creating user accounts

# Objectives

* + Create user accounts using the GUI and PowerShell**.**
  + Add users to group using GUI and PowerShell.

## Skills Reviewed

* + Create Local accounts.
  + Log in locally and with Active Directory

## New Skills

* + Create local user and passwords with **New-LocalUser** command.
  + Add user to local group (Backup Operators and Event Log Readers).
  + Add users to local group with **Add-LocalGroupMember,** command.
  + Create domain accounts with **Active Directory Users and Computers.**
  + Create domain accounts with **New-AdUser** cmdlet.
  + Add member to Domain Admins group with **Add-ADGroupMember** cmdlet.
  + Use **Get-ADGroupMember** cmdlet to list member of group.
  + Use **Enable-ADAccount** to enable user accounts.
  + Modifying AD user accounts with **Set-AdUser** cmdlet.

# Initial Conditions

The following guided practices should be completed:

* + Guided Practice: Install and Upgrade Windows 10 and its prerequisite GP.

# Final Conditions

At the end of this exercise, you will have:

* + **TEMP-MS** user created on **CLIENT1** and added to **Backup Operators** and **Event Log Readers** local groups.
  + Two administrative accounts created in Active Directory and added to **Domain Admins** group. One based on student name and one for Master Chief.
  + Domain user accounts created for: Bob Greene, Jack Hunt, Jimmy Meeks, Ed Hart, Larry Bell, Jane King, Carol Ashworth, Ann Fellows, Betty Lou Craft, and Kate Neal.

# Instructions – Creating Local users

A Microsoft support employee requires administrative access to **CLIENT1** to troubleshoot an issue. In this step, you will create an account for the employee and configure the account with administrative rights.

1. Log onto **CLIENT1** with **ecpi** account.
2. Create the account as follows:
   1. Open **Computer Management** and browse to **Local Users and Groups** **>** **Users**.
   2. Right click **Users** and select **New User…**
   3. Create the new user with the following settings:
      1. **Username**: TEMP-MS
      2. **Password**: Password1
      3. **User must change password at next logon** is unchecked.
3. Add the account to the local **Administrators** group by
   1. Double-click the **TEMP-MS** user
   2. Click the **Member Of** tab. Click **Add**.
      1. The **Select Groups** dial opens. Click **Advanced** and then click **Find Now**. This lists all groups available on the local machine.
      2. Select **Administrators** and click **OK**.
      3. On the **Select Groups** dialog, click **OK**.
   3. Notice that **Administrators** is in the **Member of:** box. Click **OK**.

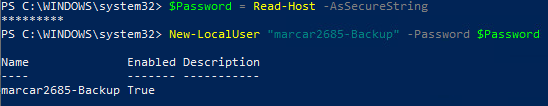
# Instructions – Creating Local Groups and adding Users

A contractor from **InfoTech** requires rights to your server to perform a backup of the system. In this task you will create a user account for the contractor and give the account the rights to perform a backup of the system. Create and configure the rights of the account within PowerShell.

1. On **CLIENT1**, open **Windows** **PowerShell (Admin)** rights.
   1. Review the **New-LocalUser** command syntax by using **Get-Help**. You will need run **Update-Help** when prompted.
   2. Use **New-LocalUser** to add ***studentid*-Backup** (where ***studentid*** is your studentID) with a password of **Password1**. This process requires two steps. Step one is to store a password as a secure string in a variable, and step two is to create the account. Notice how the ***$Password*** variable is created and then passed to second command. You will need to type **Password1** after issuing the first command. **Note**: Variables are case-sensitive.

$Password = Read-Host -AsSecureString

New-LocalUser “marcar2685-Backup” -Password $Password



* 1. Use the **Get-LocalUser** cmdlet to verify the account creation.
  2. Review the **Add-LocalGroupMember** command syntax by using **Get-help**.
  3. Use **Add-LocalGroupMember** to configure the account to have the rights to backup **CLIENT1** by adding the account to the **Backup Operators** local group.

Add-LocalGroupMember -Group “Backup Operators” -Member “marcar2685-Backup”

Why did **Backup Operators** require quotes around the name?

* 1. Verify the user was added to the **Backup Operators** local group. What PowerShell command did you use?
  2. This same contractor from **InfoTech** now requires access to read event logs so that he can troubleshoot why backups are failing. Use PowerShell to add this user to the appropriate group.

# Instructions – Creating users in Active Directory

Your supervisor would like you to create a user account for each of your employees. First, you must create a user account for yourself and give your account administrative rights to the entire domain. You will then use this administrative account to add the remaining user accounts.

