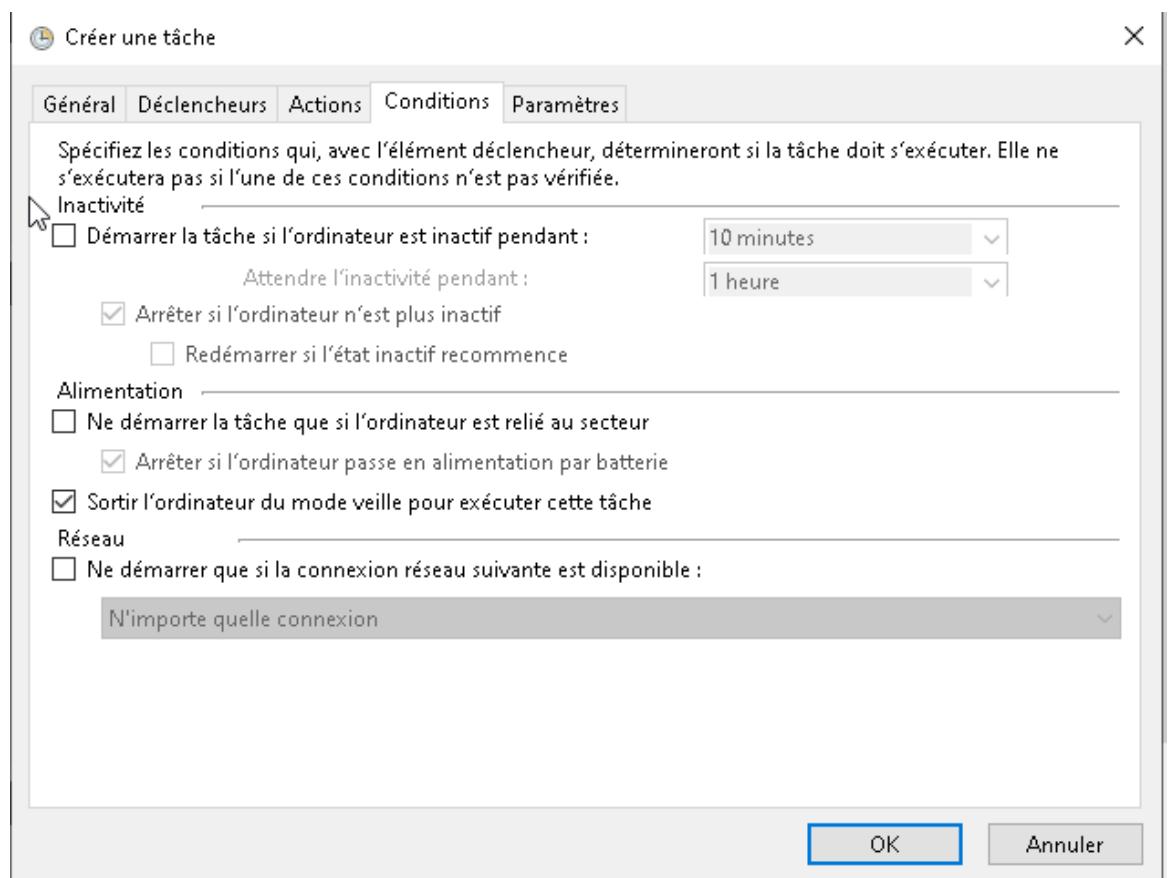
Puis on utilise le planificateur de tâches.

