**CYB631 Automating Information Security with Python and Shell Scripting**

**Lab 3: Managing and Hardening Hosts**

Your Name: **Vaibhav A Mayekar**

**Goals:** This lab will guide you through basic PowerShell commands and use them to gather host information from Windows 10 OS.

**Readings and References**:

* The contents of this lab are adapted from Chapter 18, Chapter 21, Chapter 26-28 in the Windows PowerShell Cookbook by Lee Holmes.

**Exercises:**

**[Environment: Starting with PowerShell ISE]**

1. **It is recommended that you use the Windows Server VM on VMware Horizon Desktop provided by this class because 1) Active Directory service used in Exercise I will require Windows Server, you cannot run Exercise I on Windows 10 or Windows 11 OS, and 2) Windows administration configuration, such as Active Directory or Registry, might mess up with the settings on your computer needed for daily works. Please see PaceLabHowTo document in Week1 for instructions to access the Windows Server VM.**
2. Windows 10 environment is needed for the lab. Launch Windows PowerShell ISE. Click Start->Run, and then type PowerShell ISE.
3. Now, open a new PowerShell ISE and run as an administrator. Make sure that you navigate to the directory where your script is. Then shorten the command prompt and change the executive policy to RemoteSigned or Unrestricted. Remote Signed will required all scripts downloaded from the Internet to be signed digitally. Unrestricted will allow execution but provide prompts for users to confirm.

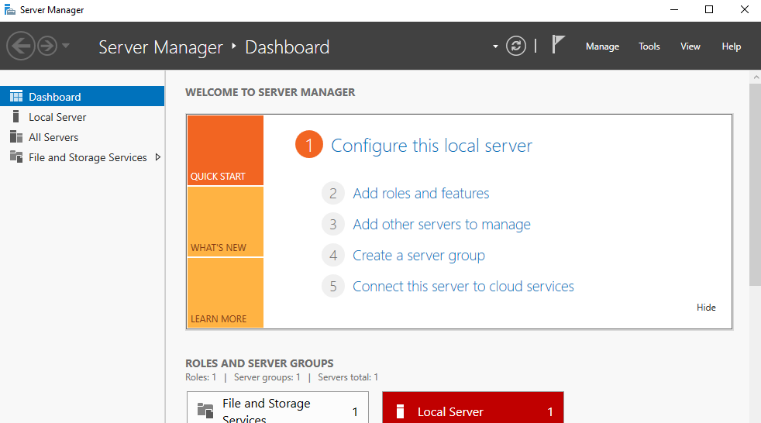
**cd \\pace.edu\shares\users\lchen\Desktop\CYB631**

**function Prompt {"LCHEN PS> "}**

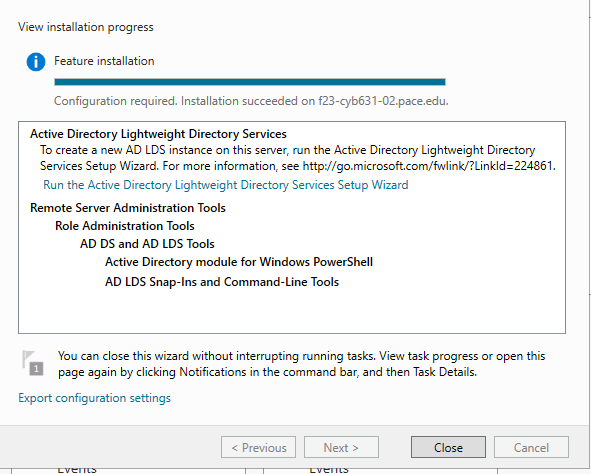
**Set-ExecutionPolicy -ExecutionPolicy Unrestricted**

**[Exercise I: Install Active Directory Lightweight Service]**

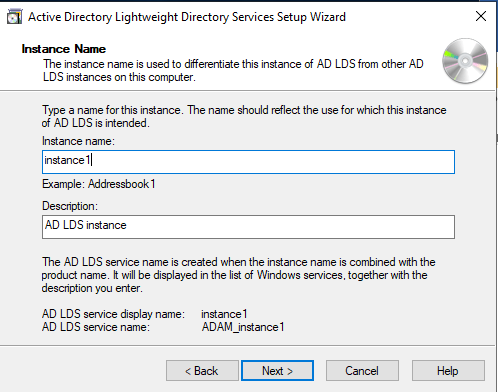
1. Install Active Directory Lightweight Services (AD LDS), a lightweight version of Active Directory.
2. In the search box next to Windows Start, type “server manager” and run as “administrator” to open server manager. The you should see the Dashboard like the one below.

