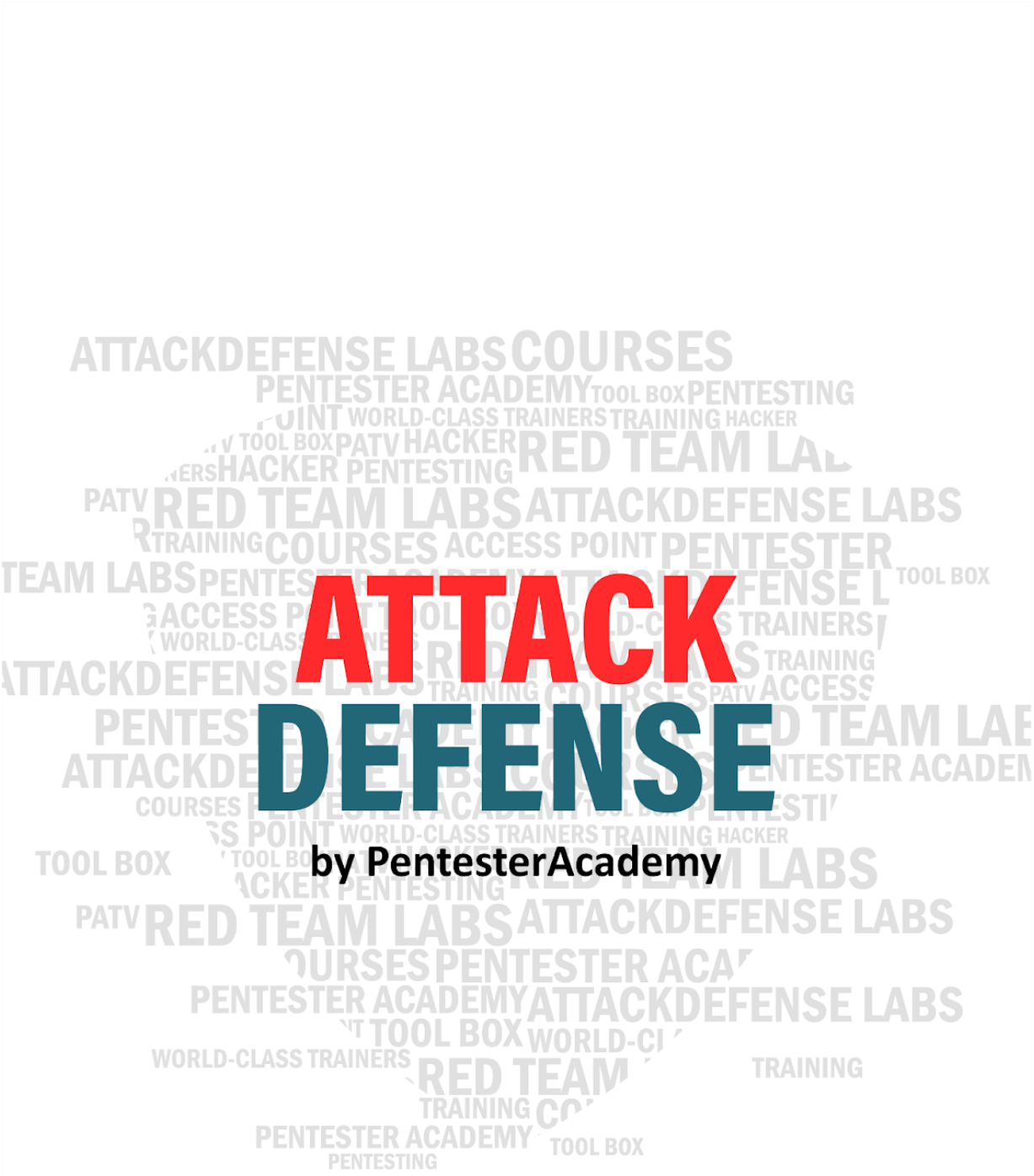

ATTACK
DEFENSE
by PentesterAcademy

Name	WinRM: Configure via Windows PowerShell
URL	https://attackdefense.com/challengedetails?cid=2033
Type	Windows Exploitation: Services

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Note: By default if you are using Windows Server then, WinRM service is already up and running. You need to configure the service in order to access it remotely. In this manual we are demonstrating how to enable WinRM service and making necessary changes for learning purposes.

Configuration of WinRM

Step 1: Checking the status of the winrm service on the target machine. Run powershell.exe to check for WinRM service status, if it's running or not.

Command: Get-Service winrm

```
Administrator: Windows PowerShell
PS C:\Users\Administrator> Get-Service winrm

Status   Name               DisplayName
------   ----               -----------
Stopped  winrm              Windows Remote Management (WS-Manag...


PS C:\Users\Administrator>
```

The WinRM service is not running. We will use powershell commands to enable the WinRM service. Start the WinRM service.

