Advanced Scripting   
Lab: Power of Remoting

Last Updated: 2/1/2024 9:52 AM Version 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. 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 should write and submit their own individual report.)*

# Overview

Much of the material presented in textbook chapter 3 was about enabling and configuring computers to be able to support PowerShell remoting. However, in your chapter 3 exercises, you used a few virtual machines that were *already* pre-configured to support PowerShell remoting. There are still more such lab VMs to find! You and your partner will explore them together.

Your instructor has also provisioned two new virtual machines (a Windows Server and a Linux server) just for your team. PowerShell remoting has not yet been enabled on these hosts, and they are not and will not be joined to a domain. Instead, they will remain independent hosts on the domain’s subnet. You are tasked with enabling and configuring remoting on each of your new machines, so that you can remotely execute script blocks on each.

To help you accomplish the lab activities, 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 prompts. Include your most useful search queries and chatbot conversations in your report.

Your instructor will provide the IP addresses of the two machines assigned to your lab team. Record them at the beginning of 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:

## Connect to each of your team’s lab VMs

1. After using SSH to log in to the “jump server,” use SSH again to connect to your team’s Windows Server. It’s an independent server (it’s not on a domain), but it has a *local* account: **ps** is the username, **PowerShellRocks!** is the password.   
   **ssh ps@***<ip address>*   
   Figure out how to obtain elevated *local administrator* privilege on your independent Windows Server. (Hint: try the same password.)
2. Back on the jump server, use SSH to connect to your team’s independent Linux server. Its local account also has the same **ps** username and **PowerShellRocks!** password. Figure out how to obtain elevated *root* (local superuser) privilege on your Linux Server. (Hint: try the same password.)

## Create your own local user accounts on your team’s lab VMs

1. Figure out and use PowerShell commands to create your own *local* user accounts on your independent Windows Server. Figure out and use bash commands to create your own local user accounts on your Linux server.
2. Figure out and use PowerShell to create objects of type PSCredential. Create a valid credential object that represents your domain account on the jump server. Create valid credential objects that represent your new local account on both of your team’s independent lab VMs.
3. Figure out how to configure your new local user accounts with the capability to use elevated (administrator or root) privilege to execute commands. *This capability must be auditable*, in the sense that privileged access from your account is secured using AAA (authentication, authorization, accounting) controls. (Hint: on Linux servers, the **sudo** command already implements local AAA capability. For your Windows server, see textbook chatper 4 for a few command-logging ideas, then choose and implement one of them as an accounting control.)

## Rename and reboot your team’s lab VMs

1. Figure out and use PowerShell commands or other configuration tools to change the hostname of your team’s independent Windows and Linux servers. (Your instructor will specify your new hostnames.)
2. Restart and verify that each new hostname has taken effect.
3. Verify that the hostname change events and reboot commands were recorded in your independent servers’ system event logs.

## Enable PowerShell remoting on your team’s lab VMs

1. Figure out and use PowerShell commands and other configurations to enable PowerShell remoting between your independent machines. (Hints: **TrustedHosts**, and PSCredential objects. You might also need to configure a PowerShell subsystem for the Linux VM’s SSH daemon **sshd**.) Verify that you can create and use PSSession objects to remotely execute a script block between each of your team’s independent servers.
2. Verify that you can create and use PSSession objects to remote between the jump server and each of your team’s independent servers.

## Explore the rest of the domain-joined lab VMs

1. Starting back with the jump server, find the names of all the domain-joined virtual machines.
   * Hints:   
     **Get-ADComputer -filter \***   
     to see what hostnames are registered in the domain’s Active Directory.
   * Use looping code to “ping” every address on the subnet. Then examine the ARP table with   
     **arp -an**   
     to see which of those addresses held machines. (It even shows those that ignored the ping!)
2. Use CIM commands, PowerShell remoting, etc. to profile as much system information as you can find about each of the discovered domain machines. Create tables that summarize your profiles of:
   * Versions and editions of PowerShell (Hint: **$PSVersionTable**) on each domain VM.
   * Operating system, OS version/build, and architecture of each domain VM.
   * Number of processors (CPUs).and amount of “physical memory” installed in each domain VM.
   * Network adapter types, IP (layer 3) addresses, and MAC (layer 2) addresses.
   * Attached storage devices (“disk drives” and/or volumes) and their storage amounts (total capacity, current amount consumed, current amount free, current percentage free).
   * Services on each domain VM: how many are configured? How many of those are running?
   * 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.

# Deliverable

Again, as mentioned in the instructions above:

Prepare your own team’s report as a Microsoft Word document, in which you explain the work you and your partner did to accomplish this lab. 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 should write and submit their own individual report.)*. Submit your report in I-Learn Canvas.

# Scoring Standard (“rubric”)

|  |  |
| --- | --- |
| Report in Microsoft Word format, with team members’ names at the top | 5 points |
| Local accounts on renamed VMs | 5 |
| PSRemoting enabled and working between your team’s VMs. | 5 |
| PSRemoting enabled and working between each of your team’s VMs and the domain VMs | 5 |
| Profiles of each domain VM:  “Hardware” (processors, memory) profiles | 5 |
| Network profiles | 5 |
| Storage profiles | 5 |
| Operating system/software profiles | 5 |
| Service software profiles | 5 |
| Other profiles | 5 |
| TOTAL | 50 points |