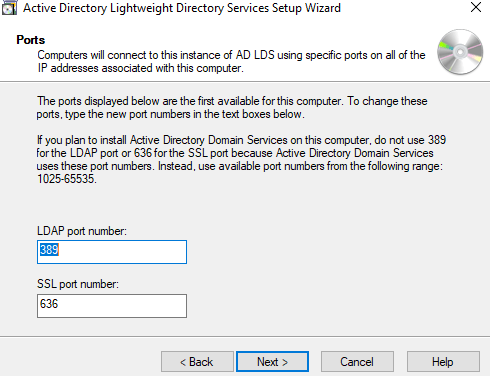


1. Click on **Server Manager**, then **Add Roles and Features**, and check **Active Directory Lightweight Directory Service (AD LDS)**. Follow the instructions to install it. You may need to click on **Add Features, Next, and Install**.
2. Once AD LDS is installed, we will need to run the Active Directory Lightweight Service Setup Wizard. Click on the link in the message box to do so.

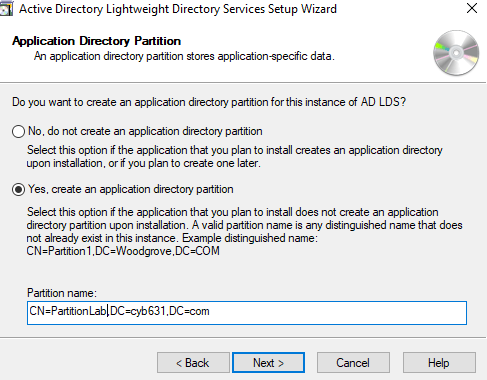


1. In the next step, select to install “a unique instance” and then use the default value for Instance name and Ports, like the ones below.

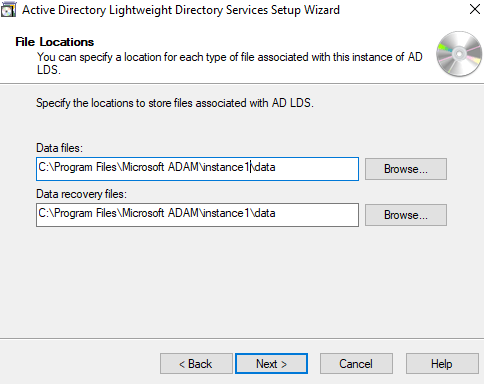




1. Next, select “Yes, create an application directory partition. The, give unique names to the partition, like the one below.



1. Use default names for file location.



1. Next, for Service Account Selection, choose Network service account.
2. Next, for AD LDS Administrators, choose “Currently logged on user.”
3. Next, for Importing LDIF Files, click on the ones that we will use for applications. These are text files which represent data and commands used by LDAP instance. For our testing, click on **MS-User.LDF**.
4. Then follow the instructions to complete the Setup Wizard. Once it is completed, you can click on AD LDS on the Server Manager to see the details of the instance.

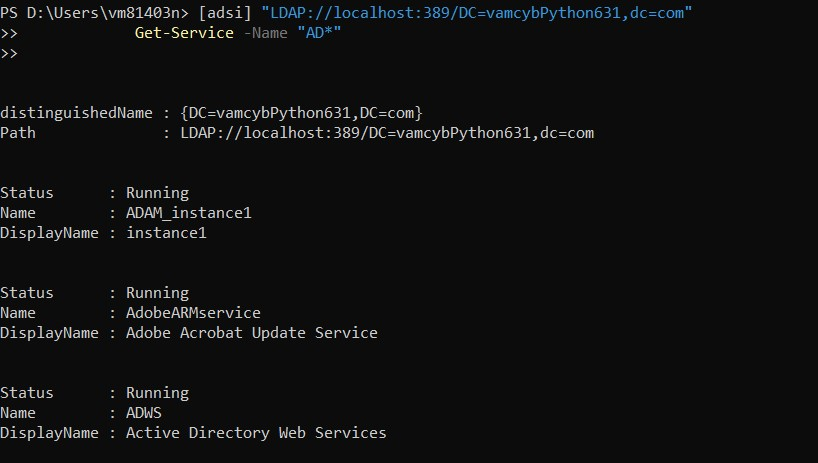
**[Exercise II: Test PowerShell on AD LDS]**

1. Show the instance that we built earlier, and show the services that were run by the AD service. You should see ADAM instance is running and ADWS service is provided.

