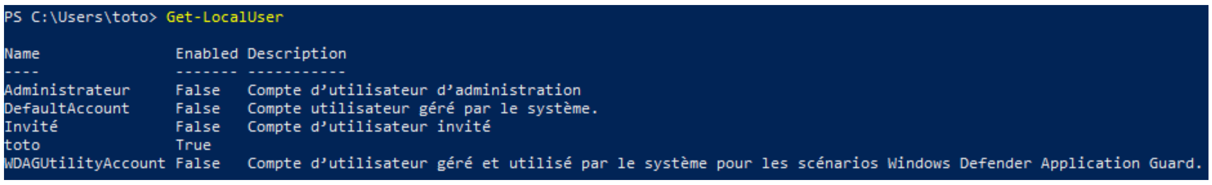
Basile LE THIEC
RT2-DevCloud

TP1 - R405

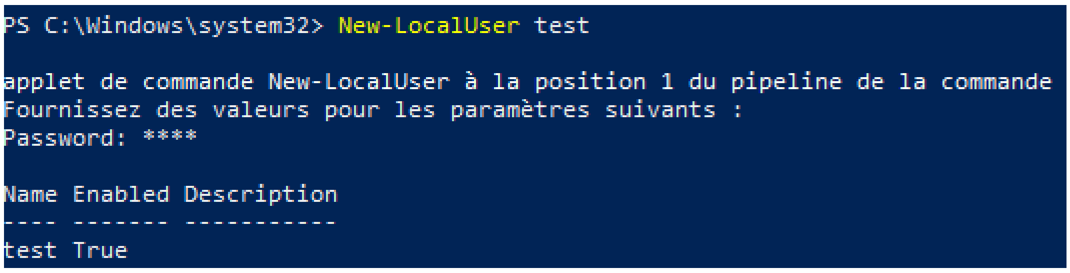
Exercice 1 :



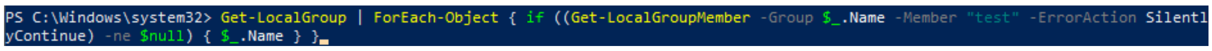
```
PS C:\Users\toto> Get-LocalUser

Name               Enabled Description
----               ------- -----------
Administrateur     False   Compte d'utilisateur d'administration
DefaultAccount     False   Compte utilisateur géré par le système.
Invité             False   Compte d'utilisateur invité
toto               True
WDAGUtilityAccount False   Compte d'utilisateur géré et utilisé par le système pour les scénarios Windows Defender Application Guard.
```

Les utilisateurs sont : **Administrateur**, **DefaultAccount**, **Invité**, **toto**.

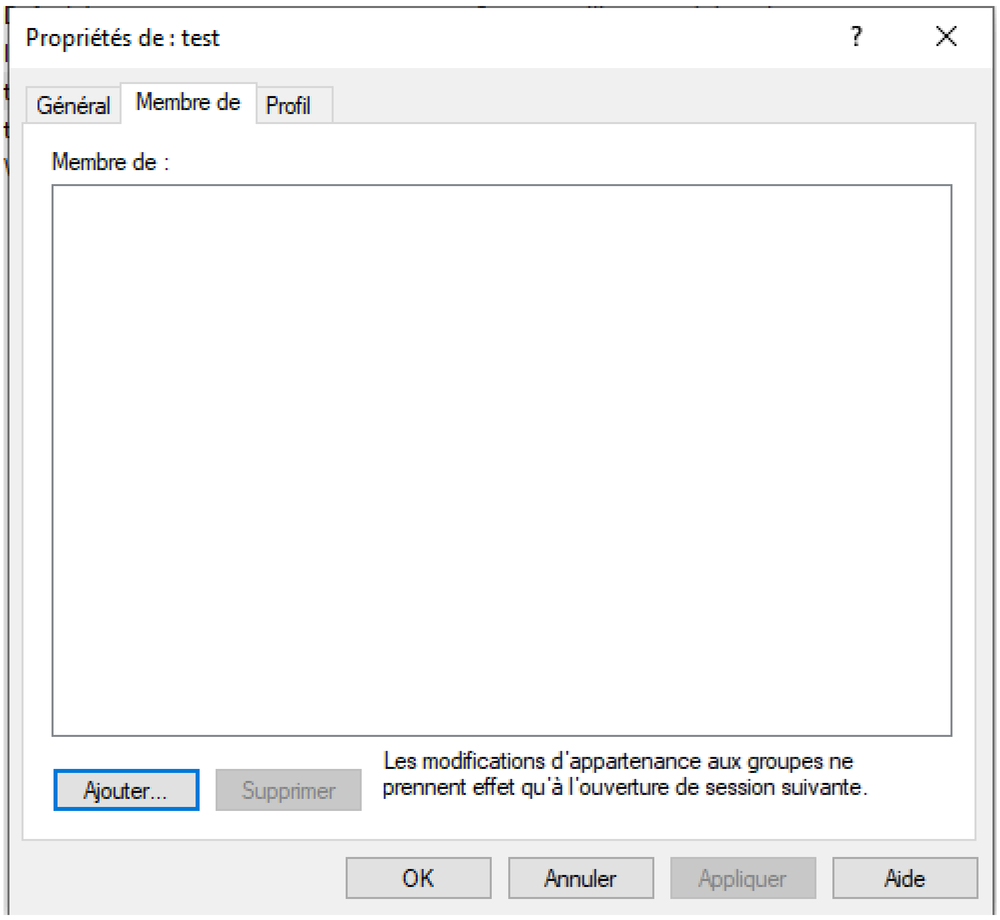
Exercice 2 :



```
PS C:\Windows\system32> New-LocalUser test

applet de commande New-LocalUser à la position 1 du pipeline de la commande
Fournissez des valeurs pour les paramètres suivants :
Password: ****

Name Enabled Description
---- ------- -----------
test True
```



```
PS C:\Windows\system32> Get-LocalGroup | ForEach-Object { if ((Get-LocalGroupMember -Group $_.Name -Member "test" -ErrorAction Silentl
yContinue) -ne $null) { $_.Name } }
```