1. Log into **SERVER** using **Administrator** account.
2. In **Server Manager,** browse to the **Tools🡪Active Directory Users and Computers**
3. Expand the **Domain** node, right-click the **Users** folder and select **New🡪User**.
   1. In the **New Object – User** dialog box, fill in the following information:
      1. **First name**: Your first name
      2. **Last name**: Your last name
      3. **Full name**: should be filled in automatically
      4. **User logon name**: first name initial + last name (e.g. mcarruth)
      5. **User logon name (pre-Windows2000):** should be filled in automatically
      6. Click **Next**.
      7. Type **Password1** in the **Password:** and **Confirm password:** dialog box.
      8. **Uncheck** the **User must change password at next logon** checkbox, check the **Password never expires** checkbox, and then click **Next**.
      9. Click **Finish** to create the user.
   2. Give administrative rights to your user by adding to **Domain Admins** global group.
      1. Double click on your new user account.
      2. Select the **Member Of** tab.
      3. Click the **Add** button.
      4. Click the **Advanced** button andclick **Find Now**, this will display all the groups in your domain.
      5. Scroll down, select **Domain Admins,** and click **OK**.
      6. Click **OK**. You should now see two groups, **Domain Users** and **Domain Admins** in the **Member Of** tab. Click **Ok**.

You now have administrative access to the entire domain.

* 1. Login to **SERVER** with your new administrative account and create the account for your corporation listed in the table below in the **Users** container:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **First Name** | **Last Name** | **Full Name** | **Logon Name** | **Logon Name (Pre-W2K)** | **Password** |
| Bob | Greene | Bob Greene | bgreene | bgreene | Password1 |
| Jack | Hunt | Jack Hunt | jhunt | jhunt | Password1 |
| Jimmy | Meeks | Jimmy Meeks | jmeeks | jmeeks | Password1 |
| Ed | Hart | Ed Hart | ehart | ehart | Password1 |
| Larry | Bell | Larry Bell | lbell | lbell | Passowrd1 |

Your supervisor (Master Chief) has requested you to create an administrative account in your domain that has administrative rights using PowerShell.

1. Use the following procedure to create a user account using PowerShell. On **SERVER**, open **PowerShell** with administrative rights.
   1. Use **Get-Help** to review the syntax for the **New-ADUser** command.
   2. Create a new user with the following settings:
      1. **First name**: Master
      2. **Last name**: Chief
      3. **Full name**: Master Chief
      4. **User logon name**: mchief
      5. **User logon name (pre-Windows2000)**: mchief
      6. **Display Name:** Master Chief
      7. **Name:** Master Chief
      8. **Password:** Password1
      9. Use the following parameters with the **New-ADUser** command
         1. –Name
         2. –GivenName
         3. –Surname
         4. –UserPrincipalName
         5. –SamAccountName
         6. –DisplayName

New-AdUser -Name “Maser Chief” -GivenName Master -Surname Chief -UserPrincipalName mchief@kmk.local -SamAccountName mchief -DisplayName “Master Chief”

* 1. Note that the **New-AdUser** cmdlet does not have the ability to set a password. You must use the **Set-AdAccountPassword** cmdlet. Use **Get-Help** to view help for this cmdlet. Note that the **–NewPassword** parameter required a **SecureString** as a parameter. When creating local accounts above, you used a two-step process for PowerShell. In this procedure you will use a single power shell command. Set the password for **Master Chief** to **Password1**. Note that everything inside the parenthesis ( ) is a complete PowerShell command. This command executes and its output object is used for the **-NewPassword** parameter.

Set-ADAccountPassword -Identity “mchief” -Reset -NewPassword (ConvertTo-SecureString -AsPlainText "Password1" -Force)

* 1. The user is account is created and the password is set. Review the **Get-ADGroupMember** and **Add-ADGroupMember** cmdlet help. Add the **Master Chief** account to the **Domain Admins** group.

Example command with different user:

Add-ADGroupMember -identity "Domain Admins" -members ecpi

* 1. Enable the user account using the **Enable-ADAccount** cmdlet. Example command with different user account:

Enable-ADAccount bcarruth

1. Create the user below using **PowerShell** cmdlets. Do not add the users to the **Domain Admins** group. Ensure all user accounts are enabled as you will be using the account in later exercises. **Hint**: Use up arrow in PowerShell.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **First Name** | **Last Name** | **Full Name** | **Logon Name** | **Logon Name (Pre-W2k)** | **Password** |
| Jane | King | Jane King | jking | jking | Password1 |
| Carol | Ashworth | Carol Ashworth | cashworth | cashworth | Password1 |
| Ann | Fellows | Ann Fellows | afellows | afellows | Password1 |
| Betty Lou | Craft | Betty Lou Craft | bcraft | bcraft | Password1 |
| Kate | Neal | Kate Neal | kneal | kneal | Password1 |

**Note:** You have created several users in the guided practice and assigned passwords. Do not change these passwords. This may cause some of the grading scripts to not function.