**[adsi] "LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com"**

**Get-Service -Name "AD\*"**

1. Paste a screenshot of your results above.



1. Review the container, **domain**.

**$domain=[adsi] "LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com"**

**$domain | format-list \***

1. Add user information to the directory.

**Add Use$user = $domain.Create("User", "cn=PartitionLab")**

**$user.Put("userPrincipalName", "KenMyer@cyb631.com")**

**$user.Put("displayName", "Ken Myer")**

**$user.SetInfo()**

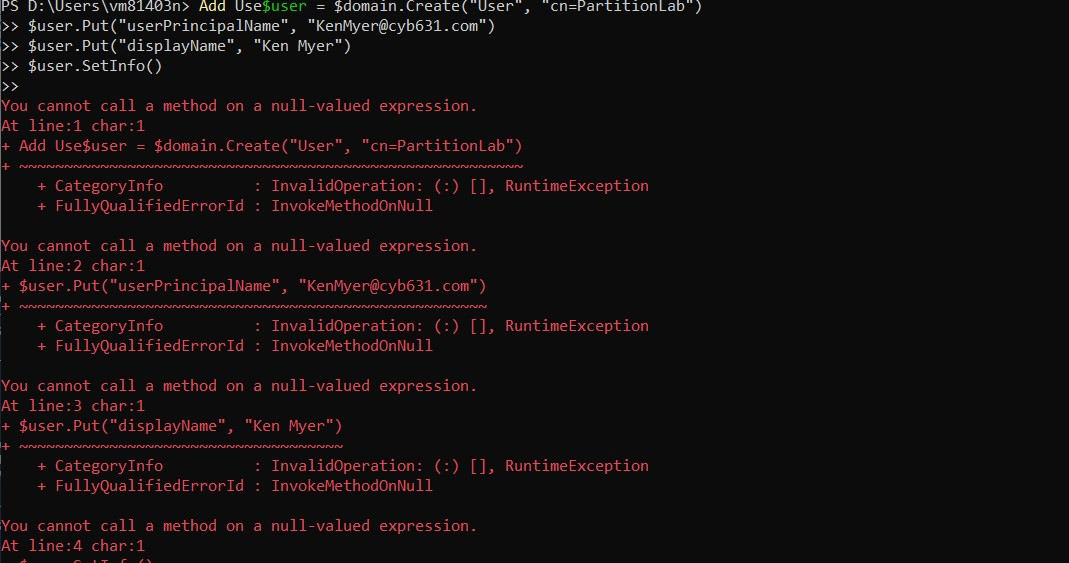
1. Display user information.

**$user.userPrincipalName**

**$user.displayName**

1. Paste a screenshot of your results above.

**No output getting an error that user cannot be generated.**

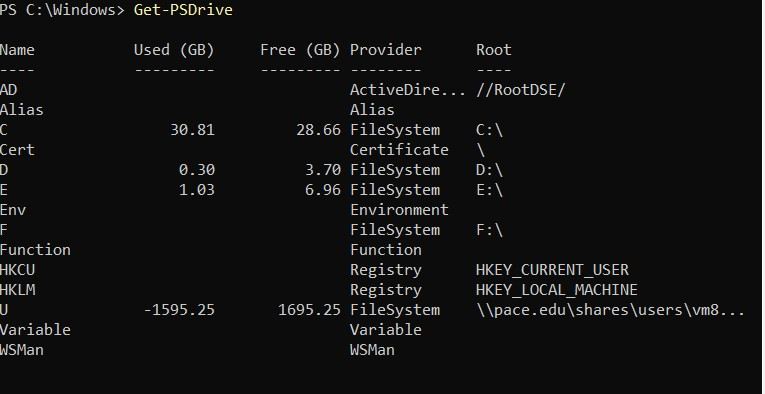


**[Exercise III: Windows Registry]**

1. To see where the registry hive keys are on the drive.

**Get-PSDrive**

1. Paste a screenshot of your results.



1. On the lower left corner, run “regedit” as a Windows command. This will open up Windows registry editor. This will show all of registry hive keys. Explore the registry and review what they are. Please list two of them here