Command: Start-Service WinRM

```
PS C:\Users\Administrator> Start-Service WinRM
PS C:\Users\Administrator> Get-Service WinRm

Status   Name               DisplayName
------   ----               -----------
Running  WinRm              Windows Remote Management (WS-Manag...


PS C:\Users\Administrator>
```

The WinRM service is up and running.

Step 2: We could configure the winrm service by invoking the "**Set-WSManQuickConfig**" The cmdlet which performs the following tasks to make the WinRM service in ready state.

- Starts the WinRM service.
- Sets the startup type on the WinRM service to Automatic.
- Creates a listener to accept requests on any IP address.
- Enables a firewall exception for WS-Management communications.
- Creates the simple and long name session endpoint configurations if needed.
- Enables all session configurations.
- Changes the security descriptor of all session configurations to allow remote access.

```
PS C:\Users\Administrator> Set-WSManQuickConfig

WinRM Quick Configuration
Running the Set-WSManQuickConfig command has significant security implications, as it enables remote management through the WinRM service on this computer.
This command:
 1. Checks whether the WinRM service is running. If the WinRM service is not running, the service is started.
 2. Sets the WinRM service startup type to automatic.
 3. Creates a listener to accept requests on any IP address. By default, the transport is HTTP.
 4. Enables a firewall exception for WS-Management traffic.
 5. Enables Kerberos and Negotiate service authentication.
Do you want to enable remote management through the WinRM service on this computer?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
WinRM has been updated to receive requests.
WinRM service type changed successfully.

WinRM has been updated for remote management.
WinRM firewall exception enabled.
Configured LocalAccountTokenFilterPolicy to grant administrative rights remotely to local users.

PS C:\Users\Administrator>
```

We have successfully configured the WinRM service on the remote server.

Note: Switch to Attacker machine and execute below commands

Step 3: We also need to allow clients (administration machines) to connect to the specific or all remote servers by modifying the "**TrustedHosts**". Switch to the **client machine** and configure TrustedHosts to allow all remote servers.

Note: You cannot modify the TrustedHosts file if the WinRM service is not running.

Starting the winrm service

Command: Start-Service winrm

```
PS C:\Users\Administrator> Start-Service winrm
PS C:\Users\Administrator>
PS C:\Users\Administrator>
```

The winrm service is now running.

Step 4: Modifying the TrustedHosts file and allowing all remote hosts to connect.

Command: Set-item wsman:localhost\client\trustedhosts -value *

```
PS C:\Users\Administrator> Set-item wsman:localhost\client\trustedhosts -value *

WinRM Security Configuration.
This command modifies the TrustedHosts list for the WinRM client. The computers in the TrustedHosts list might not be au
thenticated. The client might send credential information to these computers. Are you sure
 that you want to modify this list?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"): Y
PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator>
```

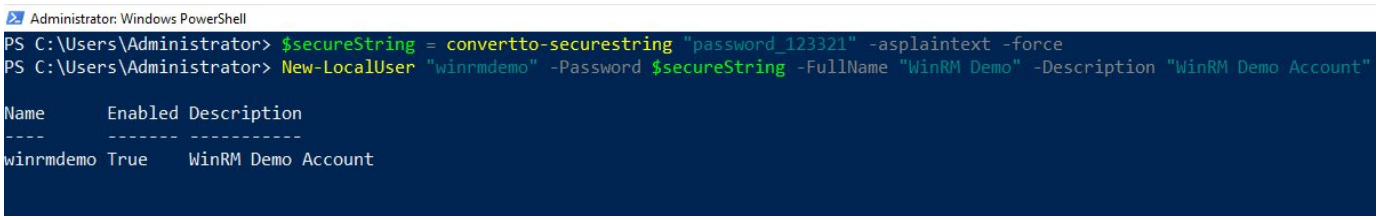
We have modified trustedhosts and allowed any remote servers.

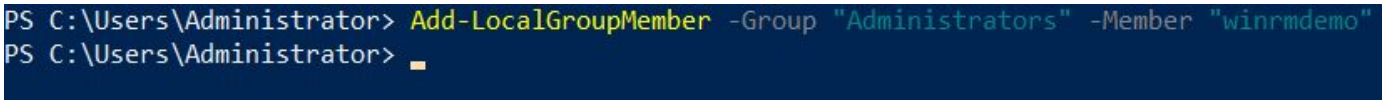
Note: It's always a good practice to mention the IP address or a computer name of the remote server. Also, you don't need to keep the WinRM service running on the client machine. We can turn it off after modifying the TrustedHosts.