# Document work

1. **Download** the **grading** **script** from the assignment page to the **C:\Scripts** folder.
2. Check your lab by running the following command:

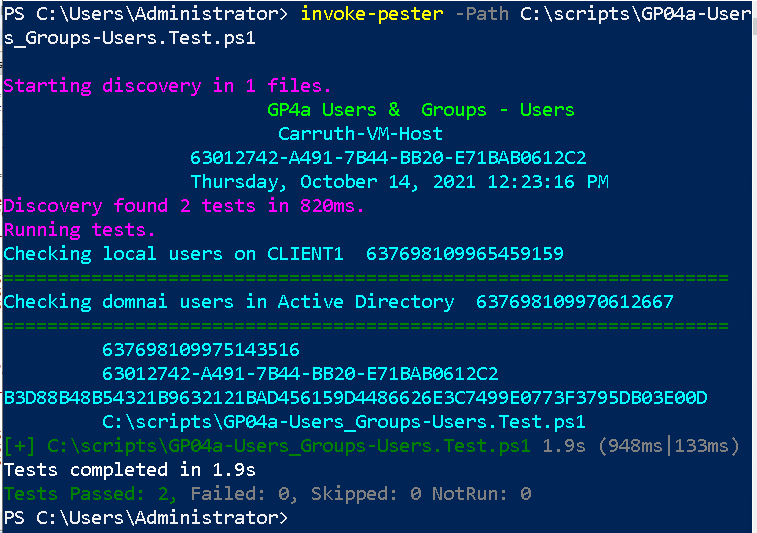
Invoke-Pester -Path C:\Scripts\GP04a-Users\_Groups-Users.test.ps1

**Note**: You will see a security warning when running the script. Enter **R** to run the script.

If you want to see more detail, add **-Output Detailed** to the command. This may assist you with troubleshooting

Invoke-Pester -Path C:\Scripts\GP04a-Users\_Groups-Users.test.ps1 -Output Detailed

1. You should not see any red in the output. Red in the PowerShell way of telling you that an error condition exists. Most of the time, the output will tell you what is wrong. If it is not obvious, contact your teacher and ask for assistance. You will be learning PowerShell during this term. **Correct** any **errors** you may have and run the script until all the output has no red. You should see the output like the images below.



1. Capture a snippet that shows the PowerShell Command and all its output. If you must use more than one snippet to capture the output, you must have at least **one line of overlap** in the snippets. The text in the snippets **must be legible** when pasted into the Word document. Paste the snippet(s) into the **Users&Groups\_*Firstname*\_*LastName.docx* (**where***Firstname***is your first name and***Lastname***is your last name**)**. The snippet should look like the image above.

Groups

# Objectives

* + You will create and manage groups using GUI and command line. You will implement AGGDLP grouping strategy**.**

## New Skills

* + Open **Active Directory Users and Computers** with **dsa.msc**
  + Create groups with **Active Directory Users and Computers**
  + Create group using **New-ADGroup** cmdlet.
  + Add group members with the **Add-ADGroupMembers** cmdlet.
  + Verify group membership with **Get-ADGroupMembers** cmdlet.

# Initial Conditions

The following guided practices should be completed:

* + Creating User Accounts section above is complete and the grading script has no errors.
  + **CLIENT1**, **CLIENT2**, and **SERVER** VMs are running.

# Final Conditions

At the end of this exercise, you will have:

* + Global security groups added: **Services**, **Employees**, **GRVL-Sales**, **GRVL-Services**, **GRVL-Projects**, **RALE-Sales**, **RALE-Services**, and **RALE-Projects**.

# Instructions

**AGDLP** strategy – **A**ccount 🡪 **G**lobal Groups; **G**lobal Groups 🡪 **D**omain **L**ocal Groups; **D**omain **L**ocal groups assigned **P**ermissions. The first step in the process is creating the Global Groups needed.

1. Create Global Groups.
   * 1. Logon to **SERVER** with your administrative account. Enter **dsa.msc** in **Run** dialog box to open **Active Directory Users and Computers**.
     2. In **Active Directory Users and Computers,** right-click **Users** container and select **New > Group**
        1. Type **Projects** in the **Group Name:** box. Notice **the Group name (pre-Windows 2000):** box. Verify that the **Group scope** is **Global** and the **Group type** is **Security** and then click **OK**.
     3. On **SERVER**, open **PowerShell** with administrative rights.
        1. Create the **Sales** group using the **New-ADGroup** cmdlet. The command will be similar to

New-ADGroup -Name Sales -GroupScope Global -GroupCategory Security -Path "CN=Users,DC=kmk,DC=local".

