Just Enough Admin (JEA)

Note: For the “PowerShell Remoting” and “Just Enough Admin (JEA)” labs you will need a second lab virtual machine. You can create the second virtual machine by selection option 2 during the setup described [here](https://onedrive.live.com/view.aspx?resid=5EFC811CDEC9D7F0!9867&ithint=file%2cdocx&authkey=!AFmEtJQN-wdNXmA).

Just Enough Administration (JEA) is a PowerShell feature that allows administrators to tightly control what functions, cmdlets and external programs are available to users of PowerShell sessions. The permissions can be set per user or group, and they can even allow users who are not administrators to perform specific administrative actions.

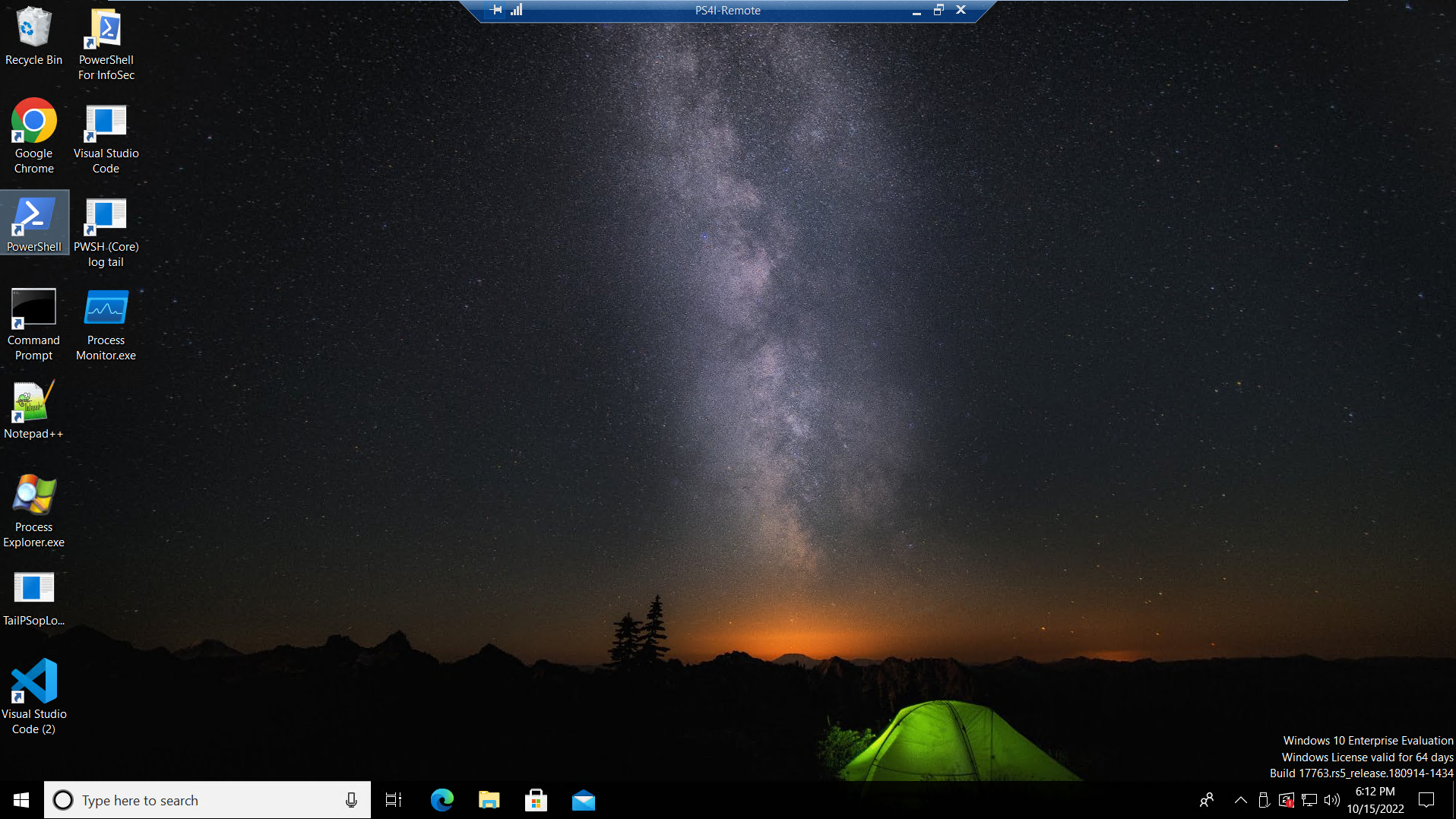
In this lab, we are going to connect to the Remote VM and run a JEA setup script. Then we will return to our main VM and connect to our remote JEA endpoint using PowerShell Remoting.

Use the **Remote.rdp** shortcut on the desktop to connect to your remote VM from your main VM.

Graphical user interface

Description automatically generated with medium confidence

Login with that same credentials you use to log into your main VM (IEUser:Passw0rd!). If you have connected correctly, you’ll see a remote lab VM with the desktop background that looks like the image below.



From the Remote VM, open the **C:\Users\IEUser\PowerShellForInfoSec\Samples\JEA.ps1** file in notepad and familiarize yourself with the code. In the first section, it creates the Role Capabilities File. This defines a role called **SpoolerRestart** and allows users with this role to run any **Get** cmdlets, and to stop or start the spooler service.

