CYB631 Lab 3: Managing and Hardening Hosts

Vijaysingh Puwar

Seidenberg School of Computer Science and Information Systems, Pace University

CYB 631: Automating Information Security with Python and Shell Scripting

Professor Alex Tsekhansky

September 2025

Contents

ABSTRACT

This lab focused on managing and hardening Windows hosts through a series of structured exercises using PowerShell and built-in Windows tools. The exercises introduced Active Directory Lightweight Directory Services (AD LDS), Windows Registry, Windows Management Instrumentation (WMI), Common Information Model (CIM), and Windows Firewall. Through these tasks, I gained hands-on experience with installing and configuring directory services, exploring and querying the registry, and retrieving system information using both WMI and CIM. Additionally, I practiced creating, modifying, and testing firewall rules directly through PowerShell, culminating in the development of an automated script to enforce security baselines.

The lab reinforced the importance of automation in system administration, showing how PowerShell scripts provide scalability, repeatability, and consistency when configuring security policies. Challenges such as running commands in the correct context and troubleshooting execution errors highlighted the need for careful attention to permissions and environments. Overall, the lab provided valuable practical skills that are directly applicable to real-world cybersecurity practices, emphasizing both manual configuration knowledge and the efficiency of automation.

EXERCISE: ENVIRONMENT — STARTING WITH POWERSHELL ISE

Step 1

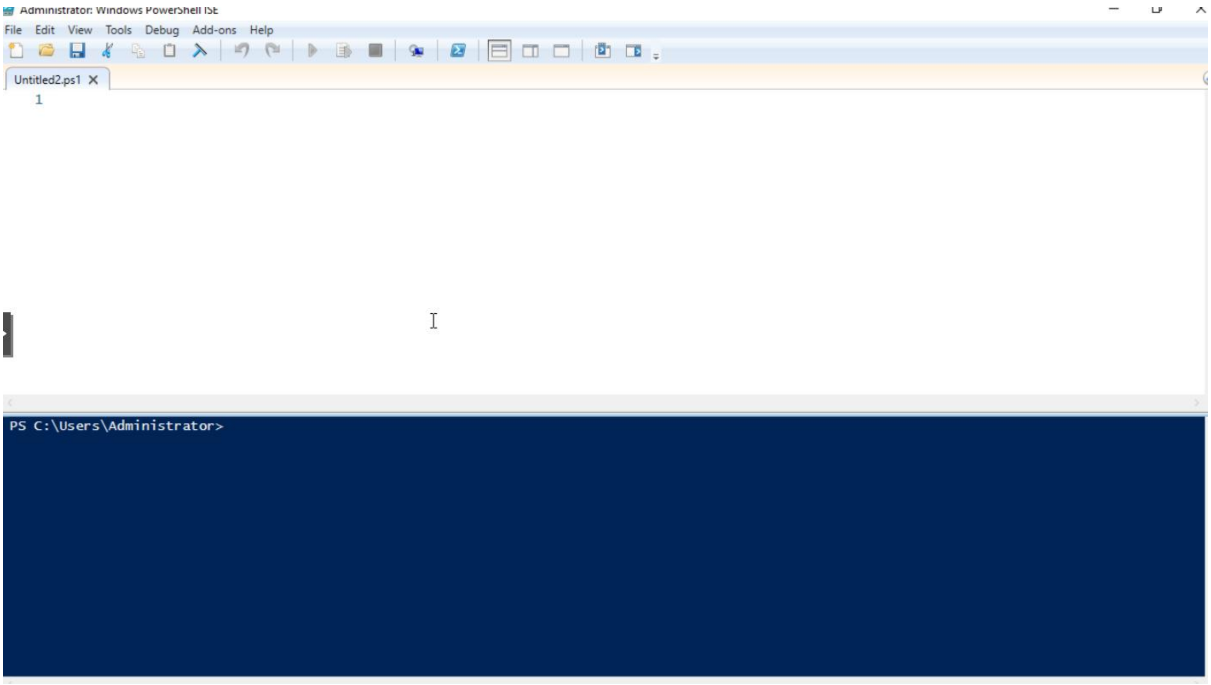
It is recommended that you use the Windows Server VM on VMware Horizon Desktop provided by this class because:

1. Active Directory service used in Exercise I will require Windows Server. You cannot run Exercise I on Windows 10 or Windows 11 OS.
2. Windows administration configuration, such as Active Directory or Registry, might interfere with settings on your personal computer needed for daily work.

Please refer to the *PaceLabHowTo* document in Week 1 for instructions to access the Windows Server VM.

Step 2

Windows 10 environment is needed for the lab. Launch Windows PowerShell ISE. Click Start->Run, and then type PowerShell ISE.


Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled2.ps1

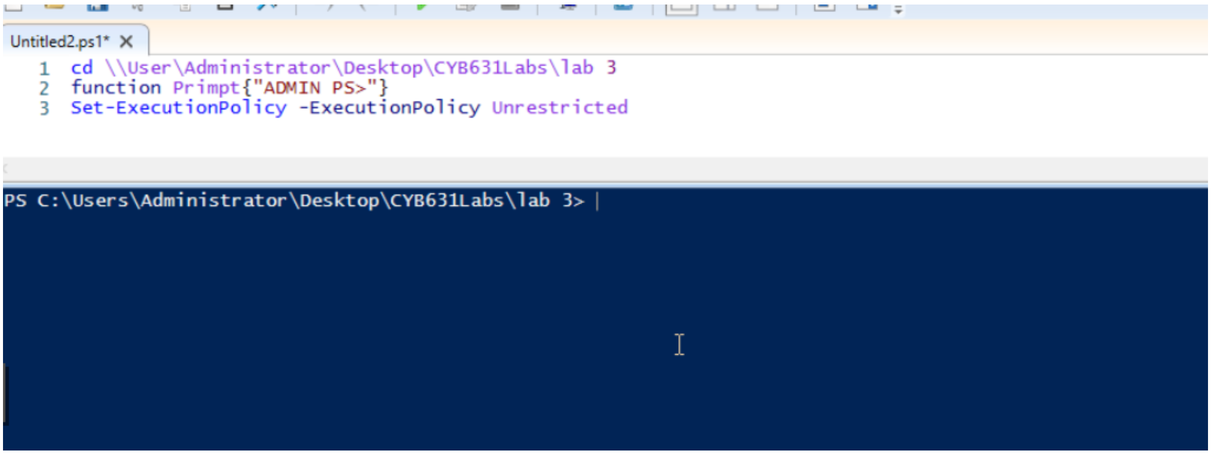
```
PS C:\Users\Administrator>
```

Figure 1

Step 3

Open a new PowerShell ISE session and run it as **administrator**. Navigate to the directory where your script is located. Then, shorten the command prompt and change the execution policy to either **RemoteSigned** or **Unrestricted**.

- *RemoteSigned* requires that all scripts downloaded from the Internet be digitally signed.
- *Unrestricted* allows execution but prompts users for confirmation.



```
cd \\User\Administrator\Desktop\CYB631Labs\lab 3
function Primpt{"ADMIN PS>"}
Set-ExecutionPolicy -ExecutionPolicy Unrestricted
```

```
PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3>
```

Figure 2

EXERCISE I: INSTALL ACTIVE DIRECTORY LIGHTWEIGHT SERVICE

Step 4

Install Active Directory Lightweight Services (AD LDS), which is a lightweight version of Active Directory.

Step 5

In the search box next to the Windows Start menu, type Server Manager. Right-click and choose Run as administrator to open Server Manager. You should then see the Dashboard window.

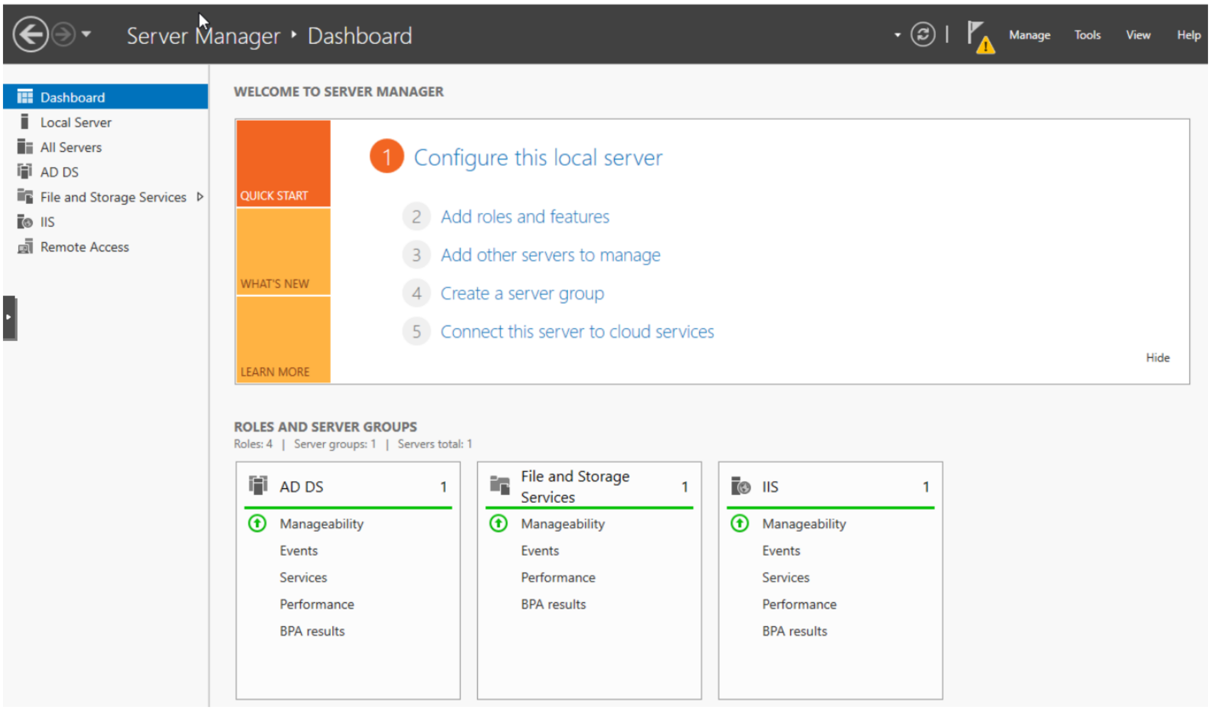
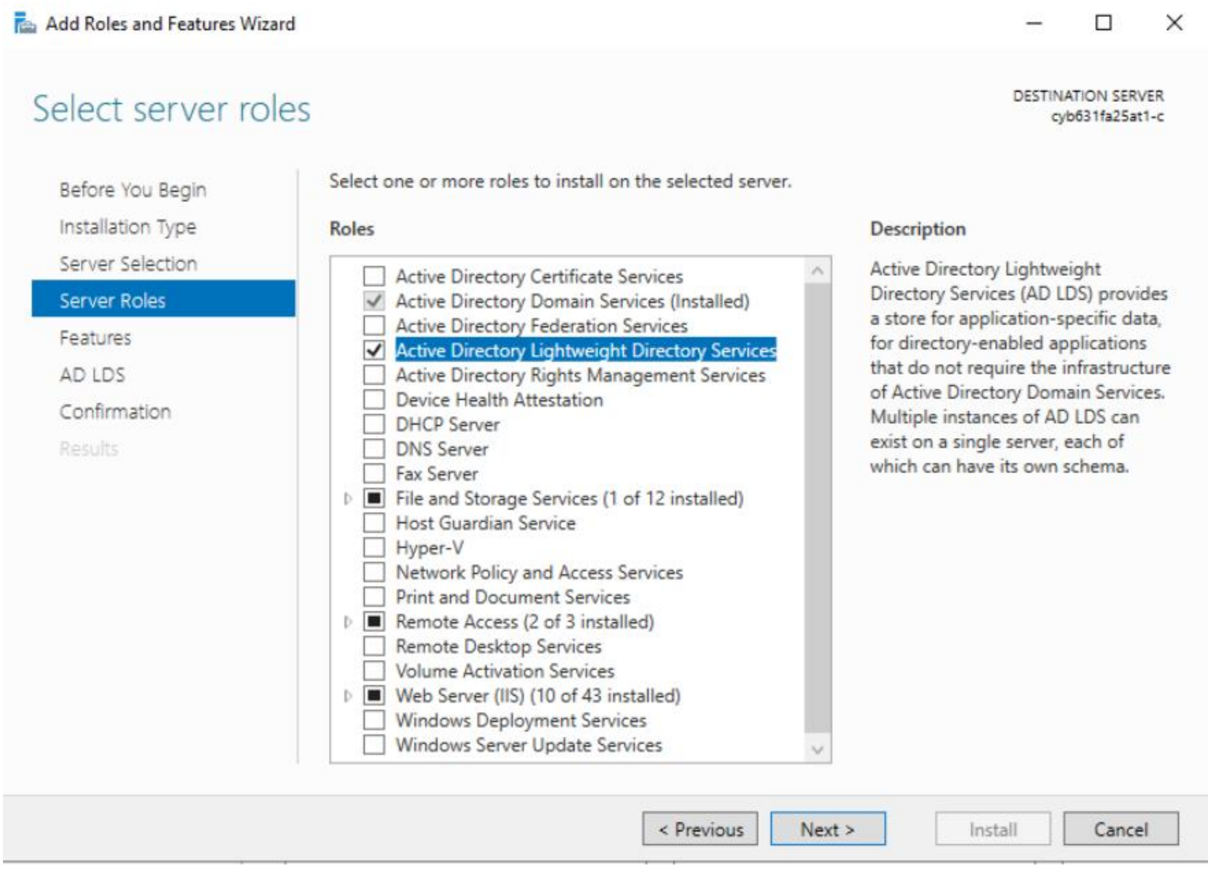


Figure 3

Step 6

In Server Manager, select Add Roles and Features, and check Active Directory Lightweight Directory Service (AD LDS). Follow the prompts to install it. You may need to click Add Features → Next → Install.



Add Roles and Features Wizard

Select server roles

DESTINATION SERVER
cyb631fa25at1-c

Select one or more roles to install on the selected server.