**HKEY\_USERS**

**HKEY\_LOCAL\_MACHINE**.

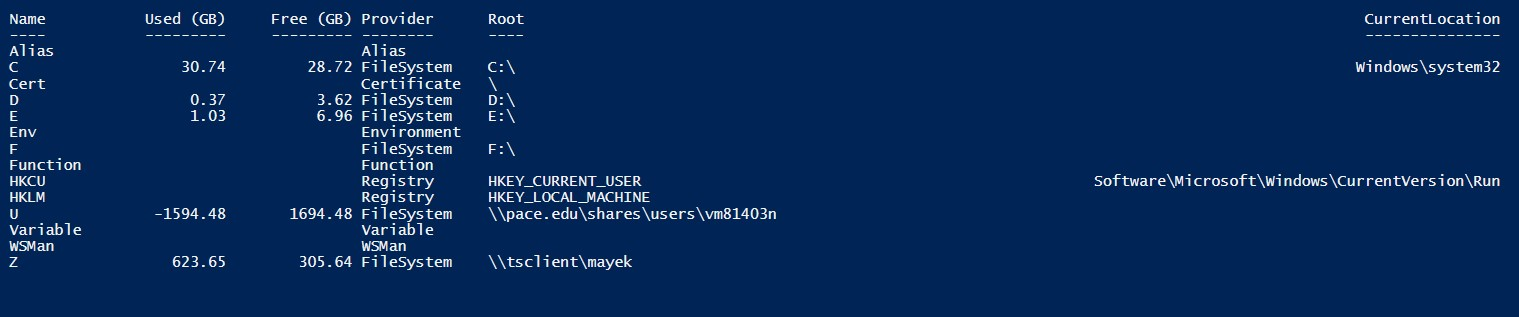
1. Close regedit. We will now use PowerShell to retrieve registry. Under PowerShell ISE,

**Set-Location HKCU:\Software\Microsoft\Windows\CurrentVersion\Run**

**$item=Get-ItemProperty .**

**$item**

1. This will show hive keys under current users (HKCU). Paste your results here.



**[Exercise IV: Windows Management Instrumentation]**

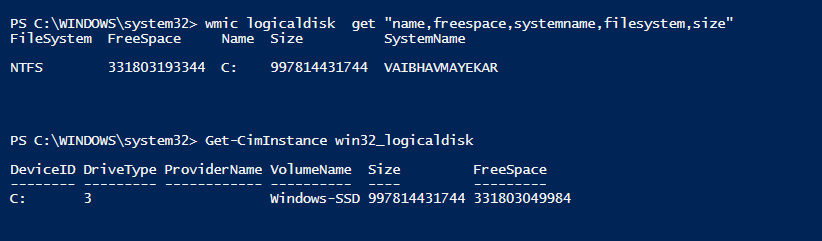
1. Let access logical disk information using WMIC.

**wmic logicaldisk get "name,freespace,systemname,filesystem,size"**

1. Now, let access the same information using CIM cmdlet. Get-CimIntance obtains an CIM instance of a class, in this case, win32\_logicaldisk

**Get-CimInstance win32\_logicaldisk**

1. Paste a screenshot of the results from above.



1. To see all of the classes available,

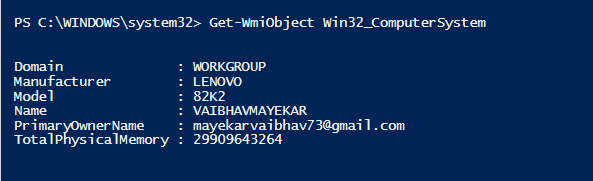
**Get-CimClass -ClassName \***

**Win32\_ACE**

1. Let us try a WMI cmdlet to obtain computer information.

**Get-WmiObject Win32\_ComputerSystem**

1. Paste the results of your results above.



1. You can use the CIM cmdlet to obtain similar information. CIM cmdlet is more portable since it is across platform.

**Get-CimInstance CIM\_ComputerSystem**

1. Try another CIM cmdlet to receive process information.

**Get-CimInstance Win32\_Process | Select Name,ProcessId,ThreadCount**

1. You can use the WQL language, similar to SQL, to access the information.

**Get-CimInstance -query "select \* from win32\_service where StartMode='auto'"**

1. Explain what the results of the above command mean.

**The command retrieves a list of Windows services that are set to start automatically when the computer boots up and also the results include information about these services, such as their names, display names, current states**