Propriétés de : test

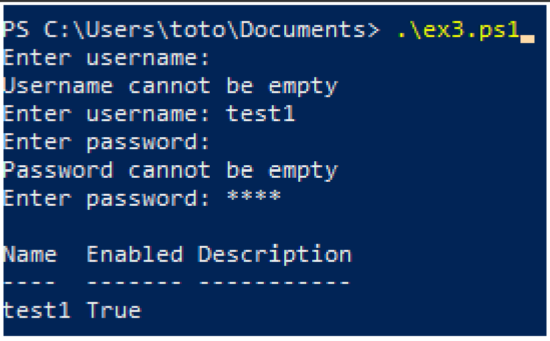
Membre de :

Les modifications d'appartenance aux groupes ne prennent effet qu'à l'ouverture de session suivante.

Le nouvel utilisateur n'est **dans aucun groupe**.

Exercice 3 :

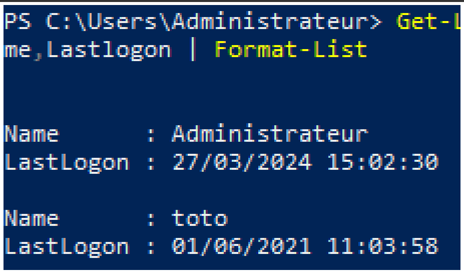
```
# Start a loop that will continue until a non-empty username is entered
do {
    # Prompt the user to enter a username
    $user = Read-Host -Prompt "Enter username"
    # If the username is empty
    if ($user -eq "") {
        # Output a message stating that the username cannot be empty
        Write-Host "Username cannot be empty"
    }
# Continue the loop as long as the username is empty
} while ($user -eq "")

# Start a loop that will continue until a non-empty password is entered
do {
    # Prompt the user to enter a password and store it as a secure
string
    $pass = Read-Host -Prompt "Enter password" -AsSecureString
    # Convert the secure string to a plain text string
    $plainPass =
[Runtime.InteropServices.Marshal]::PtrToStringAuto([Runtime.InteropServi
ces.Marshal]::SecureStringToBSTR($pass))
    # If the password is empty
    if ($plainPass -eq "") {
        # Output a message stating that the password cannot be empty
        Write-Host "Password cannot be empty"
    }
# Continue the loop as long as the password is empty
} while ($plainPass -eq "")

# Create a new local user with the entered username and password
New-LocalUser $user -Password $pass
# Add the new user to the "Utilisateurs" group
Add-LocalGroupMember -Group "Utilisateurs" -Member $user
```



```
PS C:\Users\toto\Documents> .\ex3.ps1
Enter username:
Username cannot be empty
Enter username: test1
Enter password:
Password cannot be empty
Enter password: ****

Name  Enabled Description
----  ------- -----------
test1 True
```