Roles

- □ Active Directory Certificate Services
- ☑ Active Directory Domain Services (Installed)
- □ Active Directory Federation Services
- ☑ Active Directory Lightweight Directory Services
- □ Active Directory Rights Management Services
- □ Device Health Attestation
- □ DHCP Server
- □ DNS Server
- □ Fax Server
- ■ File and Storage Services (1 of 12 installed)
- □ Host Guardian Service
- □ Hyper-V
- □ Network Policy and Access Services
- □ Print and Document Services
- ■ Remote Access (2 of 3 installed)
- □ Remote Desktop Services
- □ Volume Activation Services
- ■ Web Server (IIS) (10 of 43 installed)
- □ Windows Deployment Services
- □ Windows Server Update Services

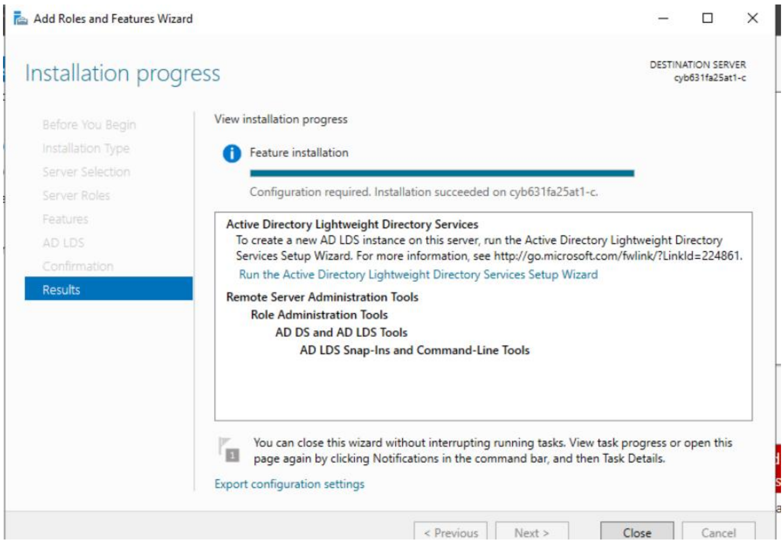
Description

Active Directory Lightweight Directory Services (AD LDS) provides a store for application-specific data, for directory-enabled applications that do not require the infrastructure of Active Directory Domain Services. Multiple instances of AD LDS can exist on a single server, each of which can have its own schema.

Figure 4

Step 7

After AD LDS is installed, run the **Active Directory Lightweight Directory Service Setup Wizard**. You can do this by clicking the link provided in the installation success message.



Add Roles and Features Wizard

Installation progress

DESTINATION SERVER
cyb631fa25at1-c

View installation progress

Feature installation

Configuration required. Installation succeeded on cyb631fa25at1-c.

Active Directory Lightweight Directory Services
To create a new AD LDS instance on this server, run the Active Directory Lightweight Directory Services Setup Wizard. For more information, see http://go.microsoft.com/fwlink/?LinkId=224861.
Run the Active Directory Lightweight Directory Services Setup Wizard

Remote Server Administration Tools
Role Administration Tools
AD DS and AD LDS Tools
AD LDS Snap-Ins and Command-Line Tools

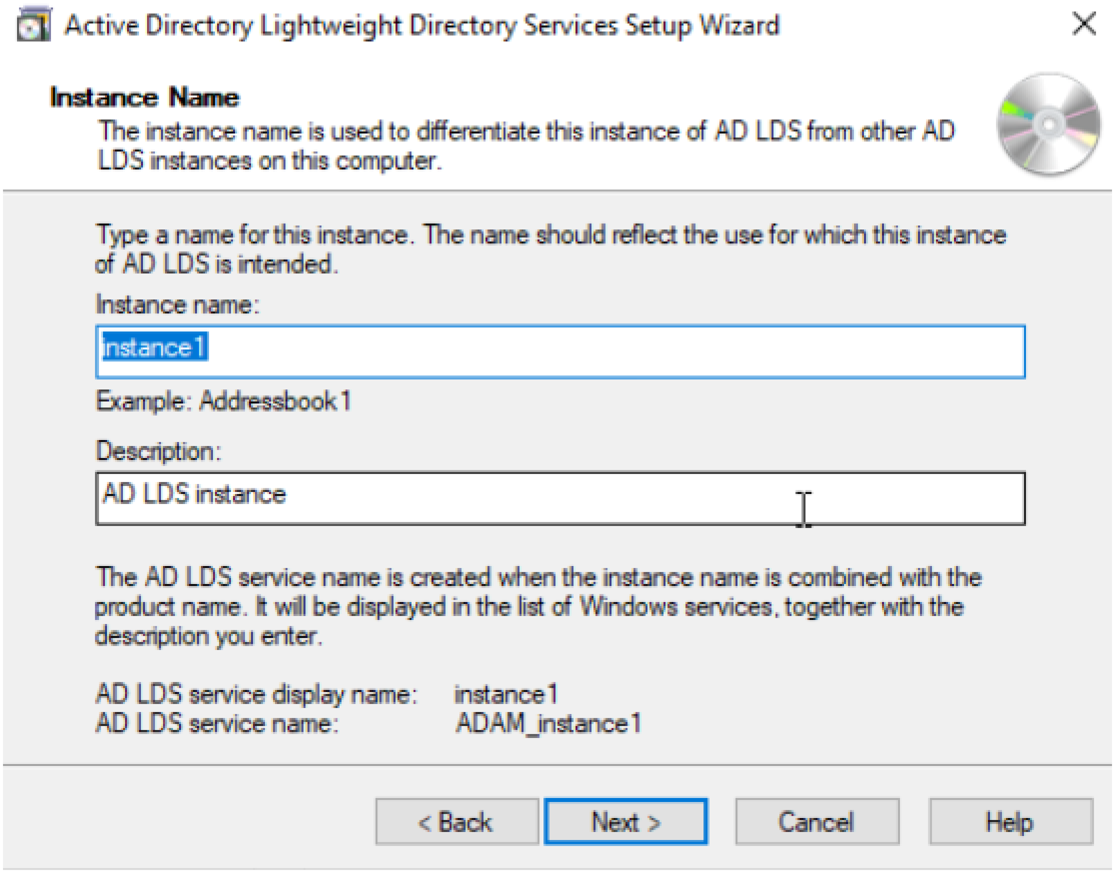
You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.

Export configuration settings

Figure 5

Step 8

Select Install a unique instance. Use the default values for Instance Name and Ports.



Active Directory Lightweight Directory Services Setup Wizard

Instance Name

The instance name is used to differentiate this instance of AD LDS from other AD LDS instances on this computer.

Type a name for this instance. The name should reflect the use for which this instance of AD LDS is intended.

Instance name: instance1

Example: Addressbook1

Description: AD LDS instance

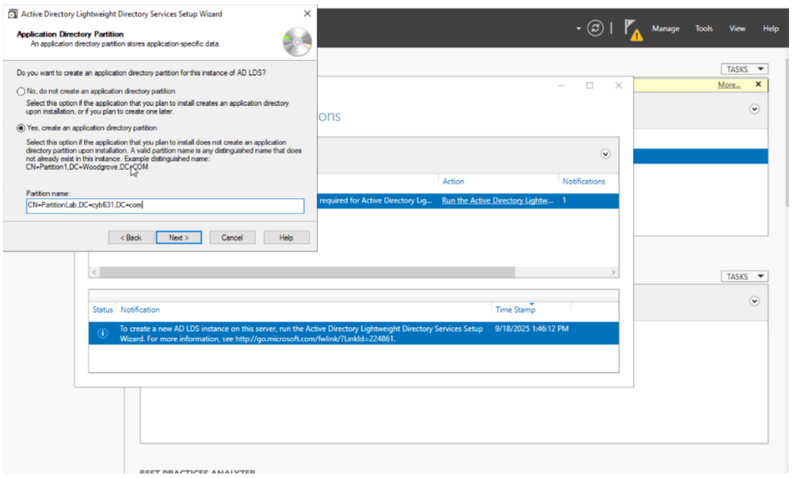
The AD LDS service name is created when the instance name is combined with the product name. It will be displayed in the list of Windows services, together with the description you enter.

AD LDS service display name: instance1
AD LDS service name: ADAM_instance1

Figure 6

Step 9

Select **Yes, create an application directory partition**. Provide a unique partition name, for example:



Active Directory Lightweight Directory Services Setup Wizard

Application Directory Partition

An application directory partition stores application-specific data.

Do you want to create an application directory partition for this instance of AD LDS?

○ No, do not create an application directory partition

Select this option if the application that you plan to install creates an application directory upon installation, or if you plan to create one later.

● Yes, create an application directory partition

Select this option if the application that you plan to install does not create an application directory partition upon installation. A valid partition name is any distinguished name that does not already exist in this instance. Example distinguished name: CN=Partition1,DC=Woodgrove,DC=COM

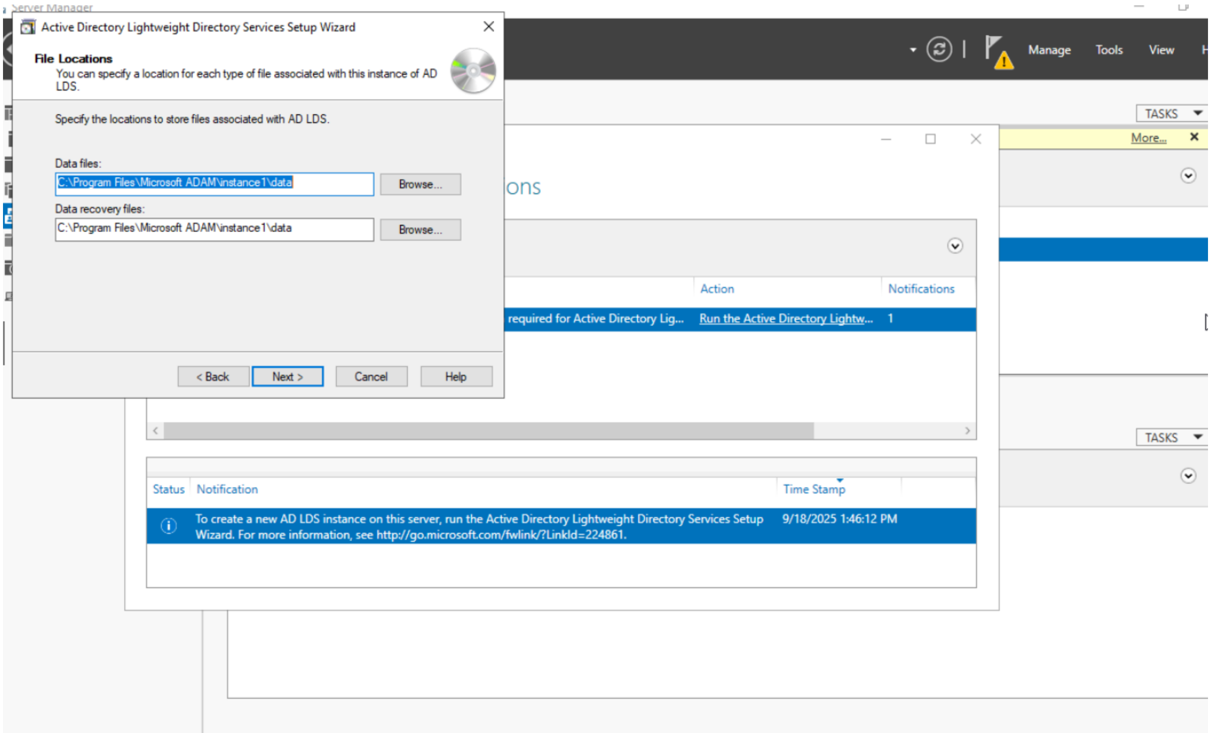
Partition name: CN=PartitionLab,DC=cyb631,DC=com

Status | Notification | Time Stamp
To create a new AD LDS instance on this server, run the Active Directory Lightweight Directory Services Setup Wizard. For more information, see http://go.microsoft.com/fwlink/?LinkId=224861. 9/18/2025 1:46:12 PM

Figure 7

Step 10

Use the default names for file locations.



Active Directory Lightweight Directory Services Setup Wizard

File Locations

You can specify a location for each type of file associated with this instance of AD LDS.

Specify the locations to store files associated with AD LDS.

Data files: C:\Program Files\Microsoft ADAM\instance1\data

Data recovery files: C:\Program Files\Microsoft ADAM\instance1\data

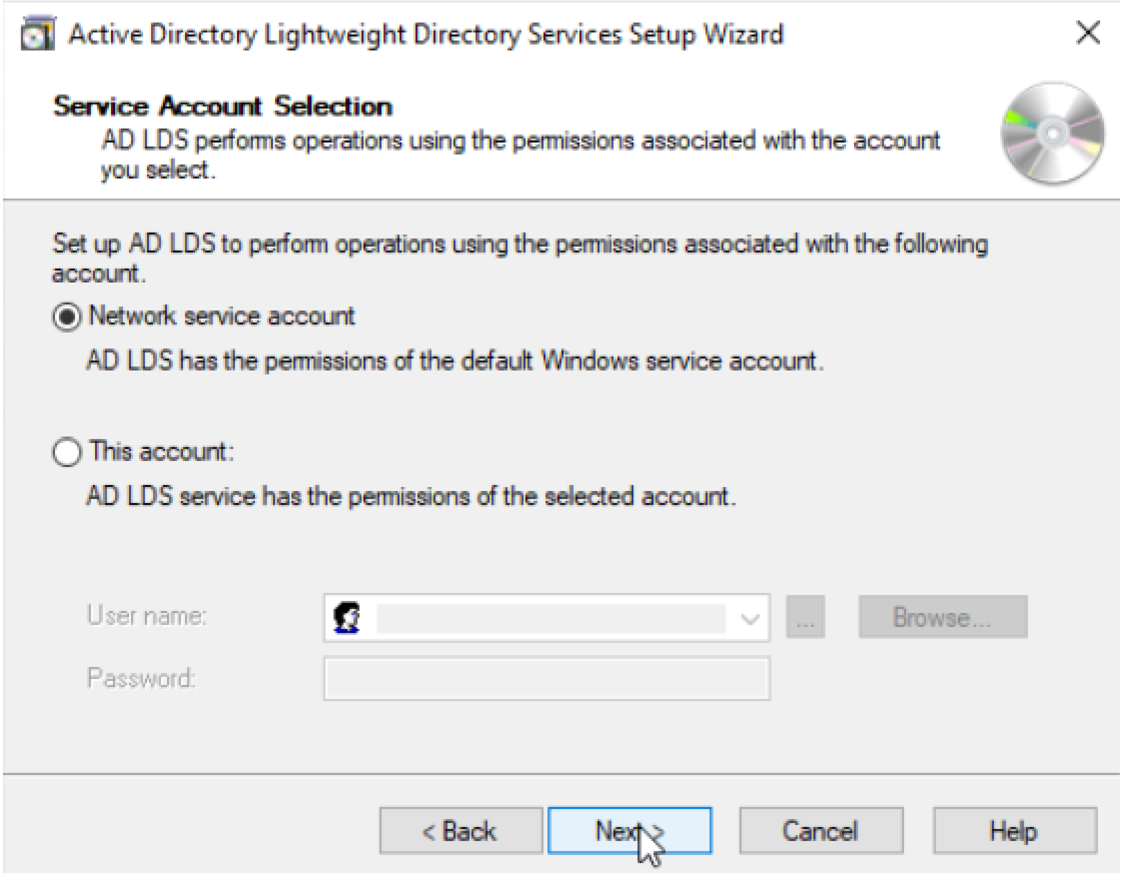
required for Active Directory Lig... | Run the Active Directory Lightw... | 1

To create a new AD LDS instance on this server, run the Active Directory Lightweight Directory Services Setup Wizard. For more information, see http://go.microsoft.com/fwlink/?LinkId=224861. | 9/18/2025 1:46:12 PM

Figure 8

Step 11

For Service Account Selection, choose Network Service account.



Active Directory Lightweight Directory Services Setup Wizard

Service Account Selection

AD LDS performs operations using the permissions associated with the account you select.

Set up AD LDS to perform operations using the permissions associated with the following account.

