Powershell 3 Lab Chapter 9

Alex Stock

1. **Would the following command work to retrieve a list of installed hotfixes from all domain controllers in the specified domain? Why or why not? Write out an explanation, similar to the ones I provided earlier in this chapter.**

Get-Hotfix -computerName (get-adcomputer -filter \* -searchbase "ou=domain controllers,dc=company,dc=pri" | Select-Object -expand name)

This command should work. The commands in brackets will be executed first, and the results passed to the ComputerName parameter of the Get-Hotfix commandlet. The Get-ADComputer commandlet outputs objects of the type ADComputer. The Select-Object commandlet with the expandProperty parameter turns them into strings. The ComputerName parameter of the Get-Hotfix commandlet accepts strings.

1. **Would this alternative command work to retrieve the list of hotfixes from the same computers? Why or why not? Write out an explanation, similar to the ones I provided earlier in this chapter.**

get-adcomputer -filter \* -searchbase "ou=domain controllers,dc=company,dc=pri" | Get-HotFix

This command should not work. The Get-ADComputer commandlet outputs objects of the type ADComputer. The Get-HotFix commandlet has only one parameter that accepts pipeline input – ComputerName – and this parameter is of type String. There is no parameter of type ADComputer for the output of Get-ADComputer to bind to.

1. **Would this third version of the command work to retrieve the list of hotfixes from the domain controllers? Why or why not? Write out an explanation, similar to the ones I provided earlier in this chapter.**

get-adcomputer -filter \* -searchbase "ou=domain controllers, dc=company,dc=pri" | Select-Object @{l='computername';e={$\_.name}} | Get-Hotfix

This command should work. The Get-ADComputer commandlet outputs objects of type ADComputer. The Select-Object commandlet turns these into objects that contain a member named ComputerName of type String. This member will match with the ComputerName parameter of the Get-Hotfix command.

1. **Write a command that uses pipeline parameter binding to retrieve a list of running processes from every computer in an AD domain. Don’t use parentheses.**

Get-ADComputer -filter \* -searchBase "ou=domain controllers, dc=company,dc=pri" | Select-Object @{l='ComputerName';e={$\_.Name}} | Get-Process

1. **Write a command that retrieves a list of installed services from every computer in an AD domain. Don’t use pipeline input; instead use a parenthetical command (a command in parentheses).**

Get-Service –ComputerName (Get-ADComputer -filter \* -searchBase "ou=domain controllers, dc=company,dc=pri" | Select-Object -expandProperty Name)

1. **Sometimes Microsoft forgets to add pipeline parameter binding to a cmdlet. For example, would the following command work to retrieve information from every domain controller in the domain? Write out an explanation, similar to the ones I provided earlier in this chapter.**

get-adcomputer -filter \* -searchbase "ou=domain controllers, dc=company, dc=pri" | Select-Object @{l='computername';e={$\_.name}} | Get-WmiObject -class Win32\_BIOS

This command should not work. The Get-ADComputer and Select-Object commandlets produce objects containing a String member named ComputerName. However, the ComputerName parameter of the Get-WMIObject command does not accept pipeline input. In fact, none of the parameters of Get-WMIObject accept pipeline input. The output of Select-Object will not bind to any input of Get-WMIObject, so the command will fail.