* + 1. Verify the group creation using the **Get-AdGroup** cmdlet. **Projects** and **Sales** should be in the list.

Get-AdGroup -Filter \*

Get-AdGroup Sales  
Get-AdGroup Projects

* + 1. Using PowerShell, create additional global security groups – **Services**, **Employees**, **GRVL-Sales**, **GRVL-Services**, **GRVL-Projects**, **RALE-Sales**, **RALE-Services**, and **RALE-Projects**.

1. Add users to Global Group.
   * 1. On **SERVER**, open **Active Directory Users and Computers.**
        1. Select **Jane King** and **Bob Greene** accounts, right click and select **Add to a group...**
           1. In the **Enter the object names to select** box, enter **GRVL-Sales** and click **Check Names**. You expect the system to respond by underlining **GRVL-Sales**. This indicates that the name was found. Click **OK**.



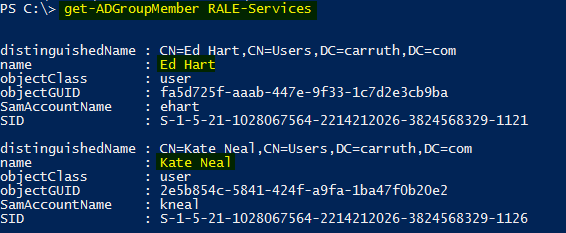
* + - 1. Use the **Add-ADGroupMember** cmdlet to add **Carol Ashworth** and **Jack Hunt** to the GRVL-Services group.

Add-AdGroupmember GRVL-Services -Members cashworth,jhunt

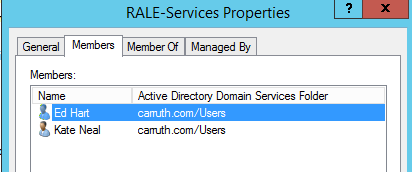
* + - 1. Using PowerShell, complete the user addition to the remaining groups.

|  |  |
| --- | --- |
| **Group Name** | **Members** |
| **GRVL-Projects** | Ann Fellows, Jane King |
| **RALE-Sales** | Betty Lou Craft, Jimmy Meeks |
| **RALE-Services** | Kate Neal, Ed Hart |
| **RALE-Projects** | Larry Bell, Master Chief |

* + - 1. Verify users have been added to the **RALE-Services** group with the **Get-AdGroupMember** cmdlet.



* + - 1. You can also verify group membership by opening the properties page of the group in **Active Directory Users and Computers**.



* + - 1. Verify that each group has the appropriate group members.

1. Nesting Global Groups. Group nesting is the process of adding groups to other groups. It’s done exactly the same way as adding users to groups.
   * 1. Add the **Projects**, **Services**, and **Sales** groups to the **Employees** group.
     2. Add the **RALE-Projects** and **GRVL-Projects** groups to the **Projects** group.
     3. Add the **RALE-Sales** and **GRVL-Sales** groups to the **Sales** group.
     4. Add the **RALE-Services** and **GRVL-Services** groups to the **Services** group.
     5. Verify the groups were added using the **Get-AdGroupMember** cmdlet.

Example command for one group

Get-AdGroupMember Sales

1. Creating Domain Local Groups. Use Domain local groups to assign permissions to resources. Create the following domain local groups using the information provided. Example PowerShell command:

New-ADGroup -Name Sales\_Modify -GroupScope DomainLocal -Description “Users in this group have Modify permissions to the Sales shared folder on SERVER” -GroupCategory Security -Path “CN=Users,DC=kmk,DC=local”

* + 1. **HR\_Read**
       1. **Description** - Users in this group have read permissions to the **HR** shared folder on **SERVER**
    2. **Corp\_Printer\_Print**
       1. **Description** - Users in this group have print permissions on the **Corp\_Printer** shared on **SERVER**
    3. **Projects\_Modify**
       1. **Description** - Users in this group have Modify permissions to the **Projects** shared folder on **SERVER**
    4. **Services\_Modify**
       1. **Description** - Users in this group have Modify permissions to the **Services** shared folder on **SERVER**
    5. **Sales\_Modify**
       1. **Description** - Users in this group have Modify permissions to the **Sales** shared folder on **SERVER**

Group creation is completed.