● Network service account

AD LDS has the permissions of the default Windows service account.

○ This account:

AD LDS service has the permissions of the selected account.

User name:

Password:

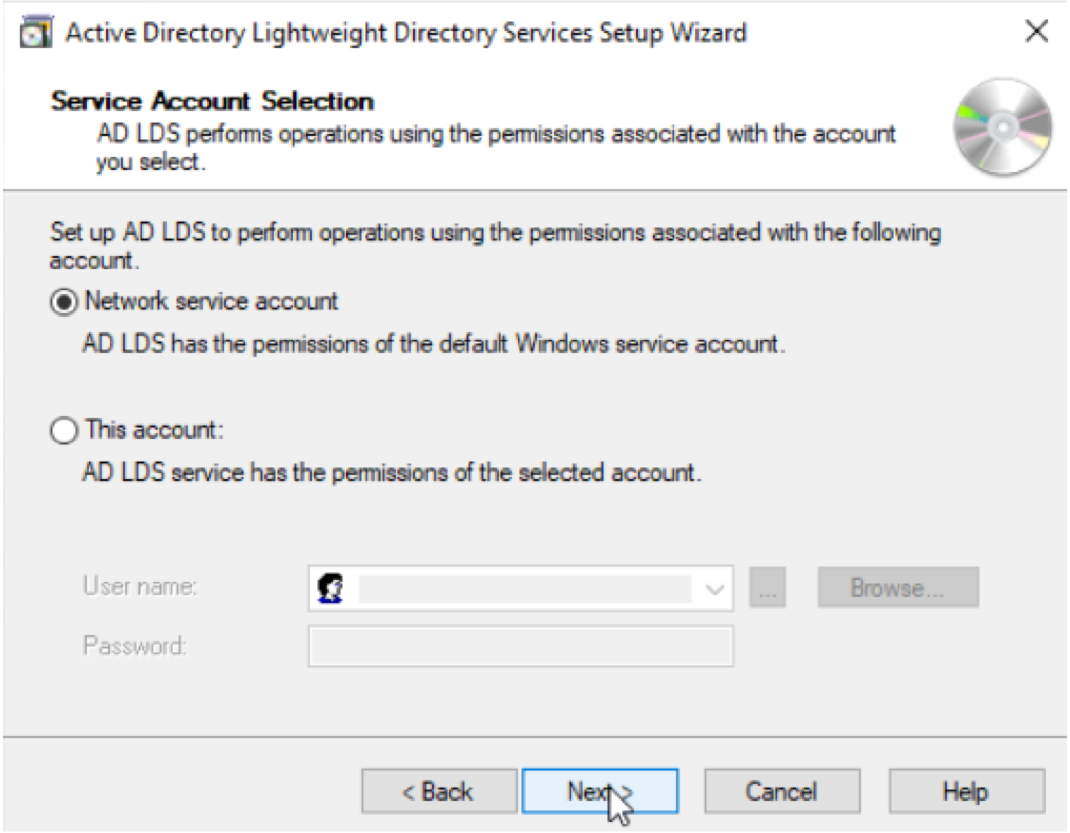
Figure 9

Step 12

For AD LDS Administrators, choose Currently logged-on user.

Step 13

For Importing LDIF Files, select the ones needed for applications. These LDIF files are text-based and represent data/commands for the LDAP instance. For testing, click on MS-User.LDF.



Active Directory Lightweight Directory Services Setup Wizard

Service Account Selection

AD LDS performs operations using the permissions associated with the account you select.

Set up AD LDS to perform operations using the permissions associated with the following account.

● Network service account

AD LDS has the permissions of the default Windows service account.

○ This account:

AD LDS service has the permissions of the selected account.

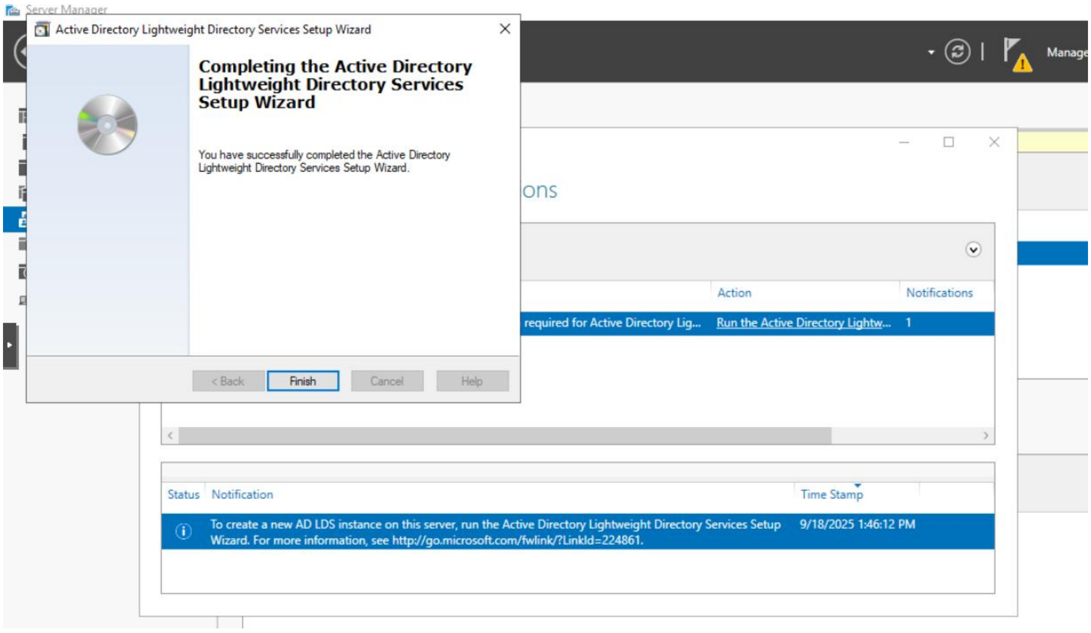
User name:

Password:

Figure 10

Step 14

Complete the **Setup Wizard** by following the instructions. Once finished, you can open **AD LDS** in **Server Manager** to review the details of the instance.



Server Manager

Active Directory Lightweight Directory Services Setup Wizard

Completing the Active Directory Lightweight Directory Services Setup Wizard

You have successfully completed the Active Directory Lightweight Directory Services Setup Wizard.

	Action	Notifications
required for Active Directory Lig...	Run the Active Directory Lightw...	1

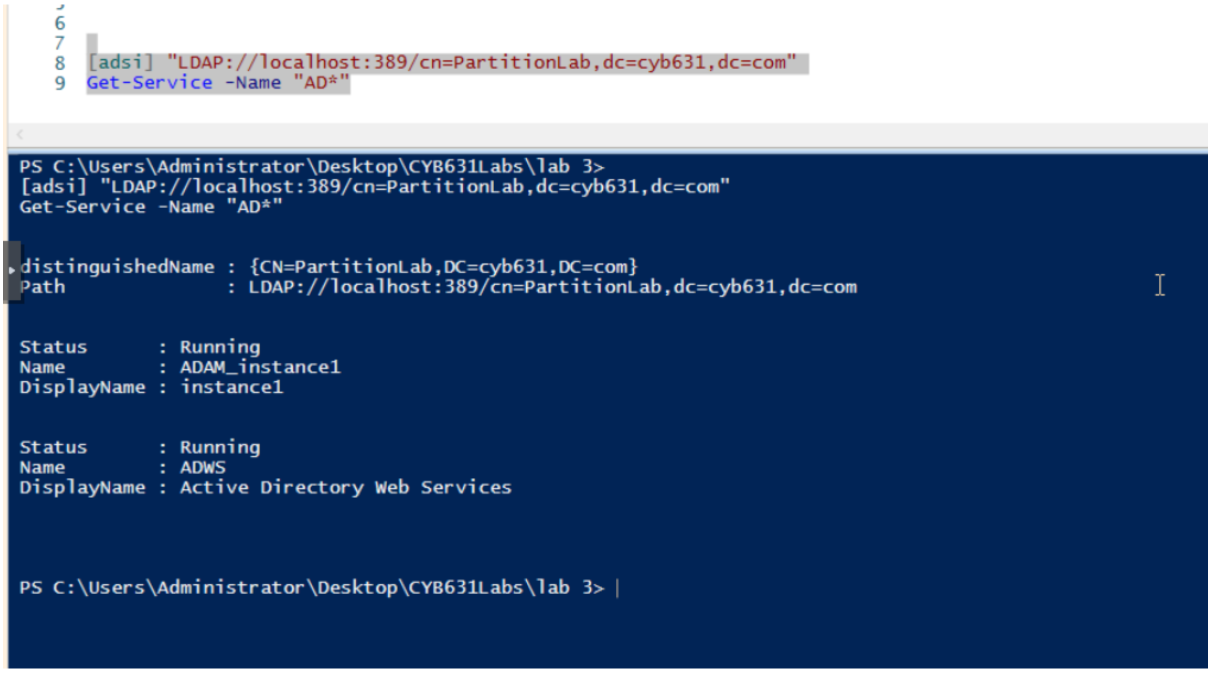
Status	Notification	Time Stamp
ⓘ	To create a new AD LDS instance on this server, run the Active Directory Lightweight Directory Services Setup Wizard. For more information, see http://go.microsoft.com/fwlink/?LinkId=224861.	9/18/2025 1:46:12 PM

Figure 11

EXERCISE II: TEST POWERSHELL ON AD LDS

Step 15

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



```
[adsi] "LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com"
Get-Service -Name "AD*"
```

```
PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3>
[adsi] "LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com"
Get-Service -Name "AD*"


distinguishedName : {CN=PartitionLab,DC=cyb631,DC=com}
Path              : LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com


Status      : Running
Name        : ADAM_instance1
DisplayName : instance1


Status      : Running
Name        : ADWS
DisplayName : Active Directory Web Services



PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> |
```

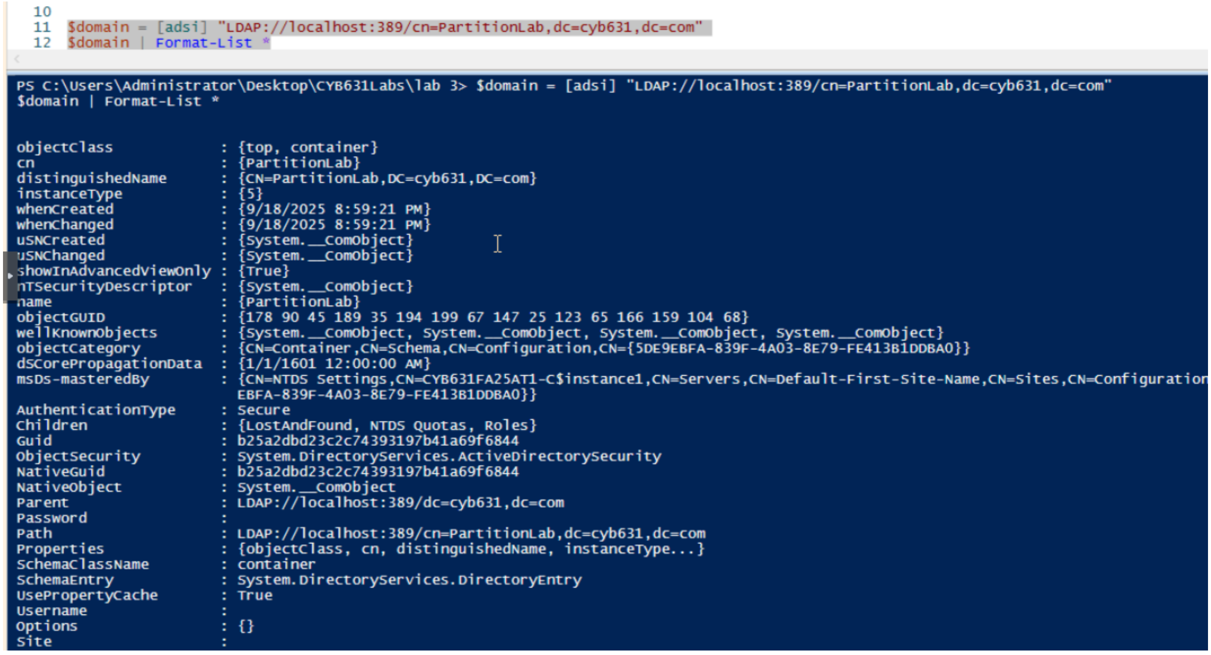
Figure 12

Step 16

The Screenshot is above this step. Output showing ADAM instance and ADWS service running.

Step 17

Review the container and domain.



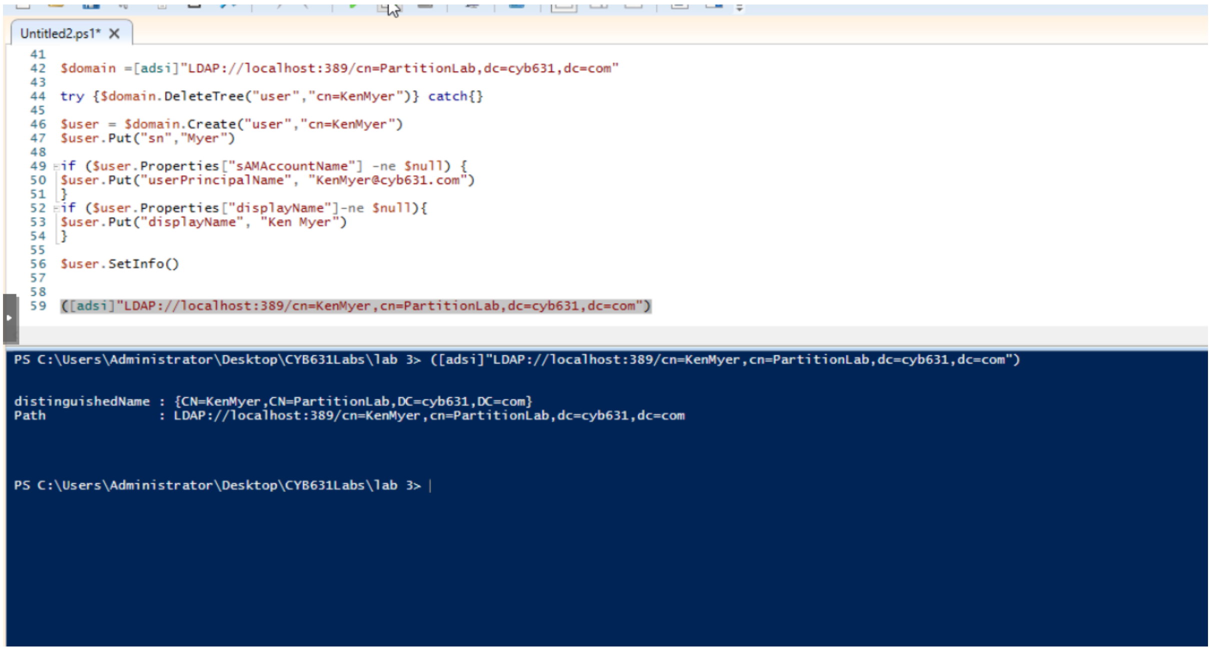
```
$domain = [adsi] "LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com"
$domain | Format-List *
```

```
PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> $domain = [adsi] "LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com"
$domain | Format-List *


objectClass              : {top, container}
cn                       : {PartitionLab}
distinguishedName        : {CN=PartitionLab,DC=cyb631,DC=com}
instanceType             : {5}
whenCreated              : {9/18/2025 8:59:21 PM}
whenChanged              : {9/18/2025 8:59:21 PM}
uSNCreated               : {System.__ComObject}
uSNChanged               : {System.__ComObject}
showInAdvancedViewOnly   : {True}
nTSecurityDescriptor     : {System.__ComObject}
name                     : {PartitionLab}
objectGUID               : {178 90 45 189 35 194 199 67 147 25 123 65 166 159 104 68}
wellKnownObjects         : {System.__ComObject, System.__ComObject, System.__ComObject, System.__ComObject}
objectCategory           : {CN=Container,CN=Schema,CN=Configuration,CN={5DE9EBFA-839F-4A03-8E79-FE413B1DDBA0}}
dSCorePropagationData    : {1/1/1601 12:00:00 AM}
msDS-masteredBy          : {CN=NTDS Settings,CN=CYB631FA25AT1-C$instance1,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration
                           EBFA-839F-4A03-8E79-FE413B1DDBA0}}
AuthenticationType       : Secure
Children                 : {LostAndFound, NTDS Quotas, Roles}
Guid                     : b25a2dbd23c2c74393197b41a69f6844
ObjectSecurity           : System.DirectoryServices.ActiveDirectorySecurity
NativeGuid               : b25a2dbd23c2c74393197b41a69f6844
NativeObject             : System.__ComObject
Parent                   : LDAP://localhost:389/dc=cyb631,dc=com
Password                 :
Path                     : LDAP://localhost:389/cn=PartitionLab,dc=cyb631,dc=com
Properties               : {objectClass, cn, distinguishedName, instanceType...}
SchemaClassName          : container
SchemaEntry              : System.DirectoryServices.DirectoryEntry
UsePropertyCache         : True
Username                 :
Options                  : {}
Site                     :
```

Figure 13

Step 18

Add user information to the directory.