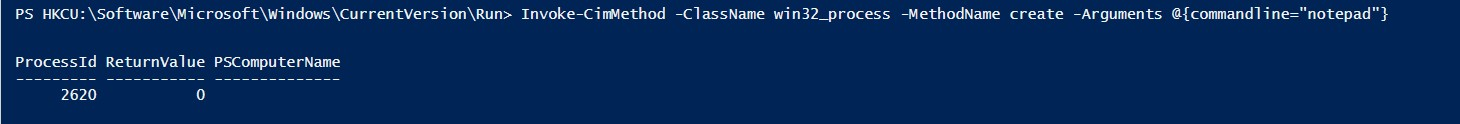
1. Invoke a method supported by a WMI or CIM class. The following invoke a notepad process.

**Invoke-CimMethod -ClassName win32\_process -MethodName create -Arguments @{commandline="notepad"}**

1. What happened after you run the script above?

**A new instance of the Notepad process should be created, and the Notepad application should open on your computer.**

Paste a screenshot of your results from PowerShell ISE.



**[Exercise V: Configure Windows Firewall]**

1. PowerShell has a NetSecurity Module for configuring Firewall and IPSec.

**Get-Command -module NetSecurity**

1. To enable Firewall to all profiles.

**Set-NetFirewallProfile -All -Enabled True**

1. Let us try several firewall rules. Block access to web servers inside this host.

**New-NetFirewallRule -DisplayName "HTTP-Inbound" -Profile Any -Direction Inbound -Action Block -Protocol tcp -LocalPort @('80','443')**

1. Use a browser to see if you can access servers on the Internet. You should be able to do so at this point. Now, block access from external web servers to this host.

**New-NetFirewallRule -DisplayName "HTTP-outbound" -Profile Any -Direction outbound -Action Allow -Protocol tcp -RemotePort @('80','443')**

1. Now, try use a web browser to see if you are able to access external web servers. Are you still be able to do that now? **No** Why or Why not?

**When you block outbound HTTP traffic, you are preventing your computer from sending HTTP requests to external web servers. This means that you will not be able to load any web pages, or use any other applications that rely on HTTP traffic, such as email clients or social media apps.**

1. Now, let us open our Internet web access. We will have to modify the rule that we set earlier.

**Set-NetFirewallRule -DisplayName "HTTP-outbound" -Profile Any -Direction outbound -Action Allow -Protocol tcp -RemotePort @('80','443')**

1. You can also remove firewall rules.

**Remove-NetFirewallRule -Action Block**

**[Exercise VI: Develop a PowerShell to Automatically Configure Windows Firewall]**

1. Please write a PowerShell script to enable Windows Firewall and include at least 2 new rules. These rules are:

Rule 1: block any attempts to access SSH servers from outside to inside of your Active Directory domain.

Rules 2: block any attempts to access DNS servers from outside to inside of your Active Directory domain.