# Document Work

1. **Download** the **grading** **script** from the assignment page to the **C:\Scripts** folder.
2. Check your lab by running the following command:

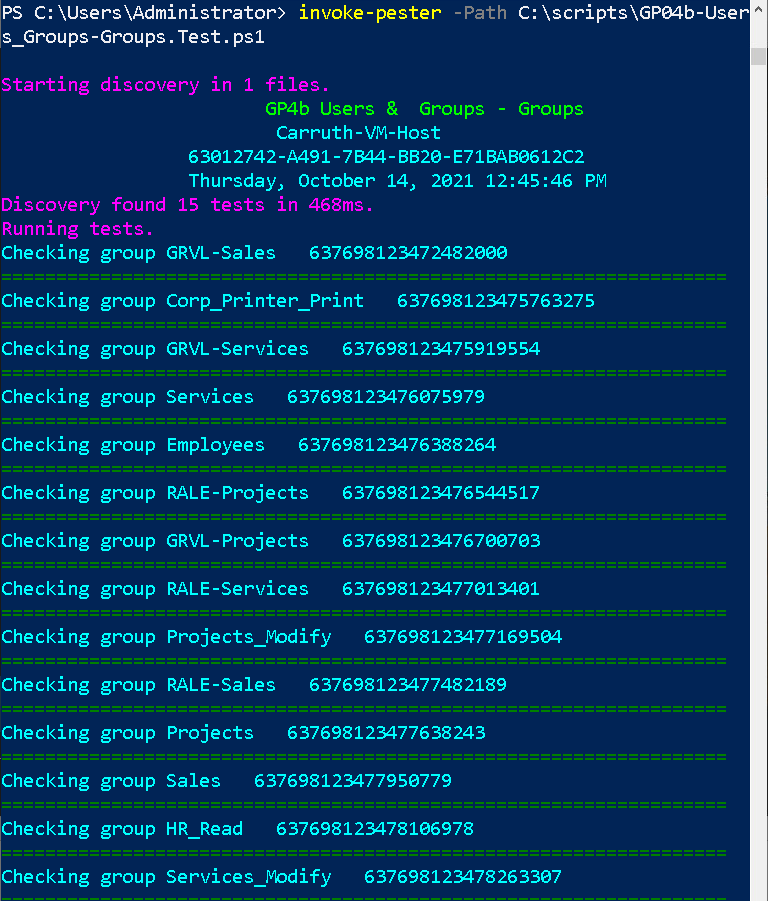
Invoke-Pester -Path C:\Scripts\GP04b-Users\_Groups-Groups.test.ps1

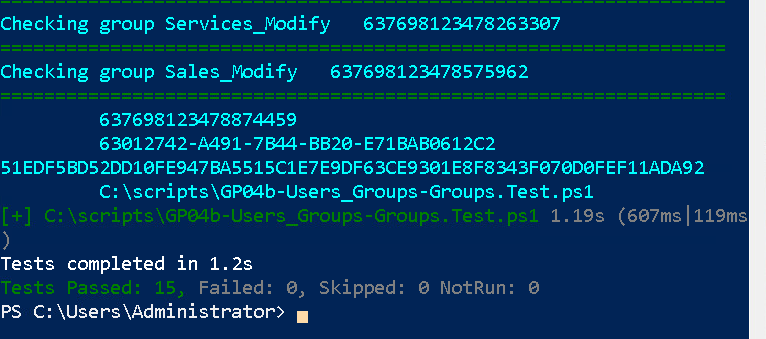
**Note**: You will see a security warning when running the script. Enter **R** to run the script.

If you want to see more detail, add **-Output Detailed** to the command. This may assist you with troubleshooting

Invoke-Pester -Path C:\Scripts\GP10-Creating\_DNS\_Resource\_ Records.test.ps1 -Output Detailed

1. You should not see any red in the output. Red in the PowerShell way of telling you that an error condition exists. Most of the time, the output will tell you what is wrong. If it is not obvious, contact your teacher and ask for assistance. You will be learning PowerShell during this term. **Correct** any **errors** you may have and run the script until all the output has no red. You should see the output like the images below.





1. Capture a snippet that shows the PowerShell Command and all its output. If you must use more than one snippet to capture the output, you must have at least **one line of overlap** in the snippets. The text in the snippets **must be legible** when pasted into the Word document. Paste the snippet(s) into the **Users&Groups\_*Firstname*\_*LastName.docx* (**where***Firstname***is your first name and***Lastname***is your last name**)**. The snippet should look like the image above.
2. **Upload** the **Users&Groups\_*Firstname*\_*LastName.docx*** documentin the submission area of the assignment.