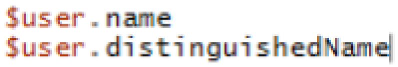


```

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

try {$domain.DeleteTree("user","cn=KenMyer")} catch{}

$user = $domain.Create("user","cn=KenMyer")
$user.Put("sn","Myer")

if ($user.Properties["sAMAccountName"] -ne $null) {
$user.Put("userPrincipalName", "KenMyer@cyb631.com")
}
if ($user.Properties["displayName"]-ne $null){
$user.Put("displayName", "Ken Myer")
}

$user.SetInfo()


([adsi]"LDAP://localhost:389/cn=KenMyer,cn=PartitionLab,dc=cyb631,dc=com")
```

```
PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> ([adsi]"LDAP://localhost:389/cn=KenMyer,cn=PartitionLab,dc=cyb631,dc=com")


distinguishedName : {CN=KenMyer,CN=PartitionLab,DC=cyb631,DC=com}
Path              : LDAP://localhost:389/cn=KenMyer,cn=PartitionLab,dc=cyb631,dc=com


PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> |
```

Step 19

Display user information.

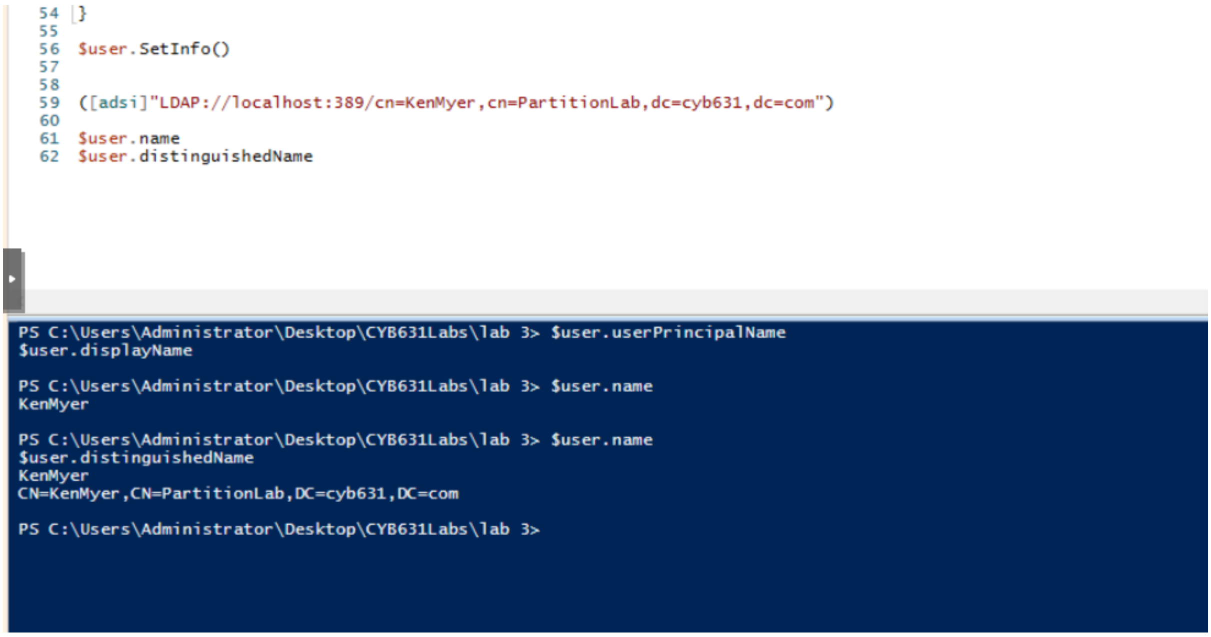


```
$user.name
$user.distinguishedName
```

Figure 14

Step 20

Paste a screenshot of your results above.



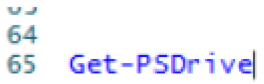
```
}

$user.SetInfo()


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

$user.name
$user.distinguishedName
```

```
PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> $user.userPrincipalName
$user.displayName

PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> $user.name
KenMyer

PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> $user.name
$user.distinguishedName
KenMyer
CN=KenMyer,CN=PartitionLab,DC=cyb631,DC=com

PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3>
```

EXERCISE III: WINDOWS REGISTRY

Step 21

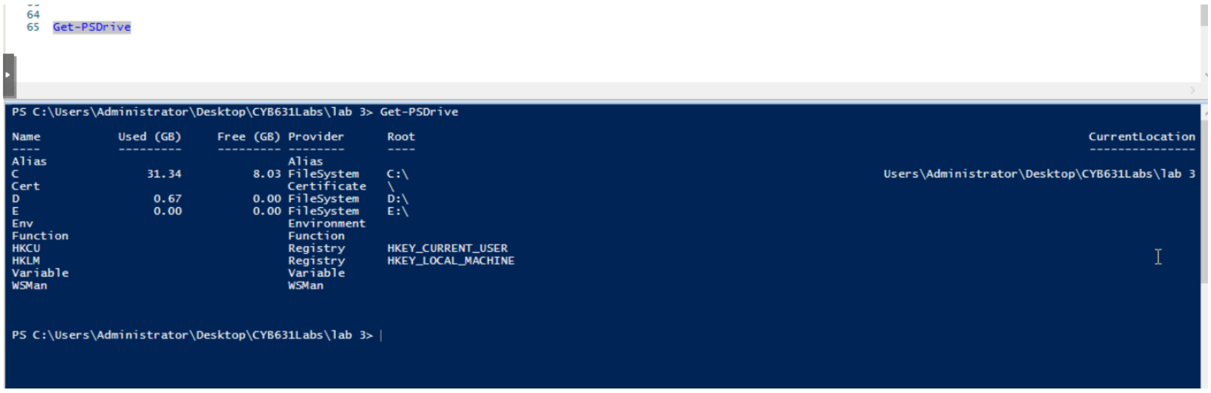
To see where the registry hive keys are located on the drive, run the following command:



```
Get-PSDrive
```

Figure 15

Step 22



```
Get-PSDrive
```

```
PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> Get-PSDrive

Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
Alias                                  Alias
C                  31.34          8.03 FileSystem    C:\                           Users\Administrator\Desktop\CYB631Labs\lab 3
Cert                                   Certificate   \
D                   0.67          0.00 FileSystem    D:\
E                   0.00          0.00 FileSystem    E:\
Env                                    Environment
Function                               Function
HKCU                                   Registry      HKEY_CURRENT_USER
HKLM                                   Registry      HKEY_LOCAL_MACHINE
Variable                               Variable
WSMan                                  WSMan


PS C:\Users\Administrator\Desktop\CYB631Labs\lab 3> |
```

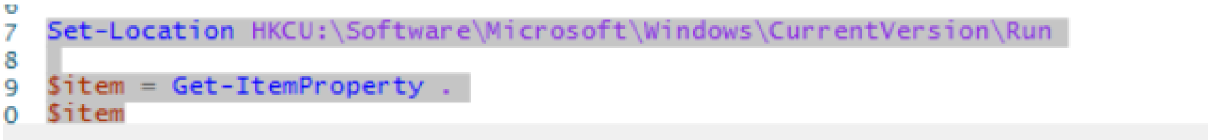
Figure 16

Step 23

On the lower left corner of the Windows desktop, run **regedit** as a command. This will open the Windows Registry Editor and display all registry hive keys. Explore the registry and review their contents.

Step 24

Close **regedit**. Next, use PowerShell to retrieve registry data. Under PowerShell ISE, run:

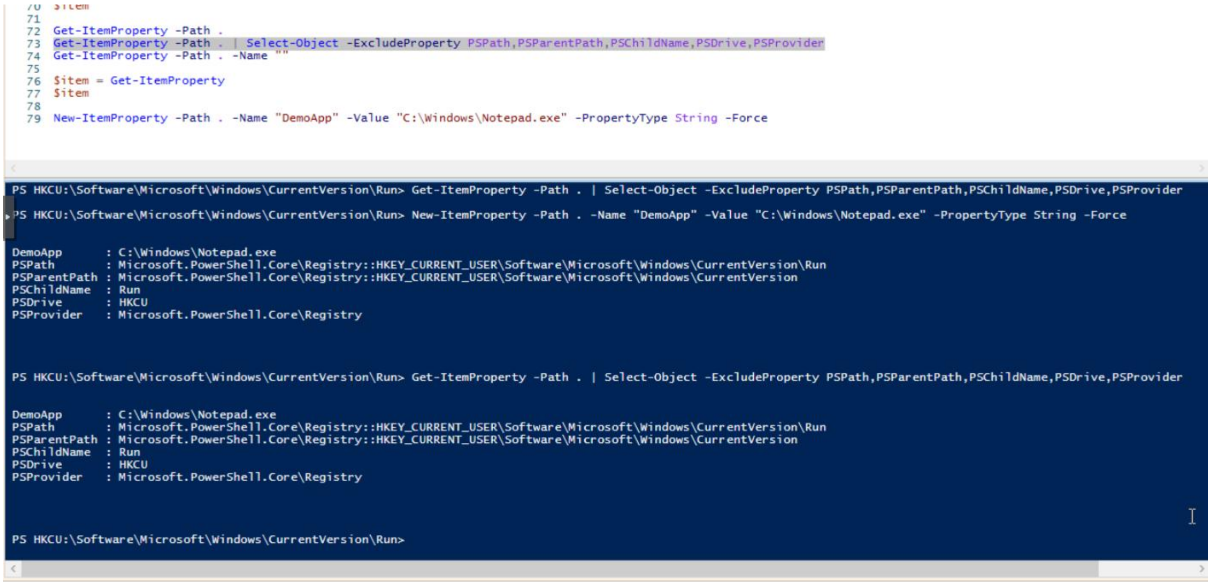


```
Set-Location HKCU:\Software\Microsoft\Windows\CurrentVersion\Run

$item = Get-ItemProperty .
$item
```

Figure 17

Step 25

This command shows registry hive keys under the current user (HKCU). Paste your results here.



```
$item

Get-ItemProperty -Path .
Get-ItemProperty -Path . | Select-Object -ExcludeProperty PSPath,PSParentPath,PSChildName,PSDrive,PSProvider
Get-ItemProperty -Path . -Name ""

$item = Get-ItemProperty
$item

New-ItemProperty -Path . -Name "DemoApp" -Value "C:\Windows\Notepad.exe" -PropertyType String -Force
```

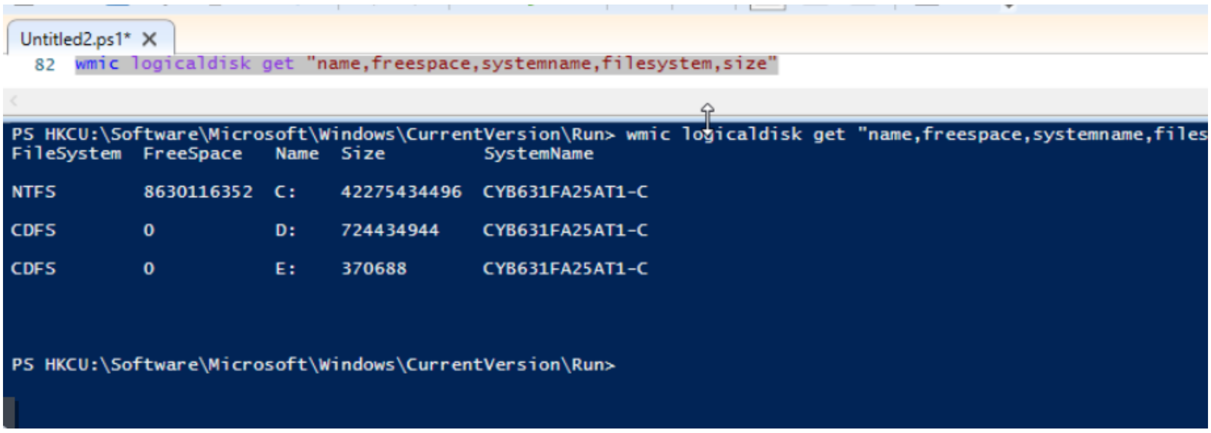
```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-ItemProperty -Path . | Select-Object -ExcludeProperty PSPath,PSParentPath,PSChildName,PSDrive,PSProvider

PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-ItemProperty -Path . -Name "DemoApp" -Value "C:\Windows\Notepad.exe" -PropertyType String -Force


DemoApp      : C:\Windows\Notepad.exe
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion
PSChildName  : Run
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry



PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-ItemProperty -Path . | Select-Object -ExcludeProperty PSPath,PSParentPath,PSChildName,PSDrive,PSProvider


DemoApp      : C:\Windows\Notepad.exe
PSPath       : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion
PSChildName  : Run
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry



PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 18

EXERCISE IV: WINDOWS MANAGEMENT INSTRUMENTATION

Step 26

Access logical disk information using **WMIC**



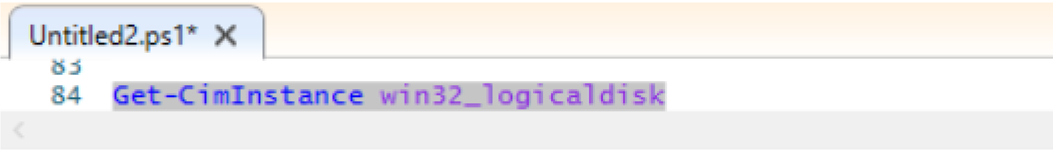
```
wmic logicaldisk get "name,freespace,systemname,filesystem,size"
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> wmic logicaldisk get "name,freespace,systemname,files
FileSystem  FreeSpace    Name  Size         SystemName