Exercice 4 :

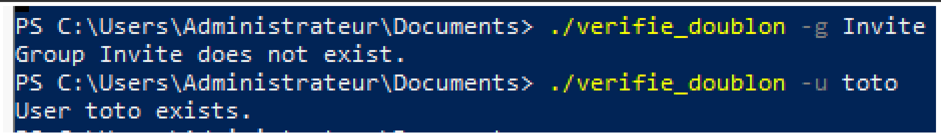
```
# Retrieve all local users
Get-LocalUser |
# Filter the users to only include those who have logged on within the last
3600 days
Where-Object {$_.Lastlogon -ge (Get-Date).AddDays(-3600)} |
# Select the name and last logon time of each user
Select-Object Name,Lastlogon |
# Format the results as a list
Format-List
```



```
PS C:\Users\Administrateur> Get-L
me,Lastlogon | Format-List


Name      : Administrateur
LastLogon : 27/03/2024 15:02:30

Name      : toto
LastLogon : 01/06/2021 11:03:58
```

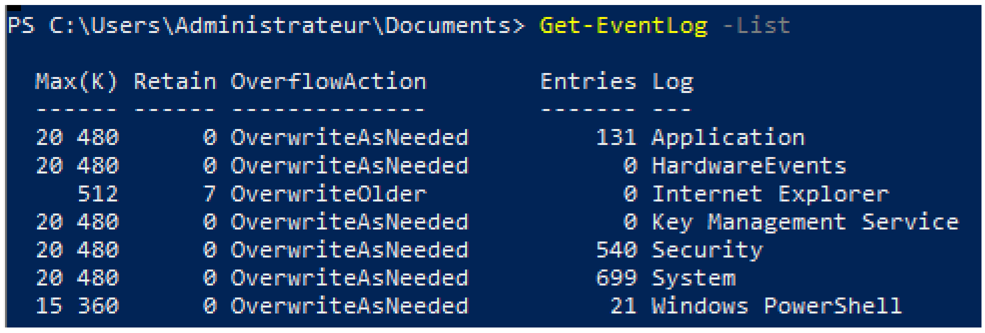
<u>Exercice 5 :</u>

```
# Define parameters for the script: a user name and a group name
param (
    # If the 'User' parameter set is used, the user name is mandatory
    [Parameter(ParameterSetName='User', Mandatory=$true)]
    [string]$u,

    # If the 'Group' parameter set is used, the group name is mandatory
    [Parameter(ParameterSetName='Group', Mandatory=$true)]
    [string]$g
)

# If a user name was provided
if ($u) {
    # Retrieve the local user with the specified name
    $user = Get-LocalUser | Where-Object { $_.Name -eq $u }
    # If the user exists
    if ($user) {
        # Output a message stating that the user exists
        Write-Host "User $u exists."
    } else {
        # Output a message stating that the user does not exist
        Write-Host "User $u does not exist."
    }



# If a group name was provided
} elseif ($g) {
    # Retrieve the local group with the specified name
    $group = Get-LocalGroup | Where-Object { $_.Name -eq $g }
    # If the group exists
    if ($group) {
        # Output a message stating that the group exists
        Write-Host "Group $g exists."
    } else {
        # Output a message stating that the group does not exist
        Write-Host "Group $g does not exist."
    }
}
```



```
PS C:\Users\Administrateur\Documents> ./verifie_doublon -g Invite
Group Invite does not exist.
PS C:\Users\Administrateur\Documents> ./verifie_doublon -u toto
User toto exists.
```

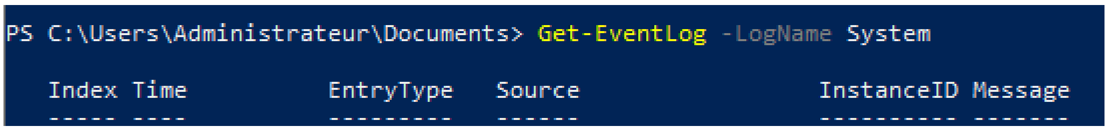
Exercice 6 :



```
PS C:\Users\Administrateur\Documents> Get-EventLog -List

  Max(K) Retain OverflowAction        Entries Log
  ------ ------ --------------        ------- ---
  20 480      0 OverwriteAsNeeded         131 Application
  20 480      0 OverwriteAsNeeded           0 HardwareEvents
     512      7 OverwriteOlder              0 Internet Explorer
  20 480      0 OverwriteAsNeeded           0 Key Management Service
  20 480      0 OverwriteAsNeeded         540 Security
  20 480      0 OverwriteAsNeeded         699 System
  15 360      0 OverwriteAsNeeded          21 Windows PowerShell
```

