Lab - Write Basic Scripts in Windows and Linux

Objectives

In this lab, you will write basic scripts in different scripting languages to help understand how each language handles automating tasks.

Background / Scenario

Writing scripts to automate common administration functions saves time and gives the administrator flexibility to perform other tasks. In the lab, you will write three types of scripts that will perform similar tasks. Compare the different languages as you automate some simple task.

Required Resources

- Windows PC
- VM running a Linux distribution

Instructions

Step 1: Create a Windows batch script.

a. In a text editor, such as Notepad, save a text file named **info.bat** in your home directory (C:\Users*yourusername*) with the following text:

```
@echo off
echo Computer Name is: %computername%
echo Windows version is:
ver
echo CPU is: %PROCESSOR_IDENTIFIER%
echo Total memory is:
rem Windows Management Instrumentation Command (WMIC) is a command line utility that can retrieve information about local or remote computers. For more inline information, enter help wmic or wmic /? at the command prompt.
wmic ComputerSystem get TotalPhysicalMemory
echo The disks that are installed and their freespace:
wmic logicaldisk get size,freespace,caption
echo All the %computername% IP addresses
rem netsh is a command line scripting utility that allows the users to view or modify the network configurations of a running computer. For more inline information, enter nesh /? at the command prompt.
rem findstr is used for searching for a text string in files. For more inline information, enter findstr /? at the command prompt.
netsh interface ip show address | findstr "IP Address"
```

b. Open a command prompt and navigate to your home directory.

c. List the content of your home directory and verify that the file **info.bat** is saved with the correct file. If not, rename the file, for example, **rename info.bat.txt info.bat**.

d. At the prompt, enter **info.bat** to run the script.

Questions:

What was the output?

Type your answers here.

What are the %*text*% used for in the script?

Type your answers here.

Identify what the following commands do in the script:

echo:

Type your answer here.

findstr:

Type your answer here.

netsh:

Type your answer here.

ver:

Type your answer here.

wmic:

Type your answer here.

Step 2: Create a Powershell ISE script.

a. Click **Start**, Search for **PowerShell ISE** and right-click the selection and click **Run as an administrator.**

b. Verify that you are in your home directory: PS C:\Users*YourUsername*

c. To allow the script to run, enter **Set-ExecutionPolicy RemoteSigned** at the prompt. Click **Yes** to allow the script to run. The settings can be changed back to **No** after the script is complete.

```
PS C:\Users\YourUsername> Set-ExecutionPolicy RemoteSigned
```

d. Choose **File** -> **New** and create a new script.

e. Enter the following text into the **Untitled.ps1** window and save it as **info.ps1** in your home directory.

```
Write-Output "Computer name is:"
get-content env:computername
Write-Output "Windows version is:"
(Get-WmiObject -class Win32_OperatingSystem).Caption
Write-Output "CPU is:"
Get-WmiObject Win32_Processor | findstr "Name"
Write-Output "Total Memory is:"
[Math]::Round((Get-WmiObject -Class win32_computersystem -ComputerName
localhost).TotalPhysicalMemory/1Gb)
```

```
Write-Output "The Disks that are installed and their freespace:"
Get-WmiObject -Class Win32_logicaldisk -Filter "DriveType = '3'"
Write-Output "IPv4 addresses"
Get-NetIPAddress -AddressFamily IPv4 | Sort-Object -Property InterfaceIndex | Format-Table
```

Note: The command **Get-NetIPAddress** is not available in Windows 7.

Note: Within PowerShell ISE, you can press F1 or select **Help** > **Windows PowerShell ISE Help** to get more information.

f. To see the functions of each command, click **Add-ons**, verify that **Command** is checked. In the Command tab, enter the name of the command in the **Name** field. Select the desired command and click the ? for more information regarding the desired command.

In Windows 7, click **Help** > Select **Windows PowerShell Help**. Select **Windows PowerShell Cmdlet Help Topics**. Search for the desired command.

g. Enter **.\info.ps1** at the PS prompt. **Note**: Make sure you are using the correct slash.

```
PS C:\Users\YourUsername> .\info.ps1
```

Question:

What is the output of the script?

Type your answer here.

h. Compare the two scripts. Match the batch command to the PowerShell commands below:

Windows Batch Command	PowerShell Command
echo Computer Name is: %computername%	
echo Windows version is: ver	
echo CPU is: %PROCESSOR_IDENTIFIER%	
echo Total memory is:	
wmic ComputerSystem get TotalPhysicalMemory	
echo The disks that are installed and their freespace:	
wmic logicaldisk get size,freespace,caption	
echo All the %computername% IP addresses	
netsh interface ip show address | findstr "IP Address"	

Step 3: Create a BASH script.

A text editor is used to create an executable script. One of the text editor tools, vi, or the improved vi version, vim, is based on letter and number-based commands to modify text. For example, **dd** will delete the whole line on which the cursor is placed. **5dd** would delete 5 lines. When vi is in command mode, input is interpreted as a command.

To enter insert mode at the current cursor position type **i**. To append text at the end of the current line, type **a**. To insert text on a new line below the current line, type **o**. Use the Esc key to exit out of insert mode to command mode.

To save a file in the vi editor use **:w** from command mode. To save and quit, type **:wq**. To quit without saving type **:q!**.

Depending on your version of Unix-like OS, you may find other text editor tool, such as nano, pico, and gedit. The text editing tools, such as vi, nano, and pico, are accessible through the command line; while the GUI-based text editors, like gedit, may be located via the application menu or the command line.

a. Start up a Linux computer or VM.

b. Use a text editor tool and create a file named **info.sh** in your home directory with the following text:

```
#!/bin/bash
echo "Computer name is: "  $HOSTNAME
echo "Operating System is:"
cat /etc/os-release | grep PRETTY_NAME
echo "CPU is"
lscpu | grep "Model name:" | sed -r 's/Model name:\s{1,}//g'
echo "Total Memory is"
cat /proc/meminfo | grep  "MemTotal"
echo "The disks that are installed and their freespace"
df -h
echo "All the" $HOSTNAME "IP addresses"
hostname -I
```

c. Open a terminal and navigate to your home directory. To make the script executable, enter **chmod 755 info.sh** at prompt.

d. At the prompt, enter **./info.sh** to execute the script.

Questions:

What is the output of the script?

Type your answer here.

What does the “#!/bin/bash” mean at the beginning of the script?

Type your answer here.

What command would you use to learn more about the **df** and **lscpu** commands?

Type your answer here.