NTFS        8630116352   C:    42275434496  CYB631FA25AT1-C

CDFS        0            D:    724434944    CYB631FA25AT1-C

CDFS        0            E:    370688       CYB631FA25AT1-C


PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 19

Step 27

Access the same information using the **CIM cmdlet**. Get-CimInstance obtains a CIM instance of a class—in this case, **Win32_LogicalDisk**:

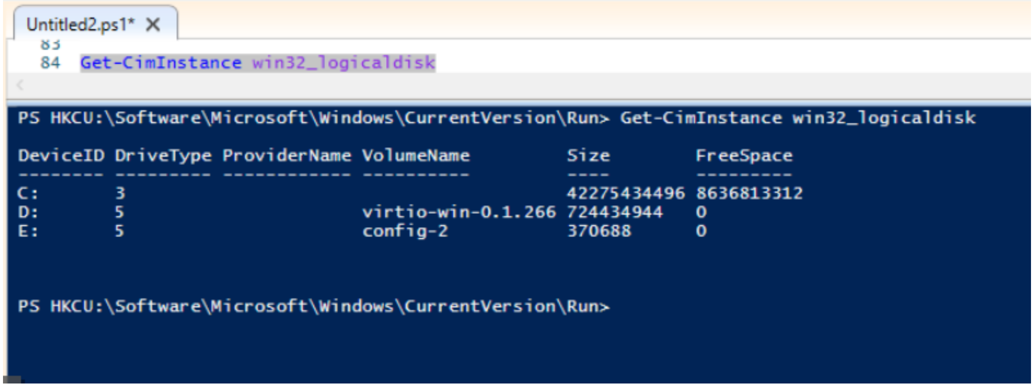


```
Get-CimInstance win32_logicaldisk
```

Figure 20

Step 28

Screenshot of the results from above.



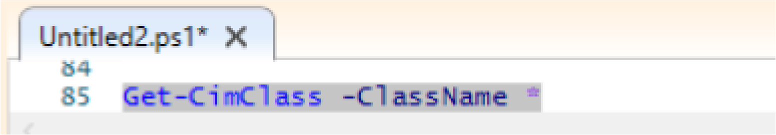
```
Get-CimInstance win32_logicaldisk
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-CimInstance win32_logicaldisk

DeviceID DriveType ProviderName VolumeName         Size        FreeSpace
-------- --------- ------------ ----------         ----        ---------
C:       3                                         42275434496 8636813312
D:       5                      virtio-win-0.1.266 724434944   0
E:       5                      config-2           370688      0


PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 21

Step 29

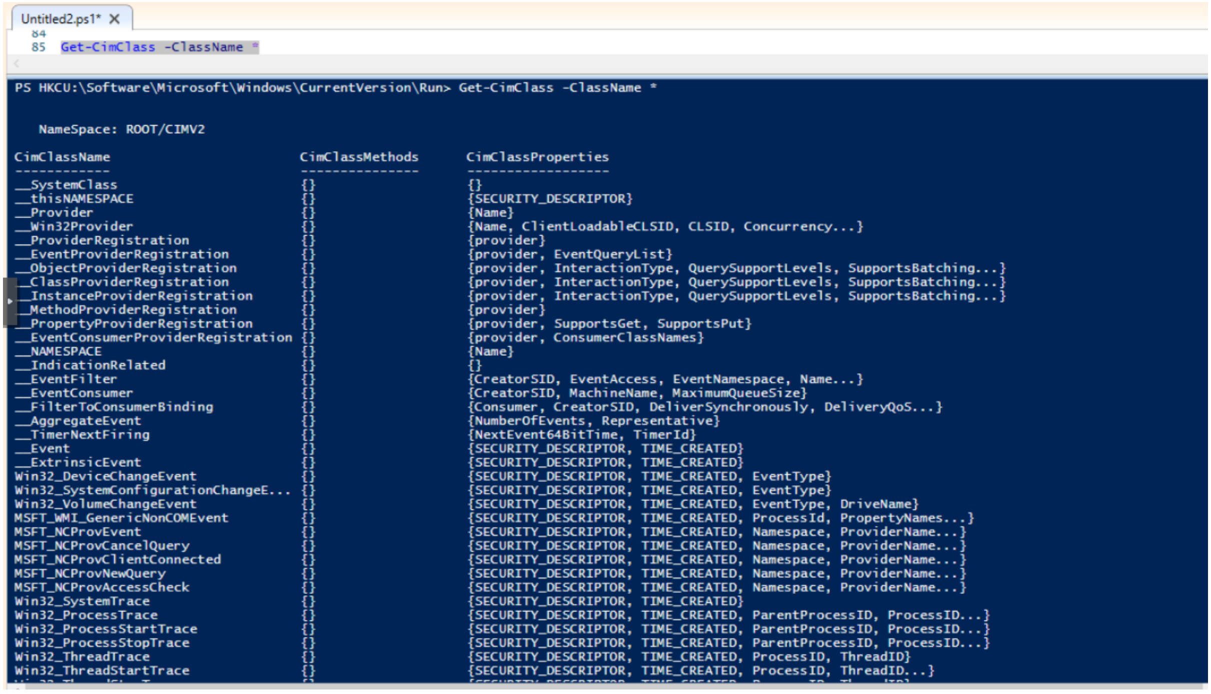
List all available CIM classes:



```
Get-CimClass -ClassName *
```

Figure 22

List of CIM classes available.



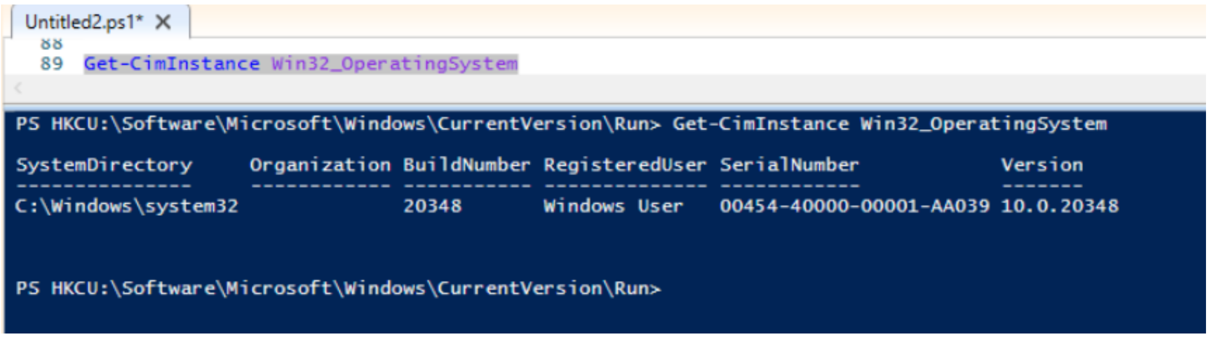
```
Get-CimClass -ClassName *
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-CimClass -ClassName *


   NameSpace: ROOT/CIMV2

CimClassName                        CimClassMethods      CimClassProperties
------------                        ---------------      ------------------
__SystemClass                       {}                   {}
__thisNAMESPACE                     {}                   {SECURITY_DESCRIPTOR}
__Provider                          {}                   {Name}
__Win32Provider                     {}                   {Name, ClientLoadableCLSID, CLSID, Concurrency...}
__ProviderRegistration              {}                   {provider}
__EventProviderRegistration         {}                   {provider, EventQueryList}
__ObjectProviderRegistration        {}                   {provider, InteractionType, QuerySupportLevels, SupportsBatching...}
__ClassProviderRegistration         {}                   {provider, InteractionType, QuerySupportLevels, SupportsBatching...}
__InstanceProviderRegistration      {}                   {provider, InteractionType, QuerySupportLevels, SupportsBatching...}
__MethodProviderRegistration        {}                   {provider}
__PropertyProviderRegistration      {}                   {provider, SupportsGet, SupportsPut}
__EventConsumerProviderRegistration {}                   {provider, ConsumerClassNames}
__NAMESPACE                         {}                   {Name}
__IndicationRelated                 {}                   {}
__EventFilter                       {}                   {CreatorSID, EventAccess, EventNamespace, Name...}
__EventConsumer                     {}                   {CreatorSID, MachineName, MaximumQueueSize}
__FilterToConsumerBinding           {}                   {Consumer, CreatorSID, DeliverSynchronously, DeliveryQoS...}
__AggregateEvent                    {}                   {NumberOfEvents, Representative}
__TimerNextFiring                   {}                   {NextEvent64BitTime, TimerId}
__Event                             {}                   {SECURITY_DESCRIPTOR, TIME_CREATED}
__ExtrinsicEvent                    {}                   {SECURITY_DESCRIPTOR, TIME_CREATED}
Win32_DeviceChangeEvent             {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, EventType}
Win32_SystemConfigurationChangeE... {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, EventType}
Win32_VolumeChangeEvent             {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, EventType, DriveName}
MSFT_WMI_GenericNonCOMEvent         {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, ProcessId, PropertyNames...}
MSFT_NCProvEvent                    {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, Namespace, ProviderName...}
MSFT_NCProvCancelQuery              {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, Namespace, ProviderName...}
MSFT_NCProvClientConnected          {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, Namespace, ProviderName...}
MSFT_NCProvNewQuery                 {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, Namespace, ProviderName...}
MSFT_NCProvAccessCheck              {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, Namespace, ProviderName...}
Win32_SystemTrace                   {}                   {SECURITY_DESCRIPTOR, TIME_CREATED}
Win32_ProcessTrace                  {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, ParentProcessID, ProcessID...}
Win32_ProcessStartTrace             {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, ParentProcessID, ProcessID...}
Win32_ProcessStopTrace              {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, ParentProcessID, ProcessID...}
Win32_ThreadTrace                   {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, ProcessID, ThreadID}
Win32_ThreadStartTrace              {}                   {SECURITY_DESCRIPTOR, TIME_CREATED, ProcessID, ThreadID...}
```

Figure 23

Step 30

Pick one class (other than **Win32_LogicalDisk**) from the list and access its information.



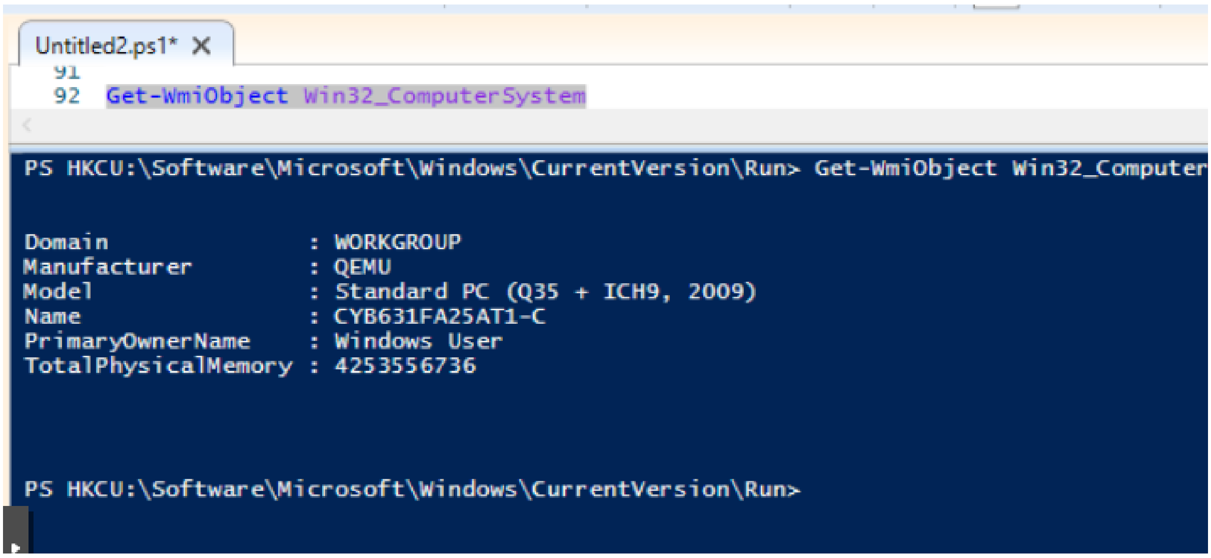
```
Get-CimInstance Win32_OperatingSystem
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-CimInstance Win32_OperatingSystem

SystemDirectory     Organization BuildNumber RegisteredUser SerialNumber            Version
---------------     ------------ ----------- -------------- ------------            -------
C:\Windows\system32              20348       Windows User   00454-40000-00001-AA039 10.0.20348


PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 24

Step 31

Use a **WMI cmdlet** to obtain computer system information:



```
Get-WmiObject Win32_ComputerSystem
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-WmiObject Win32_Computer


Domain              : WORKGROUP
Manufacturer        : QEMU
Model               : Standard PC (Q35 + ICH9, 2009)
Name                : CYB631FA25AT1-C
PrimaryOwnerName    : Windows User
TotalPhysicalMemory : 4253556736



PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

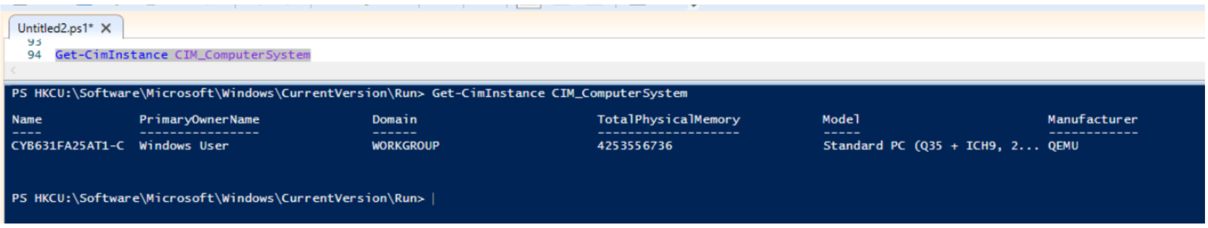
Figure 25

Step 32

Can see the Above Screenshot.

Step 33

Obtain similar information with the CIM cmdlet (cross-platform and more portable):



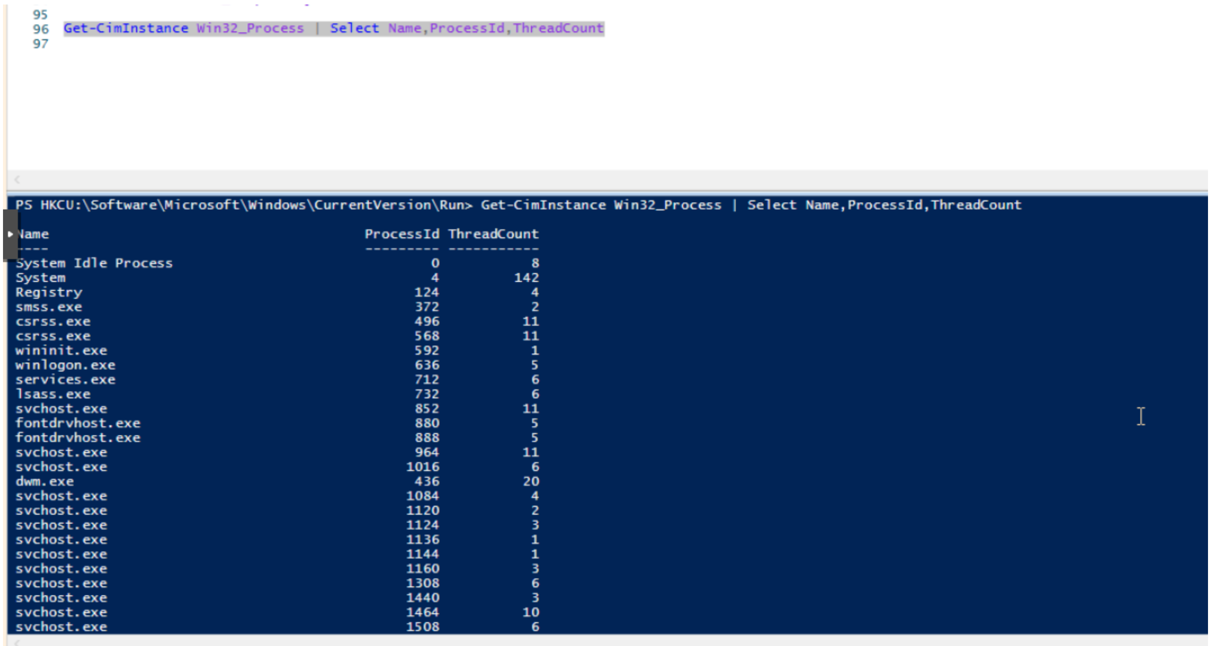
```
Get-CimInstance CIM_ComputerSystem
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-CimInstance CIM_ComputerSystem