Creating a Demo User

Step 1: We will create a demo user on the target machine and the same user credentials we will use to execute commands from the client .

Commands:
$secureString = convertto-securestring "password_123321" -asplaintext -force
New-LocalUser "winrmdemo" -Password $secureString -FullName "WinRM Demo" -Description "WinRM Demo Account"
Add-LocalGroupMember -Name 'Administrators' -Member 'winrmdemo'





We have successfully, created an user i.e “winrmdemo” and added the user in the administrators group.

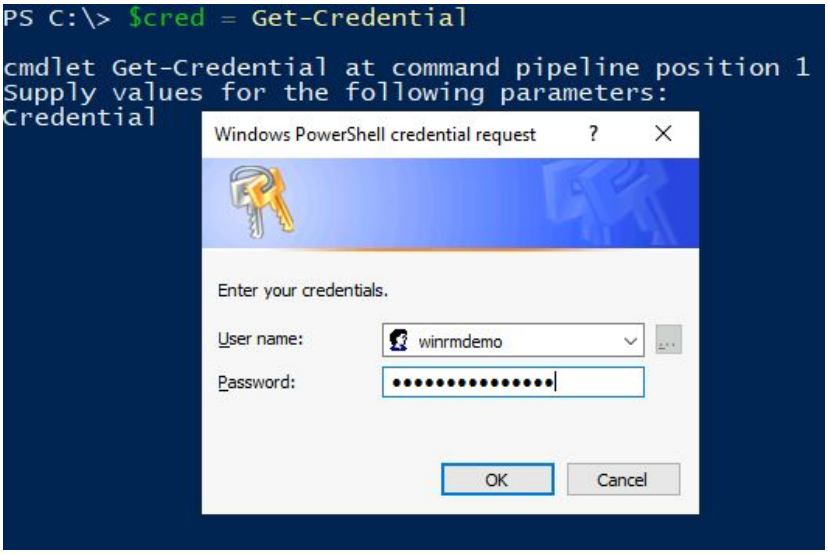
Execute Commands On Remote Server

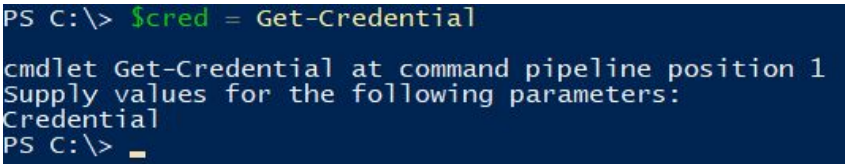
Step 1: We will execute the command using “invoke-command” cmdlet to find all the running processes.

“*The Invoke-Command cmdlet runs commands on a local or remote computer and returns all output from the commands, including errors.*”

Command:
Store the target machine credentials in the $cred variable.

