KIT304 Server Administration and Security Assurance

Tutorial 7

Goal

In your previous tutorial you set up the Active Directory Domain Controller and then created some users and groups using the GUI (graphical user interface). Today you will build on that experience by completing similar tasks using the **PowerShell** command-line interface.

Introduction

During the first module you spent most of your time using the command-line. In the previous Windows tutorial you briefly saw the PowerShell command-line, but spent the bulk of the tutorial working within the GUI portion of Windows Server. Today you will spend the bulk of your time with the command-line interface.

Windows Server has historically been known to be very “graphically-based”, part of the reason being the failed attempt to replace the DOS command line interface with Cscript in the late 1990s. PowerShell was released in 2006 and we are now up to version 5.1 in Server 2016. Windows PowerShell offers various commands (which Microsoft refers to as *cmdlets*) that you can use to perform administrative tasks. Windows also gives you the ability to create Windows PowerShell scripts by adding individual cmdlets to text files and saving the files with the *.ps1* extension.

To highlight the shift away from GUI driven administration in Windows Server, prior to the release of PowerShell, it was effectively 100% GUI driven. By the time Windows Server 2008 was released, there were 230 cmdlets for undertaking administration tasks. There are over 2,400 cmdlets in Server 2012, and another 500 in Server 2016. Common applications that are used with Windows Server such as Exchange, Azure and Office 365 are all configured using the PowerShell.

In 2018 PowerShell Core 6.0 was released. This is different to the version of PowerShell (5.1) that you will be using in this unit, and it actually has a different target audience. Version 5.1 is considered a complete product, and 6.0 is a new product targeting cross-platform environments (Windows, macOS, and Linux), open-source, and hybrid cloud. It has significantly different capabilities to version 5.1.

Windows Administration goes hand in hand with a command-line interface, just like in Unix. Hopefully this module gives you a good appreciation of the differences in the syntax and approach to your Unix experience.

Activity

1. This time around, you're going to launch **Windows 2016** using its default snapshot, which has Active Directory already set up. All you should need to do is double-click the Win2016 server entry in the VMWare Fusion virtual machine list.

   Active Directory in this snapshot is set up as you would have configured it at the end of the previous tutorial. Thus, there will be a pre-configured Domain Controller for a domain called **networks.local**. (We could have chosen any domain at setup, but we chose "networks" as that's the common theme of this unit, and "local" implying that this domain is not on the internet). The Administrator of the server is called **Administrator** and the password is **ToorToor1**.

   Note that the configured version of Windows Server takes longer to boot and be fully ready for use than the unconfigured version. After you've logged in, it's best to wait a minute or so for the various services to start before trying to do anything that is resource intensive.

2. Start up **Windows 7**, so that it loads in the background while you do the next activity. (Our Windows 7 install also includes PowerShell, but this is an older version – 2.0).

3. You should spend the first portion of the tutorial familiarising yourself with **PowerShell**. To do that, work through some of the sections of the Windows PowerShell Fundamentals at this link:

   https://docs.microsoft.com/en-us/powershell/scripting/learn/understanding-important-powershell-concepts?view=powershell-5.1

   The most useful subsections to read are:

   - Learning PowerShell (the page at the link above)
   - Learning PowerShell names
   - Using familiar command names
   - Getting detailed help information

   Since you have Windows Server up and running, be sure to try some of the commands in PowerShell as you read about them. Becoming confident with the Windows equivalent of **man** and the ability to lookup lists of commands is an important skill.

   If you can't recall how to access PowerShell, click the magnifying glass at the lower left of the screen, and enter the text **power**. You'll be presented with several PowerShell variants. For basic command access, you can just select the Desktop app, but for more advanced use you may want to consider using the PowerShell ISE (Integrated Scripting Environment).

> ➢ Note: If you find the text in the PowerShell window too small, right-click in the window's title bar and select **Properties**. Change the text size in the **Font** tab.

4. After you've completed reading the introductory material on PowerShell, you can set the Windows Server IP address using it – specifically, with commands `Get-NetAdapter` and `New-NetIPAddress`.

   Select an IP address for your server in the **192.168.1.*X*** range, and apply it to the server like this:

   ```
   Get-NetAdapter | New-NetIPAddress -IPAddress ipAddress -Prefixlength 24
   ```

5. Configure the Windows 7 virtual machine with an IP address in the same subnet. While doing this, set the **Gateway** and **DNS** values to the IP address of the Windows 2016 Server. Confirm that you have network connectivity between the two VMs by pinging between them.