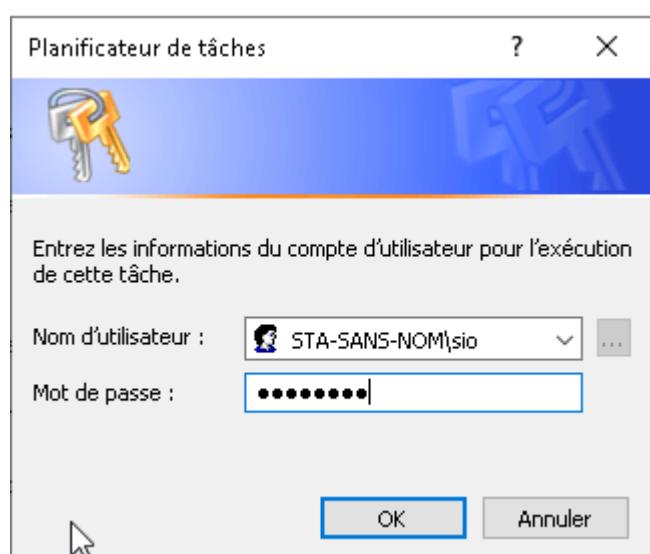
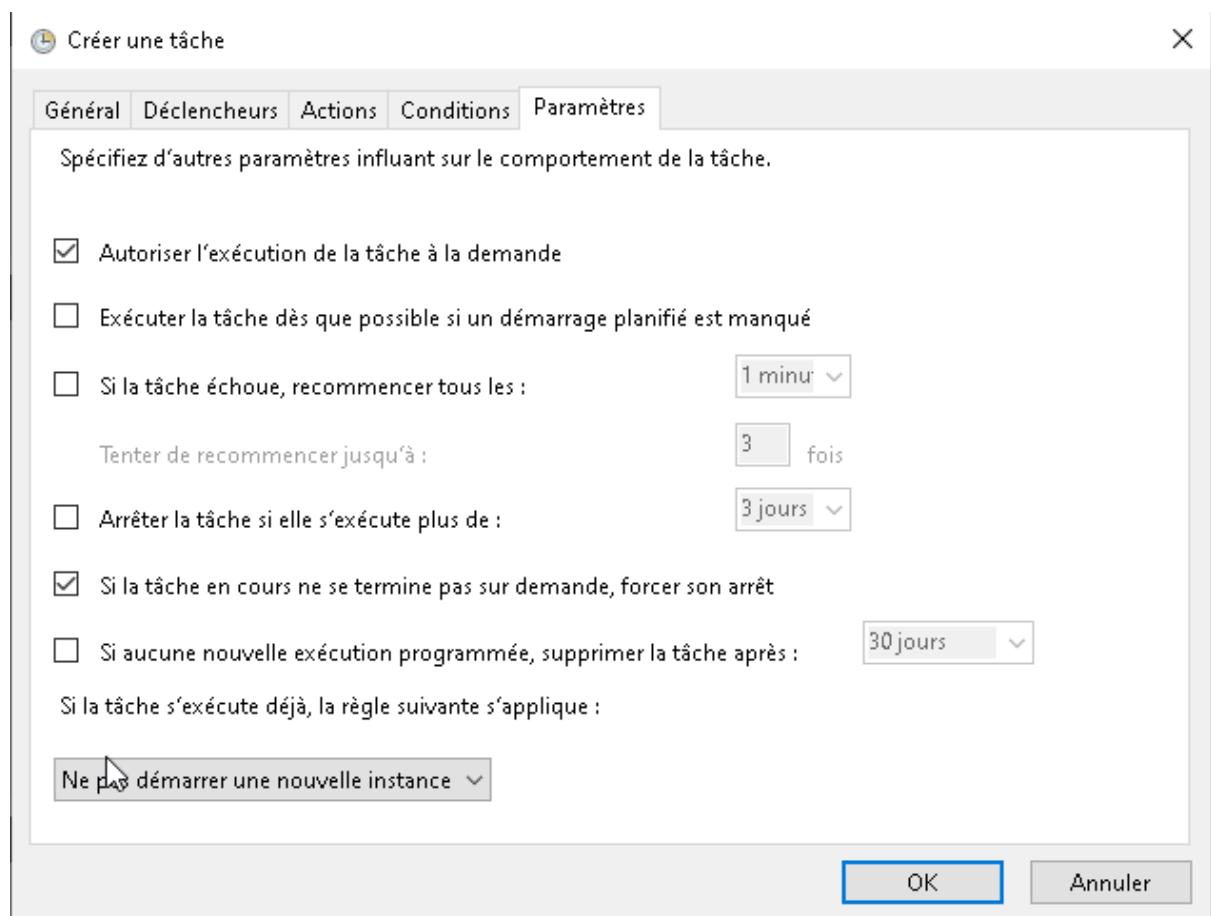
Les paramètres les plus importants sont :

- 30 jours sans connexion
- Lancement tous les dimanches









Nom	Statut	Déclencheurs
CreateExplor...	Prêt	Lors de la création ou de la modification de la tâche
MicrosoftEd...	Prêt	Plusieurs déclencheurs sont définis.
MicrosoftEd...	Prêt	À 13:38 tous les jours - Après le déclenchement, recommencer tous les 1 heure pendant 1 jour
Script cyber	Prêt	Plusieurs déclencheurs sont définis.
Script de net...	Prêt	Lorsque l'ordinateur est inactif

```

# Durée d'inactivité en jours
$delailnactivite = 30
$aujourdHui = Get-Date
$seuilDate = $aujourdHui.AddDays(-$delailnactivite)

# Comptes à exclure (ajoute d'autres si nécessaire)
$comptesExclus = @("Administrator", "Default", "Default User", "All Users",
"Public")

# Obtenir les profils utilisateurs
$profils = Get-WmiObject Win32_UserProfile | Where-Object {
    $_.LastUseTime -ne $null -and
    $_.Special -eq $false -and
    ($comptesExclus -notcontains ($_.LocalPath -split '\\')[-1]) -and
    ([datetime]::FromFileTime($_.LastUseTime) -lt $seuilDate)
}

foreach ($profil in $profils) {
    $cheminUtilisateur = $profil.LocalPath
    Write-Host "Utilisateur inactif détecté : $cheminUtilisateur"

    try {
        # Supprime uniquement le contenu du dossier utilisateur
        $contenu = Get-ChildItem -Path $cheminUtilisateur -Force -ErrorAction Stop
        foreach ($item in $contenu) {
            Remove-Item -Path $item.FullName -Recurse -Force -ErrorAction Stop
            Write-Host " → Supprimé : $($item.FullName)" -ForegroundColor Green
        }
    } catch {
        Write-Host "Erreur lors de la suppression du contenu de : $cheminUtilisateur"
        -ForegroundColor Red
        Write-Host $_.Exception.Message
    }
}

```