Name            PrimaryOwnerName                Domain                          TotalPhysicalMemory             Model                           Manufacturer
----            ----------------                ------                          -------------------             -----                           ------------
CYB631FA25AT1-C Windows User                    WORKGROUP                       4253556736                      Standard PC (Q35 + ICH9, 2... QEMU


PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> |
```

Figure 26

Step 34

Retrieve process information with another CIM cmdlet:



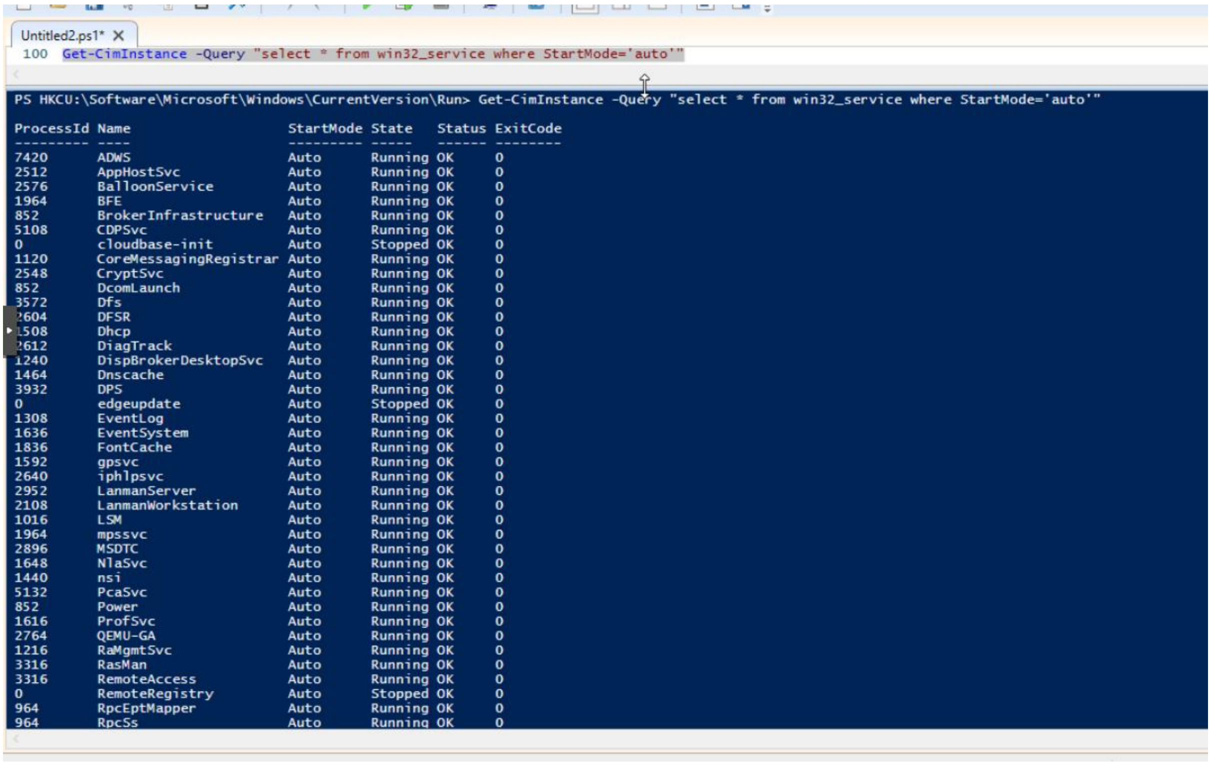
```
Get-CimInstance Win32_Process | Select Name,ProcessId,ThreadCount
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-CimInstance Win32_Process | Select Name,ProcessId,ThreadCount

Name                                      ProcessId ThreadCount
----                                      --------- -----------
System Idle Process                               0           8
System                                            4         142
Registry                                        124           4
smss.exe                                        372           2
csrss.exe                                       496          11
csrss.exe                                       568          11
wininit.exe                                     592           1
winlogon.exe                                    636           5
services.exe                                    712           6
lsass.exe                                       732           6
svchost.exe                                     852          11
fontdrvhost.exe                                 880           5
fontdrvhost.exe                                 888           5
svchost.exe                                     964          11
svchost.exe                                    1016           6
dwm.exe                                         436          20
svchost.exe                                    1084           4
svchost.exe                                    1120           2
svchost.exe                                    1124           3
svchost.exe                                    1136           1
svchost.exe                                    1144           1
svchost.exe                                    1160           3
svchost.exe                                    1308           6
svchost.exe                                    1440           3
svchost.exe                                    1464          10
svchost.exe                                    1508           6
```

Figure 27

Step 35

Use a **WQL query** (similar to SQL) to access service information:



Untitled2.ps1*

```
Get-CimInstance -Query "select * from win32_service where StartMode='auto'"
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-CimInstance -Query "select * from win32_service where StartMode='auto'"

ProcessId Name                   StartMode State   Status ExitCode
--------- ----                   --------- -----   ------ --------
7420      ADWS                   Auto      Running OK     0
2512      AppHostSvc             Auto      Running OK     0
2576      BalloonService         Auto      Running OK     0
1964      BFE                    Auto      Running OK     0
852       BrokerInfrastructure   Auto      Running OK     0
5108      CDPSvc                 Auto      Running OK     0
0         cloudbase-init         Auto      Stopped OK     0
1120      CoreMessagingRegistrar Auto      Running OK     0
2548      CryptSvc               Auto      Running OK     0
852       DcomLaunch             Auto      Running OK     0
3572      Dfs                    Auto      Running OK     0
2604      DFSR                   Auto      Running OK     0
1508      Dhcp                   Auto      Running OK     0
2612      DiagTrack              Auto      Running OK     0
1240      DispBrokerDesktopSvc   Auto      Running OK     0
1464      Dnscache               Auto      Running OK     0
3932      DPS                    Auto      Running OK     0
0         edgeupdate             Auto      Stopped OK     0
1308      EventLog               Auto      Running OK     0
1636      EventSystem            Auto      Running OK     0
1836      FontCache              Auto      Running OK     0
1592      gpsvc                  Auto      Running OK     0
2640      iphlpsvc               Auto      Running OK     0
2952      LanmanServer           Auto      Running OK     0
2108      LanmanWorkstation      Auto      Running OK     0
1016      LSM                    Auto      Running OK     0
1964      mpssvc                 Auto      Running OK     0
2896      MSDTC                  Auto      Running OK     0
1648      NlaSvc                 Auto      Running OK     0
1440      nsi                    Auto      Running OK     0
5132      PcaSvc                 Auto      Running OK     0
852       Power                  Auto      Running OK     0
1616      ProfSvc                Auto      Running OK     0
2764      QEMU-GA                Auto      Running OK     0
1216      RaMgmtSvc              Auto      Running OK     0
3316      RasMan                 Auto      Running OK     0
3316      RemoteAccess           Auto      Running OK     0
0         RemoteRegistry         Auto      Stopped OK     0
964       RpcEptMapper           Auto      Running OK     0
964       RpcSs                  Auto      Running OK     0
```

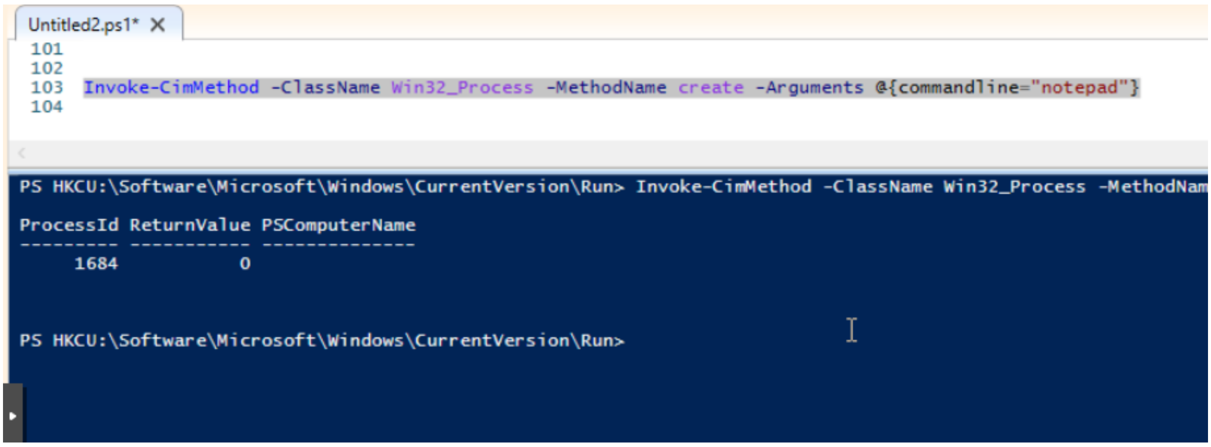
Figure 28

Step 36

The command from Step 35 retrieved a list of all Windows services configured with **StartMode = Auto**. This means these services automatically start whenever the system boots. The results give insight into which background processes are essential for system operation and can also help identify unnecessary or potentially vulnerable services that launch at startup.

Step 37

The command



```
Invoke-CimMethod -ClassName Win32_Process -MethodName create -Arguments @{commandline="notepad"}
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Invoke-CimMethod -ClassName Win32_Process -MethodNam

ProcessId ReturnValue PSComputerName
--------- ----------- --------------
     1684           0


PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 29

Step 38

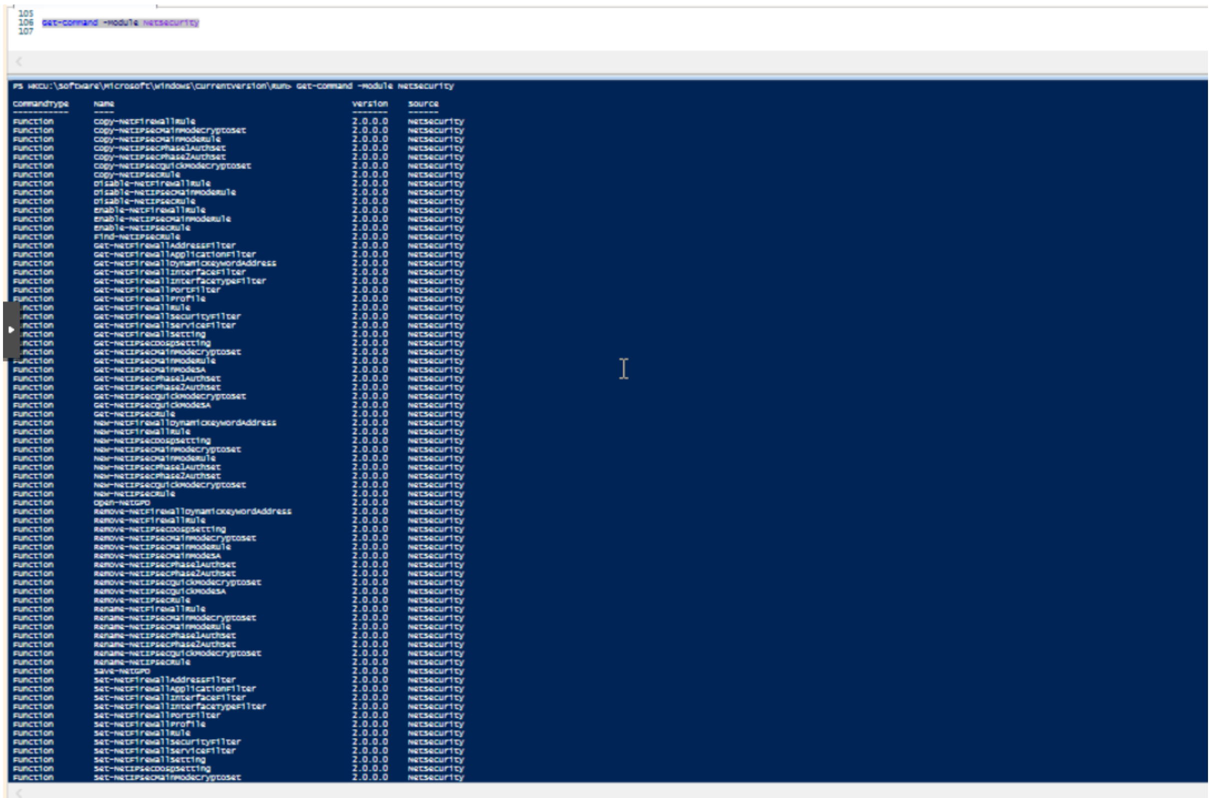
What happened after running the script above?

The **Invoke-CimMethod** command successfully launched **Notepad** as a new process. The PowerShell output confirms this with a **ProcessId (1684)** and a **ReturnValue of 0**, indicating the process creation was executed without errors.

EXERCISE V: CONFIGURE WINDOWS FIREWALL

Step 39

PowerShell has a **NetSecurity module** for configuring Firewall and IPSec.