6. Now add the **Windows 7** client to the domain being managed by the Windows 2016 server:

   i. On the Windows 7 VM, click the Windows "Start" button (at the bottom left corner of the screen), in the menu that is displayed, right-click on **Computer**, and then select **Properties**.
   ii. Under **Computer name, domain, and workgroup settings**, click **Change settings**.
   iii. In the **Computer Name** tab, click the **Change** button. Under **Member of**, click **Domain** and then type the name of the domain (**networks.local**) and click **OK**.
   iv. When prompted for an administrator's credentials, use **Administrator** and **ToorToor1**. After a short pause, you'll be welcomed to the domain, and you'll see a message telling you to restart. Close the Properties box, and restart.

7. Next, on the 2016 Server, you're going to create a group and some users from the **PowerShell** command-line. You can read extensive documentation from Microsoft on managing user accounts and groups at the following URL:

   https://technet.microsoft.com/en-us/library/dd378937(v=ws.10).aspx

   These documents cover these topics in much more depth and detail than you'll need today, but if you're going to administer a Windows server, this documentation is well worth reading.

   Before you can create users and groups, you need to import the Active Directory cmdlets. To do this enter the command `Import-Module ActiveDirectory`.

8. To create a group, use the **`New-ADGroup`** command. You can enter it with command line arguments, but if you enter the command by itself it will prompt you to enter the minimum values needed (as do most PowerShell commands). Try both ways to create several groups for different groups (eg., marketing, research and accounting). Here's how you would enter the command and specify all parameters:

   ```
   New-ADGroup -name groupName -groupscope Global
   ```

9. To display a list of groups, use the `Get-ADGroup` command. This command requires a `-Filter` parameter to limit the groups listed to just the subset that you're interested in. To see a list of all groups, enter the command like this:

   ```
   Get-ADGroup -Filter *
   ```

   To use filters meaningfully, you need to specify which attribute of the group you want to filter by. For example, to get a list of groups whose name starts with the letter **m**, you would need to specify the filter like this:

   ```
   Get-ADGroup -Filter {Name -like "m*"}
   ```

   or, to get a list of the groups whose **GroupScope** parameter was set to **Global**, you would specify the filter like this:

   ```
   Get-ADGroup -Filter {GroupScope -eq "Global"}
   ```

   There are many other filter variants. To see some examples, enter the command:

   ```
   help Get-ADGroup -examples
   ```

   (This command will fail on our 2016 viraulaRemember too that you can look up groups in the **Active Directory Users and Computers** tool launched from the **Server Manager**.

10. Before you create some users, you should create an *Organisational Unit* to store them in. An organizational unit (OU) is a subdivision of a domain into which you can put users, groups, computers, and other organizational units. They often parallel an organization's internal structure.

    ➢ If you're wondering what the difference is between groups and OUs, groups are used to control a member's access to resources (such as shared files and folders), whereas OUs are used to define who has administrative control over users, groups, and computers.

    A user can be a member of many groups, but they can only exist within a single OU.

    Select a name for your OU, and write it here. You'll use this multiple times when entering commands later in the tutorial:

    ____

11. To create your OU, enter the following command:

```
New-ADOrganizationalUnit -Name name -Path "dc=networks,dc=local"
```

Note in this command you also needed to enter the domain name (**`networks.local`**) as a series of *Domain Components* (eg., **`dc=network`**).

12. Next, you're going to create some users with the **`New-ADUser`** command. First, use the **`help New-ADUser`** command to see how many options exist for it – you will see that there are many optional parameters.

    The following commands are a few variants to try. After you enter each one, look up the result in either the Server Manager's **Active Directory Users and Computers** tool, or use **Powershell**'s **`Get-ADUser`** *name* command. Note down the differences between them, or anything of note about each one. (Note that each of these commands should be entered on a single line, even though they are listed here on multiple lines).

    ```
    New-ADUser Mary
    ```

    ____

    ```
    New-ADUser -Path "ou=yourOU,dc=networks,dc=local" -Name bob
    -UserPrincipalName bob@networks.local
    ```

    ____

    ```
    New-ADUser -Path "ou= yourOU,dc=networks,dc=local" -Name Jane -
    UserPrincipalName jane@networks.local -Enabled 1
    ```

    ____

    When trying this next command, you need to provide a password that matches certain complexity requirements. Don't include the user's account name in the password, and be sure to use a mix of lower and upper case letters and digits.