1. Paste a screenshot of the script.

**# Enable Windows Firewall**

**Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True**

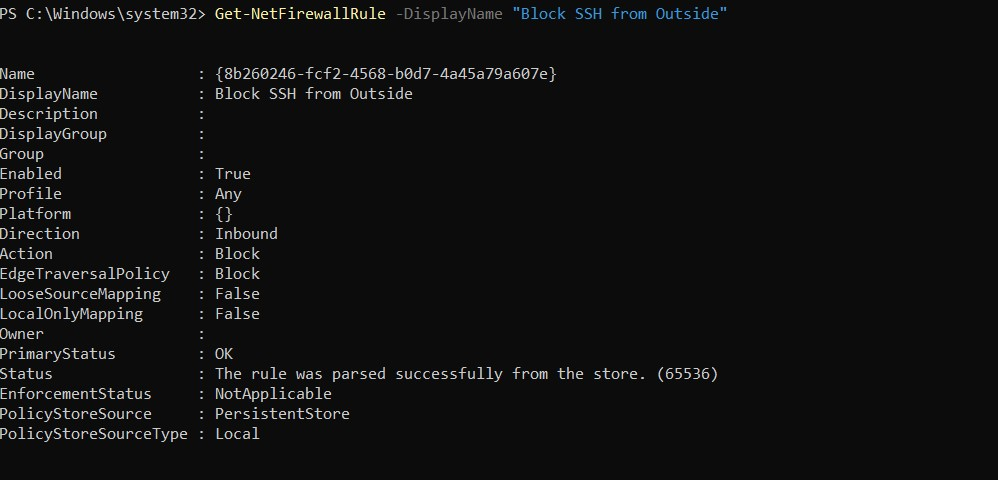
**# Create SSH blocking rule**

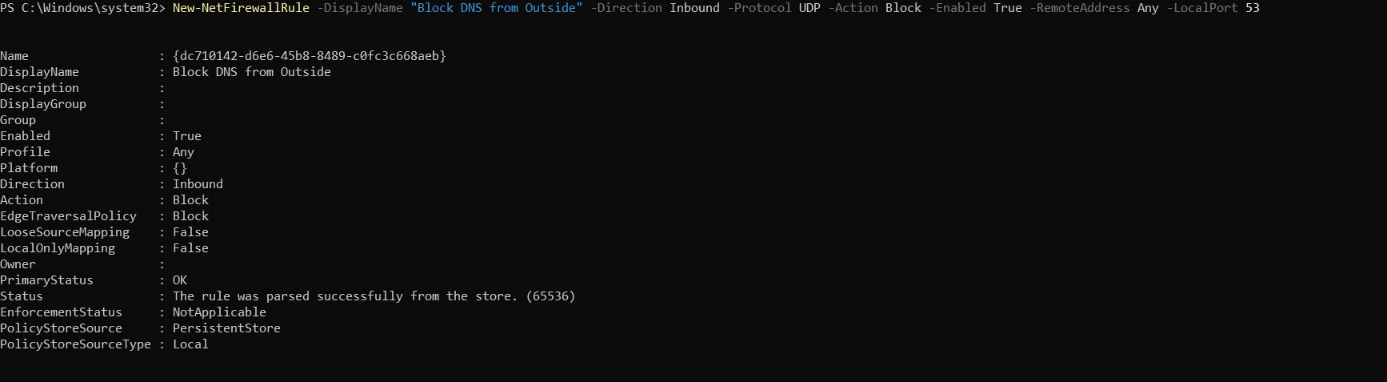
**New-NetFirewallRule -DisplayName "Block SSH from Outside" -Direction Inbound -Protocol TCP -Action Block -Enabled True -RemoteAddress Any -LocalPort 22**

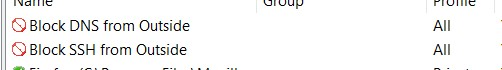
**# Create DNS blocking rule**

**New-NetFirewallRule -DisplayName "Block DNS from Outside" -Direction Inbound -Protocol UDP -Action Block -Enabled True -RemoteAddress Any -LocalPort 53**

1. Paste a screenshot of results from running the script (the results can be long, and you only must show the beginning part of the results).







1. Briefly discuss the advantages of using PowerShell scripts to configure Windows Firewall.

**Using PowerShell scripts to configure Windows Firewall offers the benefits of automation, ensuring consistent rule enforcement across multiple systems, scalability for efficient management, customization to tailor rules to specific needs, version control for tracking changes, logging and reporting for monitoring and auditing, and seamless integration with other tools and systems, all of which collectively enhance network security and simplify rule management.**

**[Lab and Class Reflection]**

Please reflect on what you have completed and learned (required).

1. What do you like about this lab? Interesting
2. What are the challenges that you have encountered in this lab? Firewall part was Difficult.
3. Do you have additional suggestions for this class to help you learn better? No
4. If you are using Windows virtual machine on the server, please turn off your virtual machines once you are done.