Chapter 8 Lab

This chapter has probably covered more, and more difficult, new concepts than any chapter so far. Hopefully we were able to make it all make sense, but these exercises should help you cement everything. See if you can complete them all, and remember that there are companion videos and sample solutions at MoreLunches.com. Some of these tasks will draw on skills you learned in previous chapters, as a way of refreshing your memory and keeping you sharp.

1. Identify a cmdlet that will produce a random number.  
     
   Get-Random
2. Identify a cmdlet that will display the current date and time.  
     
   Get-Date
3. What type of object does the cmdlet from task #2 produce? (What is the type name of the object produced by the cmdlet?)  
     
   System.DateTime
4. Using the cmdlet from task #2 and Select-Object, display only the current day of the week in a table like this (caution: The output will right-align, so make sure your PowerShell window doesn’t have a horizontal scroll bar):  
     
   Get-Date | select DayofWeek
5. Identify a cmdlet that will display information about installed hotfixes.  
     
   Get-Hotfix
6. Using the cmdlet from task #5, display a list of installed hotfixes. Sort the list by the installation date, and display only the installation date, the user who installed the hotfix, and the hotfix ID.  
     
   Get-HotFix | Sort InstalledOn | Select InstalledOn,InstalledBy,HotFixID
7. Repeat task #6, but this time sort the results by the hotfix description, and include the description, the hotfix ID, and the installation date. Put the results into an HTML file.  
     
   Get-HotFix | Sort Description | Select Description,InstalledOn,InstalledBy,HotFixID | ConvertTo-Html -Title "HotFix Report" | Out-File HotFixReport.htm
8. Display a list of the 50 newest entries from the Security event log (you can use a different log, such as System or Application, if your Security log is empty). Sort the list so that the oldest entries appear first, and so that entries made at the same time are sorted by their index. Display the index, time, and source for each entry. Put this information into a text file (not an HTML file, just a plain text file). You may be tempted to use Select-Object and its –first or –last parameters to achieve this; don’t. There’s a better way. Also, avoid using Get-WinEvent for now – there’s a better cmdlet to work with for this particular task.  
     
   Get-EventLog -LogName System -Newest 50 | Sort TimeGenerated,Index | Select Index,TimeGenerated,Source | Out-File elogs.txt