    ```
    New-ADUser -Path "ou= yourOU,dc=networks,dc=local" -Name Jane
    -UserPrincipalName jane@networks.local -AccountPassword
    (ConvertTo-SecureString -AsPlainText "password" -Force) -Enabled 1
    ```

    ____

    Note – if you got a password-related error in the last command, you should note that the system still created the account, but it left it disabled. You can delete the account with this command:

    ```
    Remove-ADUser Jane
    ```

    and then try again with a better password.

13. On the **Windows 7** computer, attempt to login using Jane's credentials. If it doesn't work, ask your tutor for help.

14. Create a new account with this command:

```
New-ADUser -Path "ou=yourOU,dc=networks,dc=local" -Name Bill
-UserPrincipalName bill@networks.local -LogonWorkstations bill
-AccountPassword (ConvertTo-SecureString -AsPlainText "password" -Force)
-Enabled 1
```

On Windows 7, log off (as Jane), and then log in as the new user Bill. What happened?

____

15. When you're using complex commands like `New-ADUser` for which there are many different optional parameters, you can use a simple graphical tool from within **PowerShell** to generate the full command. The way to invoke this is by entering the command `Show-Command`. Try that now, and in the window appears, enter the text `New-ADU` in the **Name:** field at the top. The list of commands should be filtered down as you enter each character.

    Click on the **New-ADUser** command from the list, and you'll be presented with a list of parameters that the command accepts.

    You may find it easier to use this interface to provide all of the parameters you have been entering on the command line. Once you have the parameters set the way you want, click the **Run** button, and the commands are entered onto the PowerShell command line for you. You can run this command like any other.

Let's take stock of where you're at: You now have a group, an organizational unit, and a number of users. In the next series of steps, you'll create a shared folder that is owned by the group, and you'll then add the users to that group, thus granting them access to the share.

16. To add members to a group, use the command `Add-ADGroupMember`. Add three of your users to the group you created back at step 8. What command did you use?

____

17. You can list the members in a group with the command `Get-ADGroupMember` *groupname* – try that now to verify that only the users you added are in the group.

18. Next, create a subdirectory (i.e., a folder) at the root of the C: drive using either the `mkdir` or `New-Item` commands. This is the folder that you will share between members of your group.

> ➢ Note: in the Unix module, you created subdirectories for group work, and then shared them among groups of users on the same system. You can do the same thing on Windows Server, but since it includes a network file sharing service, you can also share those directories to users on other machines that are connect to the domain.

19. To make your directory available over the network, you need to create a *share* for it. You do that with the **`New-SMBShare`** share command, which requires a share name (**`-Name`**) and the path to the directory being shared (**`-Path`**). The share name is how the share is presented to users browsing the network, and is typically the same as the name of the directory being shared (but does not have to be). You can optionally grant **`-ReadAccess`** or **`-FullAccess`** to groups and/or users.

    Create a new SMB share for the folder you created in the previous step, and grant full access to one of the groups you created earlier. What is the full command you used to do this?

    ____

20. The various SMBShare commands and documentation can be found at this link:

    https://docs.microsoft.com/en-us/powershell/module/smbshare

    Visit the link and read through the list of SMB-related commands in order to answer the following questions (be sure to try these on your server while you work).

21. Which command lists all of the shares on the server?

    ____

22. What is the command to list the access permissions on one of your shares?

    ____

23. What is the command you would use to give another user or group access to an existing share?

    ____

24. Switch to your **Windows 7** VM, and try logging in as one of the users you created who is a member of the group who should have access to the SMB share you created. Verify that you can access the share as that user. If you can't remember the steps to do this:

    i. open the Windows file explorer;

ii. click on the **Network** item in the left-hand column – if prompted to do so, enable Network Discovery;
iii. locate the server in the discovered list (**CORPDC1**) and open it – you may to click the circular arrow at the right-hand end of the address bar to refresh the view;
iv. locate the share, and open it – try to create a file to verify that you have full access (or whatever level of access you specified when you granted access to it)

25. Boot up the **Windows 10** VM, log in (username **root**, password **toor**), give it an IP address in the appropriate subnet and set the **Gateway** and **DNS** values to the IP address of the Windows 2016 Server, and finally add it to the domain, enabling you to have two users in the one group logged in at the same time.

