Diving into PowerShell

@iLabAfrica, Strathmore University
John Ombagi

Execution Policy

- The execution policy in PowerShell is not a security boundary.
- It’s designed to prevent a user from unknowingly running a script. A determined user can easily bypass the execution policy in PowerShell.
- PowerShell command can be run interactively even with the execution policy setting. It is effected when exectuing Powershell Scripts.
- The Get-ExecutionPolicy cmdlet is used to determine what the current execution policy setting is and the Set-ExecutionPolicy cmdlet is used to change the execution policy.

Execution Policy in Practice

- Check the current execution policy:
  - Get-ExecutionPolicy (should be Restricted by default)
- Create a Powershell script with the following code:
  - Get-Service -Name W32Time | Stop-Service -PassThru
  - Save the file as **Stop-TimeService.ps1**
- When executing, you should see and error:
  - .\Stop-TimeService.ps1
- Best option is RemoteSigned policy which requires downloaded scripts to be signed by a trusted publisher in order to be run.
  - Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
- Now the script should run!
  - Remember to restart Windows Time service: Start-Service -Name w32time

More Tricks

- Another way to run PowerShell scripts is to use Bypass as ExecutionPolicy:
  - powershell.exe -ExecutionPolicy Bypass -File "C:\Stop-TimeService.ps1"
- While using the ISE, under an existing PowerShell console or ISE session run the following:
  - Set-ExecutionPolicy Bypass Process
- It’s not advisable to use the use a common policy you will see online:
  - Set-ExecutionPolicy Unrestricted
- Execution Policy may be enforced by Group Policy. You can see the execution policy set at the various scopes using
  - Get-ExecutionPolicy -List

The Help System

- Mastering the help system is the key to being successful with PowerShell.
- Two groups of IT Pros were given a written test without access to a computer to determine their skill level with PowerShell....

The differences in the two tests mentioned in the previous scenario were observed because experts don’t memorize how to use thousands of commands in PowerShell.

They learn how to use the help system within PowerShell extremely well.

CMDs in Powershell

- So how do you figure out what the commands are in PowerShell?
- Both Get-Command and Get-Help can be used to determine the commands.
- Get-Help is a multipurpose command. It helps you learn how to use commands once you find them.
- Get-Help can also be used to help locate commands, but in a different and more indirect way when compared to Get-Command.

Get-Help

- When Get-Help is used to locate commands, it first searches for wildcard matches of command names based on the provided input.
- If it doesn’t find a match, it searches through the help topics themselves, and if no match is found an error is returned.
  - Contrary to popular belief, Get-Help can be used to find commands that don’t have help topics.
- The first thing you need to know about the help system in PowerShell is how to use the Get-Help cmdlet.
- The following command is used to display the help topic for Get-Help.
  - Get-Help -Name Get-Help

Get-Help Parameters

- While not specific to PowerShell, a parameter is a way to provide input to a command.
- Get-Help has numerous parameters that can be specified in order to return the entire help topic or a subset of it.
- The following parameters each reside in different parameter sets:
  - Full
  - Detailed
  - Examples
  - Online
  - Parameter
  - ShowWindow

Decipher the cryptic output :)

- When the Full parameter of Get-Help is specified, the entire help topic is returned.
  - Get-Help -Name Get-Help -Full
- Display available help topics: Get-Help *
- Display basic information about a cmdlet
  - Get-Help Get-Alias
  - Help Get-Alias
  - Get-Alias -?
- These commands display basic information about the Get-Alias cmdlet.
- The Get-Help and ? commands display the information on a single page. The Help command displays the information one page at a time.

Decipher the cryptic output :)

- Display a list of conceptual topics
  - Get-Help about_* for example Get-Help about_Signing
- Download and install help files
  - Update-Help cmdlet is used when help files are missing
- Display detailed help: Get-Help ls -Detailed
- Display full information for a cmdlet: Get-Help Format-Table -Full
- Display examples for a cmdlet: Get-Help Start-Service -Examples
- Display parameter help: Get-Help Format-List -Parameter GroupBy
- Search for a word in cmdlet help
  - Get-Help Add-Member -Full | Out-String -Stream | Select-String -Pattern Clixml
- Display online version of help: Get-Help Get-Member -Online
- Display a list of topics that include a word: Get-Help remoting
- Display help for a script: Get-Help C:\PS-Test\MyScript.ps1

Get-Command

- Get-Command is designed to help you locate commands.
- Running Get-Command without any parameters returns a list of all the commands on your system.
- Using Get-Command cmdlet to determine what commands exist for working with processes:
  - Get-Command -Noun Process
- Using wildcard characters with the Name parameter
  - Get-Command -Name *service* -CommandType Cmdlet, Function, Alias

Challenge

- Learn a PowerShell command a day.
- Get-Command | Get-Random | Get-Help -Full

Assessment

- Using what we have learned try the following:
  - Is the DisplayName parameter of Get-Service positional?
  - How many parameter sets does the Get-Process cmdlet have?
  - What PowerShell commands exists for working with event logs?
  - What is the PowerShell command for returning a list of PowerShell processes running on your computer?
  - How do you update the PowerShell help content that’s stored on your computer?

EOF

jnyabuti@strathmore.edu