Advanced Scripting   
Lab: Power of Remoting

Last Updated: 9/14/2024 11:27 AM Version 1.1  
Document Prepared for: CYBER360 Student

# Instructions

Work together with your partner to complete these lab activities. Prepare your own report as a Microsoft Word document, in which you explain the work you and your partner did to accomplish this lab. Where appropriate, you may illustrate your report by including screen-capture evidence showing that you accomplished each lab task. *(DO NOT submit one jointly-authored document. Each team member must write and submit their own individual report. DO NOT just submit a sequence of screenshots; you must submit a professionally written report that explains what you did, what you learned, and what specific evidence each screenshot illustration conveys.)*

# Overview

In your chapter 3 exercises, you used a few virtual machines in an Active Directory domain that were *already* pre-configured to support PowerShell remoting. There are still more such domain-joined lab VMs to find! You and your partner will explore them together.

To complete this lab, use the CIM and PSRemoting skills you practiced in your homework exercises. Also, please leverage Internet resources at your disposal, including advanced search engines and generative-AI LLM chatbots. Collaborate with your partner and share successful searches and chatbot conversation prompts. Include your most useful search queries and chatbot conversations in your report.

# Requirements

* SSH client
* Internet access

# Setup

Use SSH as a remote access tool to connect to the lab’s “jump server:”   
Computer: **cit361-lab.cit.byui.edu**Port: **22**   
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 mer23079@cit361-lab.cit.byui.edu**

# Tasks:

## Explore the domain-joined lab VMs

1. SSH to the jump server as mentioned in the Setup above. Get the name of the jump server:  
   **hostname**
2. Find the names and addresses of all the domain-joined virtual machines.
   * Hints:   
     **Get-ADComputer -filter \***   
     to see what hostnames are registered in the domain’s Active Directory.
   * Write your own looping code to “ping” every address on the subnet. Then examine the ARP table with   
     **arp -a**   
     to see which of those addresses held machines. (It even shows those that ignored the ping!)
3. Use CIM commands, PowerShell remoting, etc. to profile as much system information as you can find about each of the discovered domain machines. More hints:
   * + Use **Get-CimClass** to get a list of the CIM classes; search them for relevant information.
     + You can see what properties a CIM class returns by looking at the **CimClassProperties** member. Example: **(Get-CimClass Win32\_PingStatus).CimClassProperties**
     + Create an array of computer names to get more information all at once.
     + Create an array of **CIMSession**s to get more information all at once.
     + Use **Select-Object** to filter output so that it returns only the information you want.
       - Example: remember that **Get-ChildItem** returns a **Length** property in bytes. If you would rather see output showing lengths in kilobytes, you could try something like this:   
           
         **Get-ChildItem|Select Name,@{Name=**'**Size**'**;Expression={$\_.Length/1kb}}**
       - Or you could change the filter for prettier formatted sizes in megabytes:   
           
         **Select Name,@{Name=**'**Size**'**;Expression={  
          "$([math]::round($\_.length/1mb,2)) MB"}}**
       - In general, you can create a calculated column by creating a hash table, with a **Name** item and an **Expression** that defines the column value you would like to create.
     + If a CIM command doesn’t work for a particular target, try PSRemoting. If that doesn’t work, try alternatives, such as using both **powershell** and **pwsh** for PSRemoting, using different PSRemoting transport types (SSH instead of WSman), etc.

Discover and make tables of profile data for:

* 1. Versions and editions of PowerShell (Hint: **$PSVersionTable**) on each domain VM.
  2. Operating system, OS version/build, and architecture of each domain VM.
  3. Number of processors (CPUs).and amount of “physical memory” installed in each domain VM.
  4. Network adapter types, IP (layer 3) addresses, and MAC (layer 2) addresses.
  5. Attached storage devices (“disk drives” and/or volumes) and their storage amounts (total capacity, current amount consumed, current amount free, current percentage free).
  6. Services on each domain VM: how many are configured? How many of those are running?
  7. Other interesting profile information:
     + Is every domain VM in the same time zone?
     + Are any domain machines acting as file servers? If so, name their file shares.
     + Look for and profile some other interesting feature or setting not already mentioned here.

For example, here’s a partially filled in table for the first profile item, a PowerShell version/edition table:

|  |  |
| --- | --- |
| HORACE | 5.1.19041.4522/Desktop, 7.4.2/Core |
| DC | 5.1.17763.5933/Desktop |
| SLAYGORE | 5.1.14393.6343/Desktop |
| ERIC | 7.2.4/Core |
| … etc. |  |
|  |  |
|  |  |

# Deliverable

Again, as mentioned in the instructions above:

Prepare your own report as a Microsoft Word document, in which you explain the work you and your partner did to accomplish this lab. Where appropriate, illustrate your report by including screen-capture evidence showing that you accomplished each lab task. Copy the most useful search queries and chatbot conversation prompts that you and your partner found most helpful, and include them in your report. *(DO NOT submit one jointly-authored document. Each team member must write and submit their own individual report. DO NOT just submit a sequence of screenshots; you must submit a professionally written report that explains what you did, what you learned, and what specific evidence each screenshot illustration conveys.)*

Submit your report in I-Learn Canvas.

# Scoring Standard (“rubric”)

|  |  |
| --- | --- |
| Report in Microsoft Word format, with your name at the top, and the name(s) of your team partner(s) below your name. | one point |
| Narrative or explanations about what you and your team partner(s) did to find and profile each domain machine. | 8 points |
| Profile tables: |  |
| PowerShell version/edition profiles | 2 |
| OS and architecture profiles | 2 |
| “Virtual hardware” (processor cores, memory) | 2 |
| Network profiles | 2 |
| Storage profiles | 2 |
| Operating systems, application software | 2 |
| Services or server-processes | 2 |
| Other profiles | 2 |
| TOTAL | 25 points |