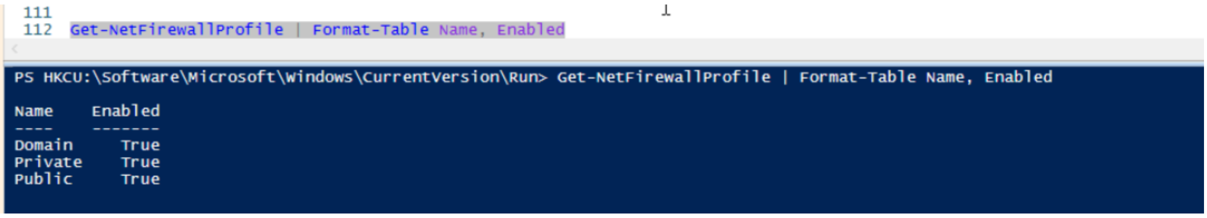
```
Get-Command -Module NetSecurity
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-Command -Module NetSecurity

CommandType     Name                                           Version    Source
-----------     ----                                           -------    ------
Function        Copy-NetFirewallRule                           2.0.0.0    NetSecurity
Function        Copy-NetIPsecMainModeCryptoSet                 2.0.0.0    NetSecurity
Function        Copy-NetIPsecMainModeRule                      2.0.0.0    NetSecurity
Function        Copy-NetIPsecPhase1AuthSet                     2.0.0.0    NetSecurity
Function        Copy-NetIPsecPhase2AuthSet                     2.0.0.0    NetSecurity
Function        Copy-NetIPsecQuickModeCryptoSet                2.0.0.0    NetSecurity
Function        Copy-NetIPsecRule                              2.0.0.0    NetSecurity
Function        Disable-NetFirewallRule                        2.0.0.0    NetSecurity
Function        Disable-NetIPsecMainModeRule                   2.0.0.0    NetSecurity
Function        Disable-NetIPsecRule                           2.0.0.0    NetSecurity
Function        Enable-NetFirewallRule                         2.0.0.0    NetSecurity
Function        Enable-NetIPsecMainModeRule                    2.0.0.0    NetSecurity
Function        Enable-NetIPsecRule                            2.0.0.0    NetSecurity
Function        Find-NetIPsecRule                              2.0.0.0    NetSecurity
Function        Get-NetFirewallAddressFilter                   2.0.0.0    NetSecurity
Function        Get-NetFirewallApplicationFilter               2.0.0.0    NetSecurity
Function        Get-NetFirewallDynamicKeywordAddress           2.0.0.0    NetSecurity
Function        Get-NetFirewallInterfaceFilter                 2.0.0.0    NetSecurity
Function        Get-NetFirewallInterfaceTypeFilter             2.0.0.0    NetSecurity
Function        Get-NetFirewallPortFilter                      2.0.0.0    NetSecurity
Function        Get-NetFirewallProfile                         2.0.0.0    NetSecurity
Function        Get-NetFirewallRule                            2.0.0.0    NetSecurity
Function        Get-NetFirewallSecurityFilter                  2.0.0.0    NetSecurity
Function        Get-NetFirewallServiceFilter                   2.0.0.0    NetSecurity
Function        Get-NetFirewallSetting                         2.0.0.0    NetSecurity
Function        Get-NetIPsecDospSetting                        2.0.0.0    NetSecurity
Function        Get-NetIPsecMainModeCryptoSet                  2.0.0.0    NetSecurity
Function        Get-NetIPsecMainModeRule                       2.0.0.0    NetSecurity
Function        Get-NetIPsecMainModeSA                         2.0.0.0    NetSecurity
Function        Get-NetIPsecPhase1AuthSet                      2.0.0.0    NetSecurity
Function        Get-NetIPsecPhase2AuthSet                      2.0.0.0    NetSecurity
Function        Get-NetIPsecQuickModeCryptoSet                 2.0.0.0    NetSecurity
Function        Get-NetIPsecQuickModeSA                        2.0.0.0    NetSecurity
Function        Get-NetIPsecRule                               2.0.0.0    NetSecurity
Function        New-NetFirewallDynamicKeywordAddress           2.0.0.0    NetSecurity
Function        New-NetFirewallRule                            2.0.0.0    NetSecurity
Function        New-NetIPsecDospSetting                        2.0.0.0    NetSecurity
Function        New-NetIPsecMainModeCryptoSet                  2.0.0.0    NetSecurity
Function        New-NetIPsecMainModeRule                       2.0.0.0    NetSecurity
Function        New-NetIPsecPhase1AuthSet                      2.0.0.0    NetSecurity
Function        New-NetIPsecPhase2AuthSet                      2.0.0.0    NetSecurity
Function        New-NetIPsecQuickModeCryptoSet                 2.0.0.0    NetSecurity
Function        New-NetIPsecRule                               2.0.0.0    NetSecurity
Function        Open-NetGPO                                    2.0.0.0    NetSecurity
Function        Remove-NetFirewallDynamicKeywordAddress        2.0.0.0    NetSecurity
Function        Remove-NetFirewallRule                         2.0.0.0    NetSecurity
Function        Remove-NetIPsecDospSetting                     2.0.0.0    NetSecurity
Function        Remove-NetIPsecMainModeCryptoSet               2.0.0.0    NetSecurity
Function        Remove-NetIPsecMainModeRule                    2.0.0.0    NetSecurity
Function        Remove-NetIPsecMainModeSA                      2.0.0.0    NetSecurity
Function        Remove-NetIPsecPhase1AuthSet                   2.0.0.0    NetSecurity
Function        Remove-NetIPsecPhase2AuthSet                   2.0.0.0    NetSecurity
Function        Remove-NetIPsecQuickModeCryptoSet              2.0.0.0    NetSecurity
Function        Remove-NetIPsecQuickModeSA                     2.0.0.0    NetSecurity
Function        Remove-NetIPsecRule                            2.0.0.0    NetSecurity
Function        Rename-NetFirewallRule                         2.0.0.0    NetSecurity
Function        Rename-NetIPsecMainModeCryptoSet               2.0.0.0    NetSecurity
Function        Rename-NetIPsecMainModeRule                    2.0.0.0    NetSecurity
Function        Rename-NetIPsecPhase1AuthSet                   2.0.0.0    NetSecurity
Function        Rename-NetIPsecPhase2AuthSet                   2.0.0.0    NetSecurity
Function        Rename-NetIPsecQuickModeCryptoSet              2.0.0.0    NetSecurity
Function        Rename-NetIPsecRule                            2.0.0.0    NetSecurity
Function        Save-NetGPO                                    2.0.0.0    NetSecurity
Function        Set-NetFirewallAddressFilter                   2.0.0.0    NetSecurity
Function        Set-NetFirewallApplicationFilter               2.0.0.0    NetSecurity
Function        Set-NetFirewallInterfaceFilter                 2.0.0.0    NetSecurity
Function        Set-NetFirewallInterfaceTypeFilter             2.0.0.0    NetSecurity
Function        Set-NetFirewallPortFilter                      2.0.0.0    NetSecurity
Function        Set-NetFirewallProfile                         2.0.0.0    NetSecurity
Function        Set-NetFirewallRule                            2.0.0.0    NetSecurity
Function        Set-NetFirewallSecurityFilter                  2.0.0.0    NetSecurity
Function        Set-NetFirewallServiceFilter                   2.0.0.0    NetSecurity
Function        Set-NetFirewallSetting                         2.0.0.0    NetSecurity
Function        Set-NetIPsecDospSetting                        2.0.0.0    NetSecurity
Function        Set-NetIPsecMainModeCryptoSet                  2.0.0.0    NetSecurity
```

Figure 30

Step 40

PowerShell has



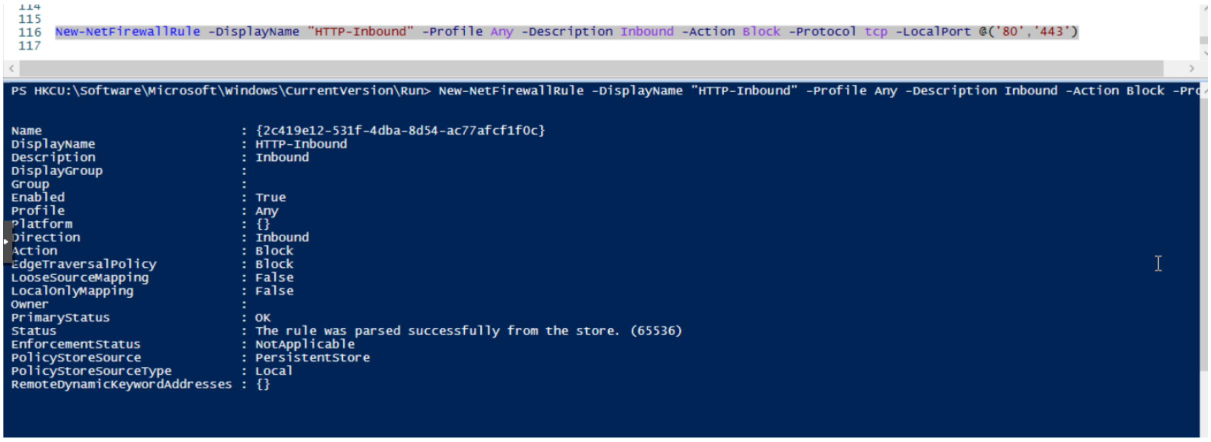
```
Get-NetFirewallProfile | Format-Table Name, Enabled
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Get-NetFirewallProfile | Format-Table Name, Enabled

Name    Enabled
----    -------
Domain     True
Private    True
Public     True
```

Figure 31

Step 41

Block access to web servers inside this host by creating a new firewall rule.



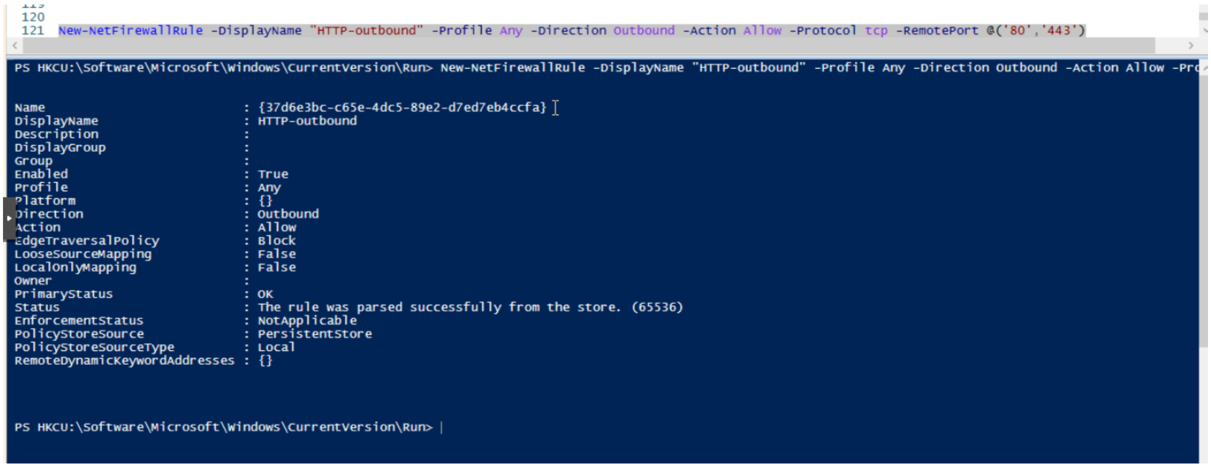
```
New-NetFirewallRule -DisplayName "HTTP-Inbound" -Profile Any -Description Inbound -Action Block -Protocol tcp -LocalPort @('80','443')
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-NetFirewallRule -DisplayName "HTTP-Inbound" -Profile Any -Description Inbound -Action Block -Pro

Name                          : {2c419e12-531f-4dba-8d54-ac77afcf1f0c}
DisplayName                   : HTTP-Inbound
Description                   : Inbound
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Inbound
Action                        : Block
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}
```

Figure 32

Step 42

Test access using a browser. At this stage, you should still be able to access external servers. 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')
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-NetFirewallRule -DisplayName "HTTP-outbound" -Profile Any -Direction Outbound -Action Allow -Pro

Name                          : {37d6e3bc-c65e-4dc5-89e2-d7ed7eb4ccfa}
DisplayName                   : HTTP-outbound
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Outbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}


PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> |
```

Figure 33

Step 43

Try accessing external web servers again using a browser.

Are you still able to access them?

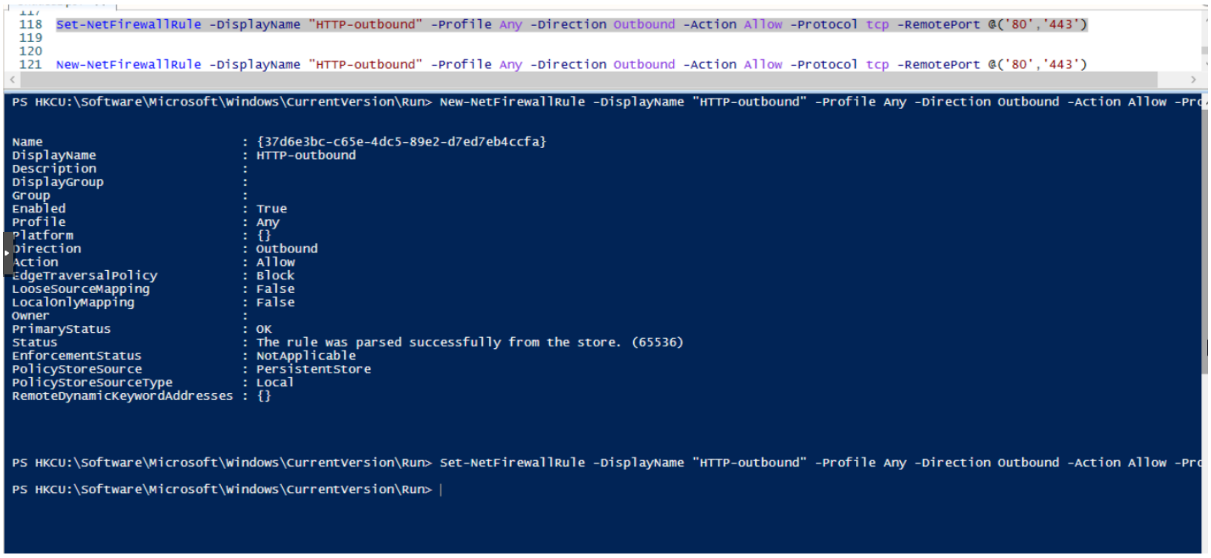
No.

Why or why not?

Because the outbound firewall rule blocks communication on ports 80 and 443, which are required for HTTP and HTTPS traffic.

Step 44

Re-enable internet web access by modifying the rule created earlier.



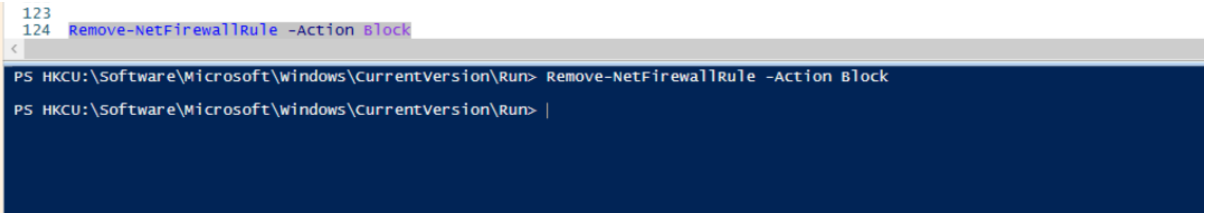
```
Set-NetFirewallRule -DisplayName "HTTP-outbound" -Profile Any -Direction Outbound -Action Allow -Protocol tcp -RemotePort @('80','443')


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

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-NetFirewallRule -DisplayName "HTTP-outbound" -Profile Any -Direction Outbound -Action Allow -Pr


Name                          : {37d6e3bc-c65e-4dc5-89e2-d7ed7eb4ccfa}
DisplayName                   : HTTP-outbound
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Any
Platform                      : {}
Direction                     : Outbound
Action                        : Allow
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}



PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Set-NetFirewallRule -DisplayName "HTTP-outbound" -Profile Any -Direction Outbound -Action Allow -Pr

PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> |
```

Figure 34

Step 45

Remove firewall rules if needed.