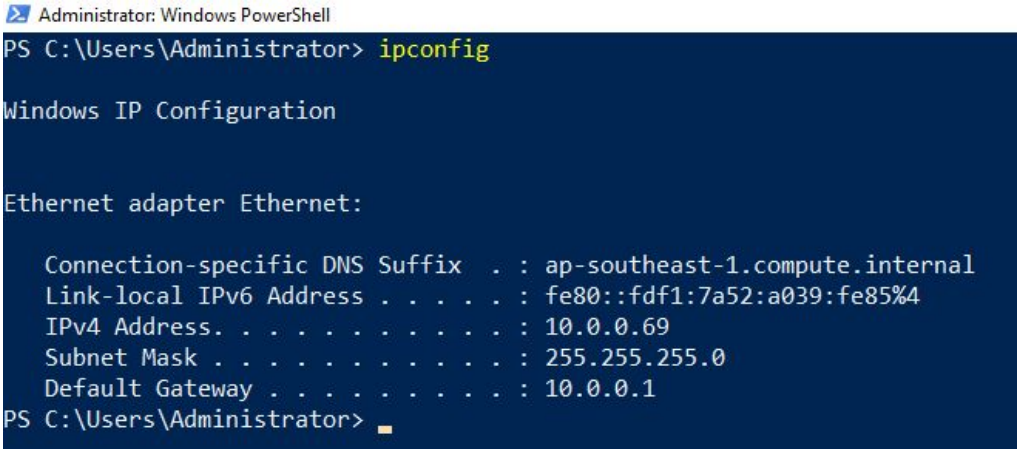
$cred = Get-Credential





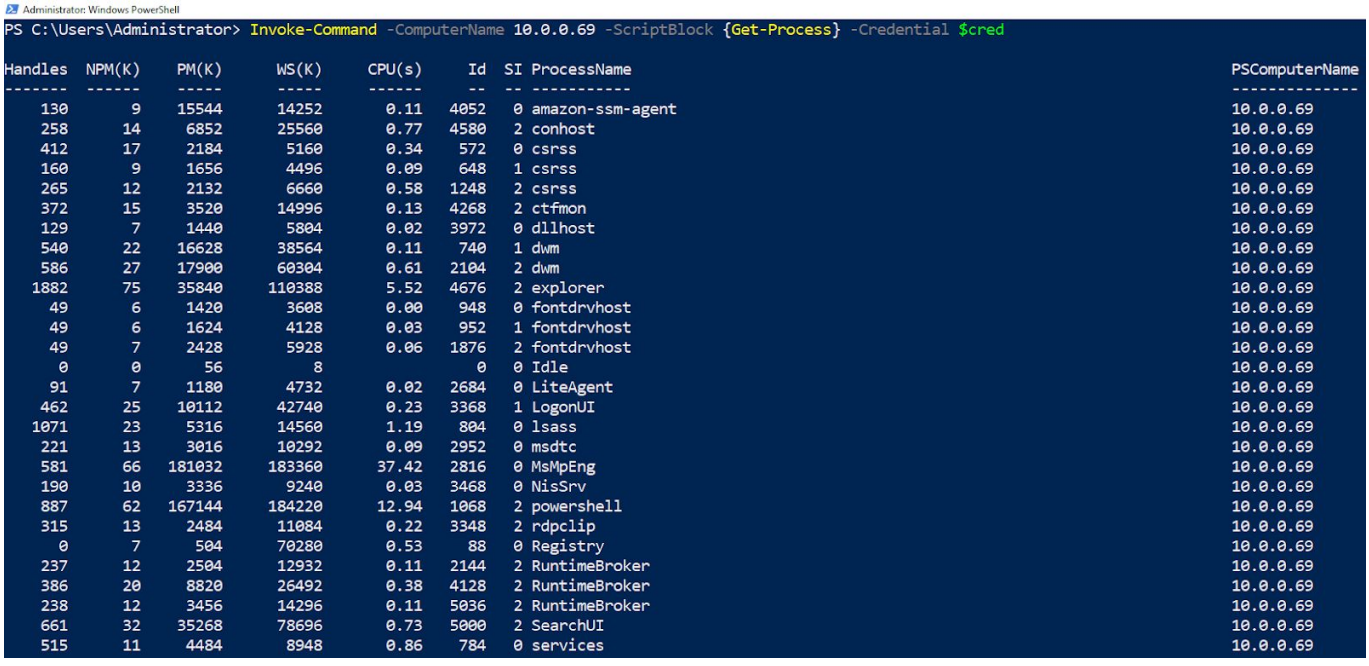
Step 2: Invoke the command cmdlet and check running processes.

Note: Please check the remote server IP address. By running the "**ipconfig**" command.



In my case the remote server IP address is "**10.0.0.69**"

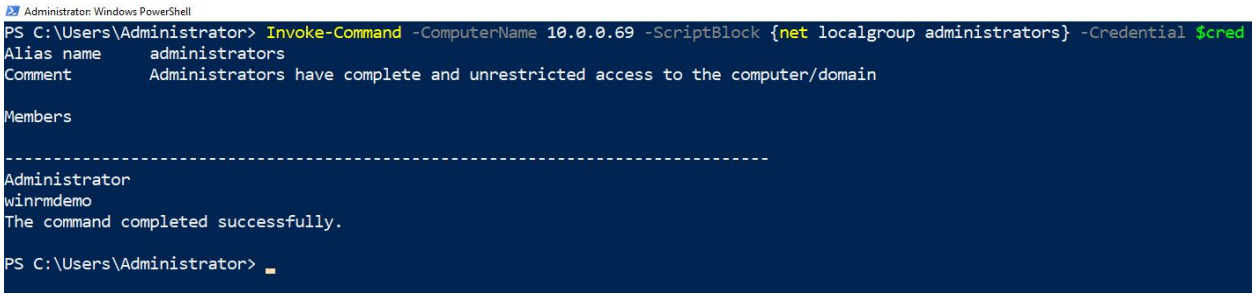
Command: Invoke-Command -ComputerName 10.0.0.69 -ScriptBlock {Get-Process} -Credential $cred



We have successfully executed a command on the remote server.

Checking members of the administrators group.

Command: Invoke-Command -ComputerName 10.0.0.69 -ScriptBlock {net localgroup administrators} -Credential $cred



References:

- Installation and configuration for Windows Remote Management (https://docs.microsoft.com/en-us/windows/win32/winrm/installation-and-configuration-for-windows-remote-management)
- Enable-PSRemoting (https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7)