A picture containing text

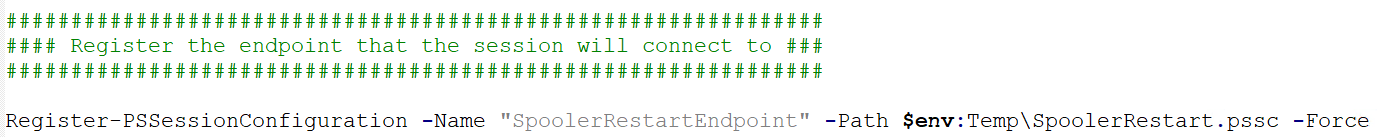
Description automatically generated

The next section in the code creates the session configuration file. This defines which roles are assigned to various users or groups. In this case, anyone in the **PrinterAdmin** group will be given a role of **SpoolerRestart**.

Text

Description automatically generated

The very last step is to create the endpoint. We give the endpoint a name of **SpoolerRestartEndpoint** and indicate that we want it configured according to the session configuration file we created.



And that sums it up. We create a file defining a role and what users with that role should be able to do (the role capabilities file). Then we create a file to assign roles to certain users/groups (the session configuration file). Lastly, we create the endpoint by specifying its name and giving it a copy of our session configuration file.

On the remote VM, start PowerShell as an administrator and run the **JEA.ps1** script that we just reviewed.

Graphical user interface, text, application, website

Description automatically generated

Make note of the users that are in the **PrinterAdmins** group on the Remote VM. You should see that bob is the only user and if you check his group membership you will see that he is not an administrator of the Remote VM.

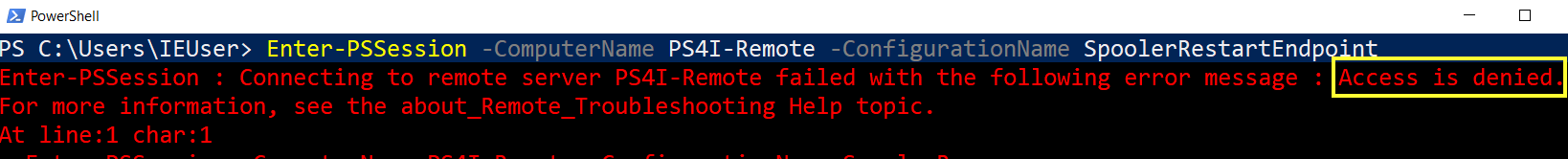
Text, application

Description automatically generated with medium confidence

Now you can close the remote desktop connection to the Remote VM.

From a PowerShell window on the Main lab VM, try to connect to the remote endpoint as follows.

Enter-PSSession -ComputerName PS4I-Remote -ConfigurationName SpoolerRestartEndpoint



In this case we are trying to connect as the current user (**IEUser**) who is not in the **PrinterAdmins** group. Due to the way we configured the remote endpoint, the only users that can connect to the **SpoolerRestartEndpoint** are the ones in the **PrinterAdmins** group (**bob** in this case). To connect as **bob**, we can pass alternate creds. Use the Get-Credential cmdlet to gather the alternate credentials (bob’s credentials)

Graphical user interface, text, application

Description automatically generated

Enter a username of **bob** and a password of **Passw0rd!**

Now we can try the **Enter-PSSession** command we tried before and add in bob’s credentials.

Graphical user interface, text

Description automatically generated

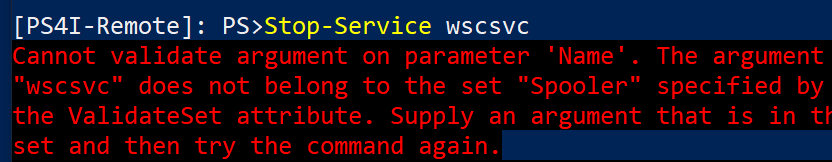
The connection is successful this time and the prompt changes to indicate that we are connected to the remote machine. Run **Get-Command** to see what commands are available to us on this endpoint. Is it what you expect to see?

A picture containing graphical user interface

Description automatically generated

The **Get**, **Stop**, and **Restart** service entries are what we defined in our JEA configuration files. The other commands listed are basic functionality added to all JEA endpoints by default.

Try to stop or restart a service. The only Service you should be able to restart is the Spooler service based on the way we set up the role capabilities file. Here we try to stop the **wscsvc** service and get rejected (this is what we were expecting based on our configuration of the endpoint).



Try changing the JEA.ps1 file to add or remove allowed commands. Also try adding a **TranscriptDirectory** so you can see what gets logged there when the endpoint is used.

A picture containing graphical user interface

Description automatically generated

Remember to rerun the JEA.ps1 file from an administrative PowerShell prompt on the Remote VM, then return to the Main VM and reconnect to the remote endpoint as the user **bob**.

Execute some commands and experiment with the changes you made to the role capabilities file. When you are done, make a remote desktop connection to the Remote VM again and have a look at the transcripts in the **PStranscripts** directory.

Graphical user interface

Description automatically generated

The endpoint allows any user in the **PrinterAdmins** group to restart the spooler service. Restarting this service requires administrative access, yet we were able to allow **bob** to restart this specific service without having to make him an admin on the remote VM. This is the power of Just Enough Administration!