```

Remove-NetFirewallRule -Action Block
```

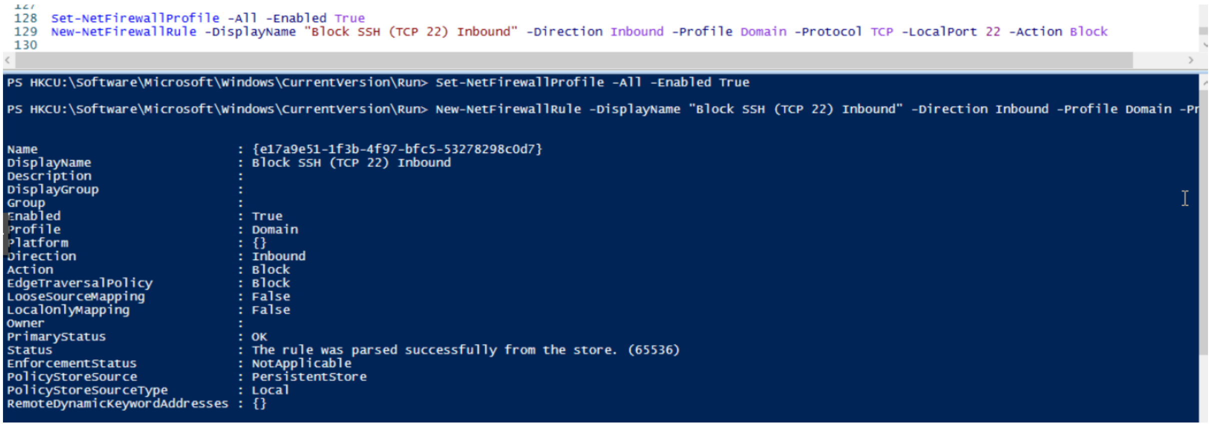
```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Remove-NetFirewallRule -Action Block

PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> |
```

Figure 35

EXERCISE VI: CONFIGURING WINDOWS FIREWALL WITH POWERSHELL

Step 46: Enabling Firewall and Blocking SSH/DNS Ports

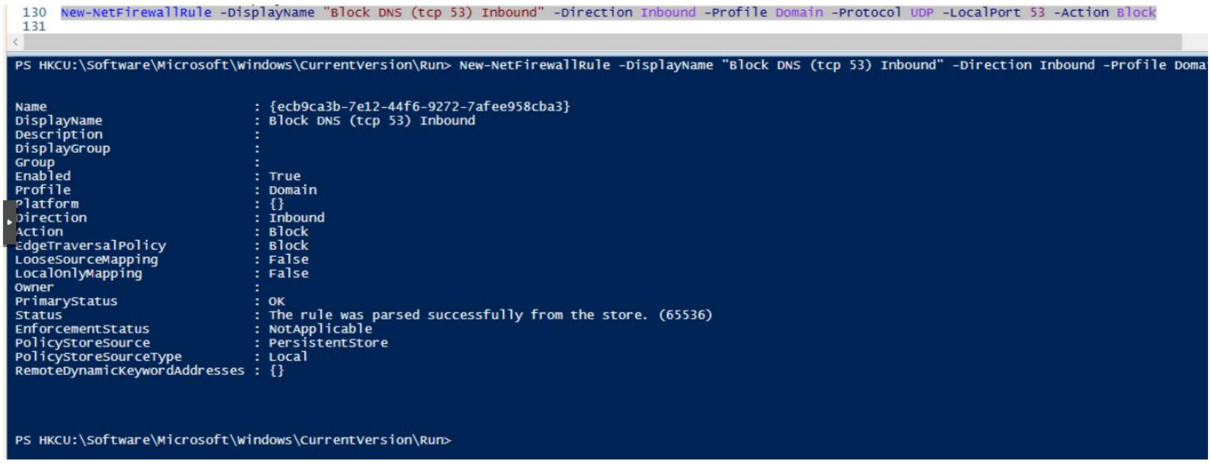
To meet the requirements, the following PowerShell script is used. It enables the Windows Defender Firewall on **all profiles** (Domain, Private, Public) and then creates two inbound firewall rules to block SSH and DNS traffic in the Active Directory **Domain** network. The first rule blocks **SSH** access on TCP port 22, and the second rule blocks **DNS** queries on port 53 (both TCP and UDP) – effectively stopping outside attempts to reach internal SSH or DNS servers.



```
Set-NetFirewallProfile -All -Enabled True
New-NetFirewallRule -DisplayName "Block SSH (TCP 22) Inbound" -Direction Inbound -Profile Domain -Protocol TCP -LocalPort 22 -Action Block
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> Set-NetFirewallProfile -All -Enabled True

PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-NetFirewallRule -DisplayName "Block SSH (TCP 22) Inbound" -Direction Inbound -Profile Domain -Pr


Name                          : {e17a9e51-1f3b-4f97-bfc5-53278298c0d7}
DisplayName                   : Block SSH (TCP 22) Inbound
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Domain
Platform                      : {}
Direction                     : Inbound
Action                        : Block
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}
```

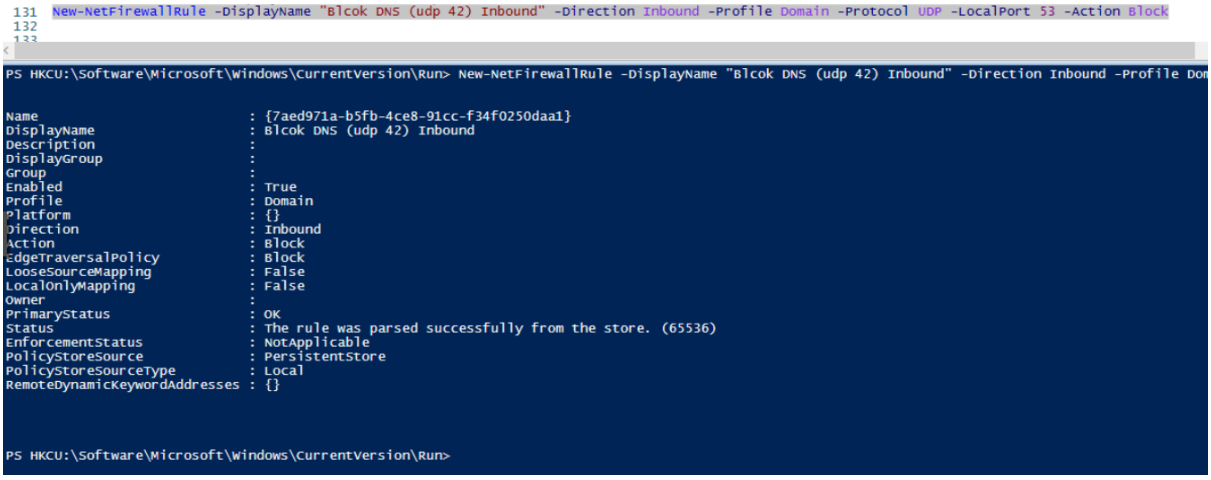
Figure 36



```
New-NetFirewallRule -DisplayName "Block DNS (tcp 53) Inbound" -Direction Inbound -Profile Domain -Protocol UDP -LocalPort 53 -Action Block
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-NetFirewallRule -DisplayName "Block DNS (tcp 53) Inbound" -Direction Inbound -Profile Doma


Name                          : {ecb9ca3b-7e12-44f6-9272-7afee958cba3}
DisplayName                   : Block DNS (tcp 53) Inbound
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Domain
Platform                      : {}
Direction                     : Inbound
Action                        : Block
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}



PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 37



```
New-NetFirewallRule -DisplayName "Blcok DNS (udp 42) Inbound" -Direction Inbound -Profile Domain -Protocol UDP -LocalPort 53 -Action Block
```

```
PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run> New-NetFirewallRule -DisplayName "Blcok DNS (udp 42) Inbound" -Direction Inbound -Profile Dom


Name                          : {7aed971a-b5fb-4ce8-91cc-f34f0250daa1}
DisplayName                   : Blcok DNS (udp 42) Inbound
Description                   :
DisplayGroup                  :
Group                         :
Enabled                       : True
Profile                       : Domain
Platform                      : {}
Direction                     : Inbound
Action                        : Block
EdgeTraversalPolicy           : Block
LooseSourceMapping            : False
LocalOnlyMapping              : False
Owner                         :
PrimaryStatus                 : OK
Status                        : The rule was parsed successfully from the store. (65536)
EnforcementStatus             : NotApplicable
PolicyStoreSource             : PersistentStore
PolicyStoreSourceType         : Local
RemoteDynamicKeywordAddresses : {}



PS HKCU:\Software\Microsoft\Windows\CurrentVersion\Run>
```

Figure 38

Step 47: Verifying the New Firewall Rules

After running the script, the new rules appear in the firewall configuration. In **Figure 1**, the PowerShell console output (for a similar rule creation) shows a firewall rule added with the specified **DisplayName** and settings. This indicates the commands executed successfully. On an actual system, you would see entries for “Block SSH (TCP 22) Inbound” and “Block DNS (TCP/UDP 53) Inbound” listed as enabled **inbound** rules in the Domain profile, confirming that any traffic on those ports will be blocked as intended.

Step 48: Explanation of the PowerShell Script

This PowerShell script performs two main functions: enabling the firewall and adding specific block rules. First, it uses the Set-NetFirewallProfile -All -Enabled True cmdlet to turn on the Windows Firewall for all three network profiles (Domain, Private, and Public)woshub.com. Enabling all profiles ensures the host’s firewall is active in any network context (especially important for the Domain profile, which represents the Active Directory network).

Next, the script defines new inbound firewall rules using New-NetFirewallRule. Key parameters are provided to precisely target the undesired trafficitprotoday.comitprotoday.com.

For example, the -DisplayName gives each rule a clear name (e.g., “Block SSH (TCP 22) Inbound”), and -Direction Inbound specifies that the rule applies to incoming traffic. The -Profile Domain parameter scope limits the rule to the domain network profile (i.e., when the machine is connected to the AD domain network). We use -Protocol TCP (for SSH and for one DNS rule) or -Protocol UDP (for the other DNS rule) along with -LocalPort set to the respective port number to identify the network packets these rules should match. The -Action Block option ensures that any traffic meeting these criteria is **dropped** (not allowed through)itprotoday.com. In summary, the script explicitly blocks inbound SSH traffic on port 22 and DNS queries on port 53 in the domain environment, thereby preventing external hosts from reaching those services on the server.

Step 49: Advantages of Using PowerShell for Firewall Configuration

Using PowerShell scripts to configure the Windows Firewall offers significant advantages over manual GUI configuration. One major benefit is **efficiency at scale**: PowerShell allows administrators to automate changes across many systems simultaneously, something that is impractical with the point-and-click GUI. This means firewall rules can be deployed or updated on multiple servers or workstations through a single script, ensuring consistency. In enterprise environments or Active Directory domains, this ability to manage computers in bulk is crucialitprotoday.com. Moreover, servers running in **Core** mode (without a GUI) can only be managed via command-line, making PowerShell not just convenient but necessary in those casesitprotoday.com.

Another advantage is that PowerShell configurations are **repeatable and auditable**. A script serves as documentation of the firewall settings implemented, which aids in compliance and troubleshooting. Changes made via script can be tracked in version control and reviewed by others, unlike manual changes that might go undocumented. PowerShell’s scriptable approach also reduces human error by automating the correct sequence of commands every time. This leads to more reliable deployments of security rules. In short, automation with PowerShell provides a **robust and scalable** method for managing firewall rules – administrators can easily re-run scripts to enforce standard rules on new machines or update settings, saving time and ensuring uniform security policies across the networkninjaone.comninjaone.com. By using scripting, firewall configurations become part of an organization's infrastructure-as-code practice, streamlining how security is maintained on Windows hosts.

LAB AND CLASS REFLECTION

What I Liked About This Lab

This lab provided a valuable hands-on experience with **PowerShell**, Active Directory, Windows Registry, WMI/CIM, and Windows Firewall. I particularly liked how the exercises built progressively—from environment setup to automation—showing how administrative tasks can be both performed manually and scripted. The lab reinforced practical cybersecurity skills that are directly relevant to real-world system hardening and automation.

Challenges Encountered

One challenge I faced was ensuring that the PowerShell commands executed properly in the **virtual environment**, especially when configuring Active Directory Lightweight Directory Services and testing firewall rules. Some commands required elevated privileges or specific context (e.g., running in the correct profile or on the correct VM). Debugging these issues required careful reading of error outputs and verification of execution policies.

Suggestions for Improving the Class

To further enhance the learning experience, I suggest incorporating **more guided troubleshooting examples**. For instance, common errors that occur when configuring AD LDS, registry permissions, or CIM queries could be provided with solutions. Additionally, short demo videos for complex steps (like configuring firewall rules or using WQL queries) would help clarify expectations and reduce confusion.

Final Note

Shut down the Cyber Range Server.

REFERENCES

Microsoft. (n.d.). *Active Directory Lightweight Directory Services overview*. Microsoft Learn. https://learn.microsoft.com/en-us/windows-server/identity/ad-lds/active-directory-lightweight-directory-services-overview

Microsoft. (n.d.). *Windows PowerShell commands for managing the Windows Firewall*. Microsoft Learn. https://learn.microsoft.com/en-us/powershell/module/netsecurity

Microsoft. (n.d.). *Windows Management Instrumentation (WMI)*. Microsoft Learn. https://learn.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page

Microsoft. (n.d.). *Windows registry information for advanced users*. Microsoft Support. https://support.microsoft.com/help/256986/windows-registry-information-for-advanced-users

NinjaOne. (2023, July 14). *Configure firewall exceptions with PowerShell (script hub)*. NinjaOne. https://www.ninjaone.com/script-hub/configure-firewall-exceptions-with-powershell

Woshub. (2022, April 18). *Manage Windows Defender Firewall with PowerShell*. Woshub. https://woshub.com/manage-windows-firewall-powershell

Wong, J. (2021, March 23). *Managing Windows firewall rules with PowerShell: Beyond the GUI*. ITPro Today. https://www.itprotoday.com/powershell/managing-windows-firewall-rules-with-powershell-part-1-beyond-the-gui