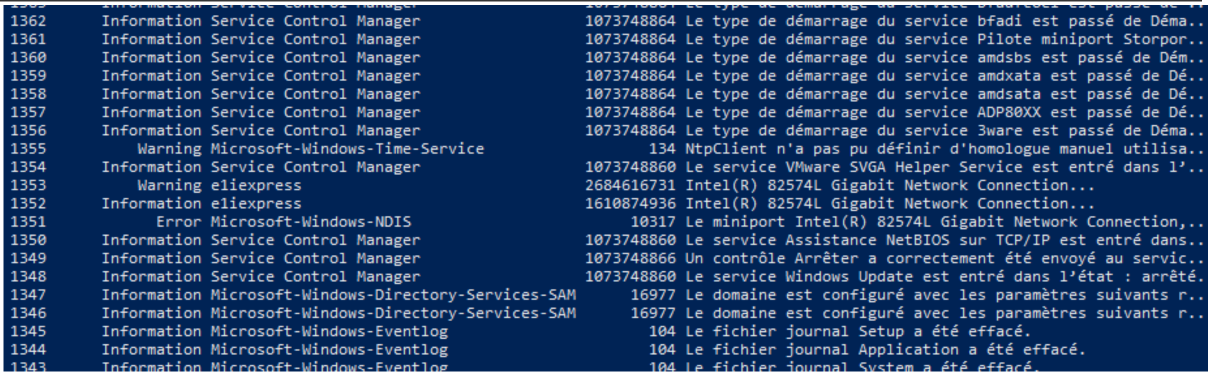
La commande *"**Get-EventLog -List**"* permet d'afficher la liste des logs disponibles.

Exercice 7 :



```
PS C:\Users\Administrateur\Documents> Get-EventLog -LogName System

   Index Time          EntryType   Source                 InstanceID Message
   ----- ----          ---------   ------                 ---------- -------
```

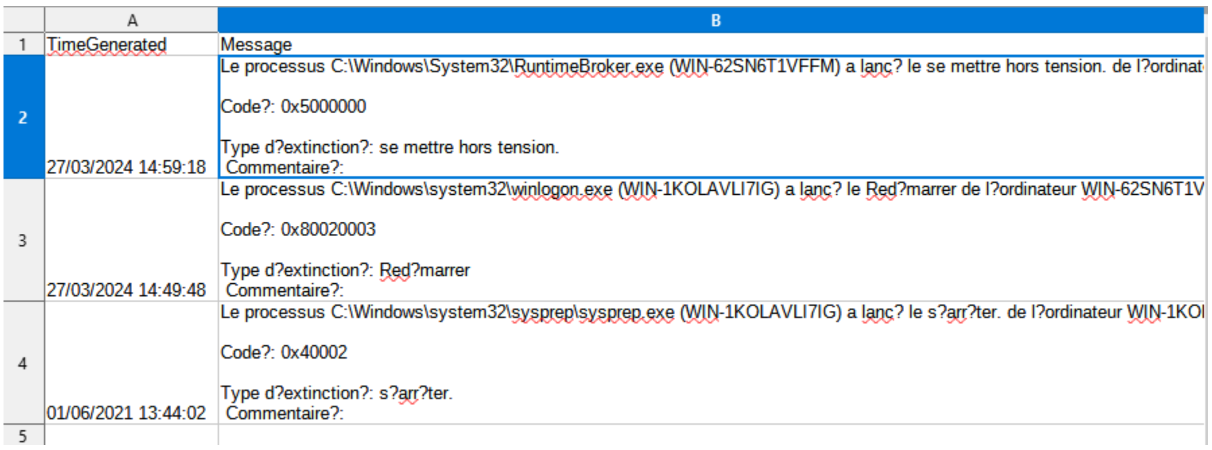
```
# Retrieve all entries from the System event log
Get-EventLog -LogName System |
# Select unique entries based on the Index, Time, EntryType, Source,
InstanceID, and Message properties
Select-Object Index, Time, EntryType, Source, InstanceID, Message
-Unique |
# Format the results as a table
Format-Table
```



```
1362       Information Service Control Manager                  1073748864 Le type de démarrage du service bfadi est passé de Déma..
1361       Information Service Control Manager                  1073748864 Le type de démarrage du service Pilote miniport Storpor..
1360       Information Service Control Manager                  1073748864 Le type de démarrage du service amdsbs est passé de Dém..
1359       Information Service Control Manager                  1073748864 Le type de démarrage du service amdxata est passé de Dé..
1358       Information Service Control Manager                  1073748864 Le type de démarrage du service amdsata est passé de Dé..
1357       Information Service Control Manager                  1073748864 Le type de démarrage du service ADP80XX est passé de Dé..
1356       Information Service Control Manager                  1073748864 Le type de démarrage du service 3ware est passé de Déma..
1355           Warning Microsoft-Windows-Time-Service                  134 NtpClient n'a pas pu définir d'homologue manuel utilisa..
1354       Information Service Control Manager                  1073748860 Le service VMware SVGA Helper Service est entré dans l'..
1353           Warning e1iexpress                               2684616731 Intel(R) 82574L Gigabit Network Connection...
1352       Information e1iexpress                               1610874936 Intel(R) 82574L Gigabit Network Connection...
1351             Error Microsoft-Windows-NDIS                        10317 Le miniport Intel(R) 82574L Gigabit Network Connection,..
1350       Information Service Control Manager                  1073748860 Le service Assistance NetBIOS sur TCP/IP est entré dans..
1349       Information Service Control Manager                  1073748866 Un contrôle Arrêter a correctement été envoyé au servic..
1348       Information Service Control Manager                  1073748860 Le service Windows Update est entré dans l'état : arrêté.
1347       Information Microsoft-Windows-Directory-Services-SAM      16977 Le domaine est configuré avec les paramètres suivants r..
1346       Information Microsoft-Windows-Directory-Services-SAM      16977 Le domaine est configuré avec les paramètres suivants r..
1345       Information Microsoft-Windows-Eventlog                      104 Le fichier journal Setup a été effacé.
1344       Information Microsoft-Windows-Eventlog                      104 Le fichier journal Application a été effacé.
1343       Information Microsoft-Windows-Eventlog                      104 Le fichier journal System a été effacé.
```

