KIT304 Server Administration and Security Assurance

Tutorial 8

Goal

Today you will continue to build on the experience you gained in the previous tutorial on creating users and groups. You will also get some experience writing PowerShell scripts.

Introduction

During KIT104 you undertook some tutorial exercises involving shell scripting in Unix. In this unit you've gone into much more depth on the commands that are used to undertake system administration tasks on Unix, but that hasn't extended to scripting.

In the previous tutorial in this module, you had your first exposure to the Windows PowerShell, and to the commands that you can use to undertake the same kinds of administrative tasks that you used previously on the Unix platform. Today, you will explore how you can write scripts to group commands together, and add logic to control the way that those tasks are executed. This is very useful when you want to undertake repetitive tasks. You *could* have written similar scripts on Unix using your shell scripting knowledge.

During today's tutorial, it is assumed you will have the previous tutorial with you so you can easily find the commands you need to complete some of today's tasks – there will be less hand-holding this time around.

Note also that this is the last Windows Administration tutorial before your next in-semester practical exam.

Activity

1. Start up both the **Windows 2016 Server** and **Windows 7** virtual machines. Assign them both an IP addresses in the **`192.168.1.0`** network. Be sure to set both the DNS and Gateway values on Windows 7 to the IP address of the server.

   Confirm that you can ping between both VMs, and then add the Windows 7 VM to the domain of the Windows 2016 Server (**networks.local**).

2. In the Windows PowerShell on Windows 2016 Server, create a new **Organisational Unit** to add users and groups to. Write the command down that you used for easy later reference:

   _____

3. Create a **Group** in this OU (Organisational Unit).

   _____

4. Create a **User** in the new OU who will able to log on to the domain via the **Windows 7** Computer:

   _____

   _____

   _____

5. Using PowerShell, create the directory **`C:\scripts`**. Change into this directory (**`cd c:\scripts`**) and use the **`New-Item`** command to create a file called **`test.ps1`**. To edit this empty script file, enter the command **`ise test.ps1`**. What command did you enter to *create* the file?

   _____

6. In the ISE Editor write the following simple script:

   ```
   echo "Hello World"
   ```

   and choose **File -> Save** to ensure the edited version is written to disk. To execute the script type **&** and the path to the script in the console (i.e., **`&./test.ps1`**)

   When you run this, it does what you probably expect it would. But from a security perspective, you may want to be careful not to run scripts from untrusted sources. Use the search feature on the right-hand side of the ISE window to learn what you can about **execution policy** and see what the different policies are that can be set for scripts.

   What Policy is currently set on our **Windows2016 Server**?

   _____

   Which command can you use to change the execution policy of the system? (You don't have to run this command)

   _____

7. You now have a script that you can run any time. Use the **New-Alias** cmdlet to assign it a name to make it easier to execute in the future. You can learn more about this command here:

   https://technet.microsoft.com/en-us/library/ee176913.aspx

   What is the command that you used to do this?

   _____

   Once you've defined an alias for a script, you no longer need to enter the **&** character, or the path to the script, in order to run it – just enter the alias name instead.

8. Create a new file in your scripts directory called **users.csv** and open it in ISE. At the top, write a series of column headings to represent the data in the file:

   ```
   UPN, Name, Password, FirstName, Surname
   ```

   (UPN is short for UserPrincipalName, which is one of the parameters you use when creating new users, e.g., **jane@networks.local**). On the lines below, enter relevant details for 10 users that you will create with a script, instead of manually entering the **New-ADUser** command 10 times, or using the GUI to do the equivalent. In the real-world, you would most likely get this information from a database or other source, but manually creating it is sufficient for today's exercise.

   Be sure to save the file before proceeding to the next step.

9. Create another file, this time with the extension **.ps1**. Open this file in the ISE editor, and enter into it the following script, using what you already know about how to create users in PowerShell to execute the appropriate command inside the **foreach** loop to create a new user.

   In this script, you can access the comma separated values from the **users.csv** file (such as the name and password) by referencing them with the **$user** variable followed by a period, and then the column name from the first line (leave out the quotes when specifying the password).

   ```
   Import-Module ActiveDirectory
   $users = Import-CSV "c:\scripts\users.csv"
   foreach ($user in $users) {
       create user here
   ```

   _____

   _____

   _____

   ```
   }
   ```

Again, be sure to save this file before proceeding to the next step.

10. Run the script you've just created on the ISE command line. If you get any error messages, read them carefully, and adjust the script as necessary to fix them. Make sure your script can create all of the users before proceeding to the next step. (You can verify that the accounts were created by using either the GUI **Active Directory Users and Computers** tool, or the PowerShell command line.)

11. Create another script that adds each of the users in the **users.csv** file to the group that you created in step 3, and run it.

12. Create a shared folder this group has full access to. Then login with one of these accounts on the **Windows 7** system and confirm that they have access to the folder.

13. It is possible to block access to a share for a user that is in the group without also blocking the rest of the group. What is the command to do this? (Hint: Look at the SMBShare documentation linked to in step 20 of the previous tutorial). What is the command?

    _____

14. Create another group, and add at least one of the new users to it. What PowerShell command gives them only read access to this share?

    _____

15. Create another CSV file of user names, with a different set of details to the file you created in step 8. This time include an extra column that contains a **1** or a **2** to indicate which of two groups each user should be in (presumably, one of the two groups you already have created). Make sure half of the new users are in one group, and the other half in the another. Then using the document at the following link (which details *conditional syntax*) write a script that will use this information to add the user to the correct group in the same loop that creates their account.

    https://technet.microsoft.com/en-us/library/hh847876.aspx

Conclusion

The next tutorial you will have will be this module's practical exam. Ask for help in the remaining time of this tutorial if there is anything that you are confused about in this module.

Skills to Master

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

- create aliases for PowerShell scripts
- grant different levels of access, or block access to SMB shares for different users
- open and parse CSV files in PowerShell scripts
- execute loops and conditional expressions in PowerShell scripts
- use PowerShell scripts to create users and set group membership based on the content of a CSV file

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