In addition to being able to administer the users and groups within a domain, Active Directory lets you change settings and install software on computers that are joined to the domain. This is done through what is known as *Group Policies*. Group Policy is a convenient way to configure computer and user settings on networks based on Active Directory Domain Services (AD DS). It enables businesses to reduce costs, control configurations, and harden security. For the final part of the tutorial, you'll configure two group policies to control connected computers.

26. First, you need to set up a shared folder into which you can place items for installation. In Windows 2016 Server, and using the command line if possible:

    i. create a new folder at the root of the C: drive called **Resources**
    ii. create an SMB share for that folder
    iii. grant full access to that SMB to the group **Everyone**.
    iv. in the GUI, browse to **C:\Users\Administrator\Downloads** folder and copy the file **`StairDismount.msi`**
    v. paste this file into the Resources folder you just created. This makes it accessible over the network to the machines that you wish to install it on.

27. Next, you need to create a group policy to actually install the software:

    i. In the **Server Manager**, open **Group Policy Management** from the **Tools** menu
    ii. In the left-hand pane of the Group Policy Management window, expand the Forest and then the Domains folder to locate your server's domain. Right click on it and select **Create a GPO in this domain, and Link it here...**
    iii. Give your GPO a name (e.g., **Install StairDismount**) and click **OK**.
    iv. Expand the domain hierarchy in the left-hand pane so that you can locate the new GPO policy you just created, right click on it, and select **Edit**.
    v. In **Computer Configuration**, expand **Policies**, and then expand **Software Settings**, then right click on **Software installation** and select **New > Package**.
    vi. In the window that opens, you'll need to navigate to and select the **StairDismount** installer that you copied to the Resources folder, **however** this needs to be done using a

network path, not a local file system path. You can accomplish this by typing the path to the server (**\\CORPDC1**) in the field at the top of the Open dialog and pressing enter. The shared folders will appear below, and you can now navigate to the **StairDismount** installer and click **Open**.

vii. Click **OK** on the **Deploy Software** dialog, leaving the selected option on *Assigned*.

viii. Close the Group Policy Management Editor Window

28. Before testing this policy, create another:

    i. in the Pictures folder there are multiple wallpaper images. Copy one of these to your newly-created **Resources** folder

    ii. In the left-hand pane of the Group Policy Management window, right click on your domain and again select **Create a GPO in this domain, and Link it here...**

    iii. Give your GPO a name (e.g., **Set Desktop**) and click **OK**

    iv. Locate the new GPO policy you just created in the domain tree, right click on it, and select **Edit**

    v. Expand **User Configuration**, **Policies**, **Administration Templates**, **Desktop**, and then click on the inner **Desktop**

    vi. In the right-hand pane, right-click on **Desktop Wallpaper** and select **Edit**

    vii. In the **Desktop Wallpaper** window that opens, check the **Enabled** setting, and in the **Wallpaper Name** field, enter the full network path to the wallpaper image you copied to the Resources folder (eg., `\\CORPDC1\Resources\wallpaper.jpg`). (Note – if you enter this path incorrectly, you may end up with a black wallpaper on the client after the policy is applied).

    viii. Choose a **Wallpaper Style** setting.

    ix. Click OK, and close the Group Policy Management Editor window.

29. You now have two new policies, although your client computers don't know about it yet. The quickest way to have them apply the policy is to type the following into the command line on each network client (i.e., your Windows 7 and Windows 10 virtual machines):

    ```
    gpupdate /force
    ```

    After the local policies are updated from the server, the machine will need to be restarted. After you reboot it, and have logged back in, can you find the installed software?

    In a real-world environment, you wouldn't have to go to the machines to physically update their policies, but this is a quick and easy way to see the change during the tutorial. Real-world systems have dozens, or even hundreds of policies, which includes settings that determine when the policy updates from the server are applied.

Conclusion

Hopefully today was an intriguing look at a different command line operating environment to what you saw in Unix previously. There is clearly a different structure to the commands, but the tasks that you can carry out are not that dissimilar. You can configure the network interface, create files, folders, users and groups, create network shares, and grant access permissions. You will get more exposure to these and other PowerShell commands in the next tutorial.

Skills to Master

To successfully complete this tutorial, you must have mastered the following skills:

- set the domain controller’s IP address using PowerShell
- create users and groups using PowerShell, including establishing restrictions on them such as which workstations they can log in on
- create an OU (organisational unit) in PowerShell
- create file shares in PowerShell, including specifying access levels
- grant share access to individual users
- create group policy objects (GPO) to install software on managed workstations

You could be required to demonstrate any of these skills in this module’s practical exam.