Name Click here to enter name ID Click here to enter id

Remoting Endpoints

Exercise 3.5

# Instructions

Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

The term “endpoint” has a special meaning in the context of PowerShell remoting. It refers to a particular session configuration on a host that can be accessed by remoting. A machine can have more than one such configuration, each of which can be distinct in granular ways. For example, if you need to remote a task on a remote target that requires interaction with legacy software designed for 32-bit architecture, you can specify a separate endpoint configuration that can handle that task.

This exercise will examine endpoint configurations on a couple of hosts, and then briefly browse the capabilities of one of the endpoints.

# Requirements

* SSH client (There is already a command-line OpenSSH client available in current versions of Windows, macOS, and Linux)
* Internet access

# Setup

Use SSH as a remote access tool to connect to the lab:  
Computer: **cit361-lab.cit.byui.edu**Port: **22** or **443** *(Important: for SSH, 443 is a* nonstandard *transport layer port number.)*  
Username: The mailbox portion of your BYU-I email address, usually three letters and five numbers. (Example: If your BYU-I email is **mer23079@byui.edu** you would use **mer23079** for your username.)  
Password: Your I-Number.

If you are using OpenSSH from a CLI, the parameters above can be specified like this:  
**ssh -p 22 mer23079@cit361-lab.cit.byui.edu**  # if that doesn’t work, try **-p 443**   
If you successfully type the correct password when prompted, you will see a prompt like this:  
PS C:\Users\mer23079>

# Task 1—Examine Session Endpoint Configurations

## Steps

1. After using SSH to log in, enter:   
   **Get-PSSessionConfiguration**   
   Pipe the result through the Measure-Object cmdlet. How many endpoint configurations are there on the bastion host? Click or tap here to enter text.
2. Run the same command on DC:   
   **Invoke-Command -ComputerName DC -ScriptBlock {Get-PSSessionConfiguration}**   
   How many endpoint configurations are there on DC? Click or tap here to enter text.

# Task 2—Use a Session Endpoint Configuration

## Steps

1. Connect to the virtual machine named DC using the **microsoft.powershell.workflow** endpoint:   
   Enter-PSSession DC -ConfigurationName microsoft.powershell.workflow
2. You should now see a prompt that starts with **[DC]**, but there is no user or path information shown in the prompt. This endpoint is a carefully restricted environment. Let’s find out what PowerShell command subset is allowed with this endpoint:   
   **Get-Command**   
   How many commands do you see that begin with the verb Get? Click or tap here to enter text.
3. Get the names and values of all of this session’s PowerShell variables:   
   Get-Variable \*
   1. What is the value of HOME? Click or tap here to enter text.
   2. What is the value of PSUICulture? Click or tap here to enter text.
4. Let’s see what modules are loaded.   
   Get-Module \*
   1. What are the names of the loaded modules? Click or tap here to enter text.
   2. By default, this alias shows its output as a table. You can get more information by piping the output through the **Format-List** cmdlet. Use its convenient alias **fl**, like this:   
      **Get-Module \* | fl**   
      What are the NestedModules in the PSWorkFlow module? Click or tap here to enter text.
5. Using the microsoft.powershell.workflow endpoint configuration, are you able to interact with DC’s filesystem in any way, such as setting locations or listing directory contents? If so, what commands let you do so? Click or tap here to enter text.

# Deliverable

When you’re finished, **exit** and **exit**. Upload this document with completed answers to I-Learn Canvas.