Exercice 8 :

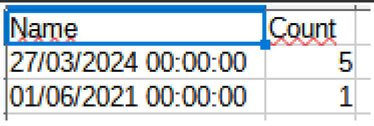
```
# Retrieve all entries from the System event log
Get-EventLog -LogName System |
# Filter the entries to only include those where the source is "user32" and
the event ID is 1074. The event ID 1074 corresponds to a system shutdown
Where-Object { $_.Source -eq "user32" -and $_.EventID -eq 1074 } |
# Select the time the event was generated and the message of the event
Select-Object TimeGenerated, Message |
# Export the results to a CSV file, without including type information in
the file
Export-CSV -Path shutdown.csv -NoTypeInformation
```



	A	B
1	TimeGenerated	Message
2	27/03/2024 14:59:18	Le processus C:\Windows\System32\RuntimeBroker.exe (WIN-62SN6T1VFFM) a lanc? le se mettre hors tension. de l?ordinat Code?: 0x5000000 Type d?extinction?: se mettre hors tension. Commentaire?:
3	27/03/2024 14:49:48	Le processus C:\Windows\system32\winlogon.exe (WIN-1KOLAVLI7IG) a lanc? le Red?marrer de l?ordinateur WIN-62SN6T1V Code?: 0x80020003 Type d?extinction?: Red?marrer Commentaire?:
4	01/06/2021 13:44:02	Le processus C:\Windows\system32\sysprep\sysprep.exe (WIN-1KOLAVLI7IG) a lanc? le s?arr?ter. de l?ordinateur WIN-1KOI Code?: 0x40002 Type d?extinction?: s?arr?ter. Commentaire?:
5		

Exercice 9 :

```
# Prompt the user to enter the number of most recent errors they want to
retrieve
$numberOfErrors = Read-Host -Prompt 'Enter the number of errors to retrieve'

# Get the specified number of most recent error entries from the System
event log
Get-EventLog -LogName System -EntryType Error -Newest $numberOfErrors |
# Select the date (without time) and the message of each error
Select-Object @{n='Date';e={$_.TimeGenerated.Date}}, Message |
# Group the errors by date
Group-Object -Property Date |
# Select the date and the count of errors for that date
Select-Object Name, Count |
# Export the results to a CSV file, without including type information in
the file
Export-CSV -Path errors.csv -NoTypeInformation
```



Name	Count
27/03/2024 00:00:00	5
01/06/2021 00:00:00	1