CYBER RANGERS

ARMORING YOUR BUSINESS

POWERSHELL BASIC SCRIPTING
MODULE 4: POWERSHELL REMOTING

Jan Marek | Cyber Rangers

OSCP | eCPPT | CEH | CHFI | CompTIA Pentest+ | CEI | MVP | MCT | MCC | MCSA | MCSE

Ethical Hacker, Forensic Investigator & Security Engineer

jan@cyber-rangers.com | www.cyber-rangers.com

AGENDA

- What is PowerShell Remoting and how it is related to PowerShell Direct
- How to configure systems to use PowerShell Remoting
- Native PowerShell Remoting features like interactive and non-interactive remoting

POWERSHELL REMOTING VS. POWERSHELL DIRECT

- Remoting over network
  - `Enter-PSSession -ComputerName server123`
- Direct over VMBus
  - Available only from Hyper-V Host to local Hyper-V Guest
  - Does not require network connectivity between Hyper-V Host and Hyper-V Guest
  - `Enter-PSSession -VMName vm123`

HOW TO ENABLE PSREMOTING

- Requires configuration on target
  - `Enable-PSRemoting -Force`
- Network communication over WinRM (WSMan)
- Requires local administrator permissions or JEA for remote connectivity

CMDLET INTEGRATED REMOTING

- Network connectivity over native PowerShell remoting (WinRM)
  - `Get-Process -ComputerName server123`
  - `Get-Service -ComputerName server123`
  - `Get-CimInstance -ComputerName server123 -ClassName win32_Bios`
  - `Get-ADUser -Server server123 -Filter *`
- Network connectivity over specific protocol (i.e. WMI over RPC)
  - `Get-WmiObject -ComputerName server123 -Class win32_bios`

INTERACTIVE REMOTING

- `Enter-PSSession -ComputerName dc`

- With specific credentials
  - `Enter-PSSession -ComputerName dc -Credential (Get-Credential)`
  - ```
    Enter-PSSession -ComputerName dc -Credential (New-Object -TypeName pscredential `
    -ArgumentList ('rangers\da',(ConvertTo-SecureString -String 'P@sswOrd' -AsPlainText -Force)))
    ```
    ```

NON-INTERACTIVE REMOTING

- `Invoke-Command -ComputerName dc -ScriptBlock {Get-Volume}`

- With specific credentials
  - ```
    Invoke-Command -ComputerName dc -Credential (Get-Credential) `
    -ScriptBlock {Get-Volume}
    ```
    ```

NON-INTERACTIVE REMOTING

- Use local variables in remote session: using $args

```
$NameOfProcess = 'svchost'
Invoke-Command -ComputerName dc -ScriptBlock {Write-Host $args[0]} `
 -ArgumentList $NameOfProcess
```

- Use local variables in remote session: using param() block

```
Invoke-Command -ComputerName dc -ScriptBlock {
    param(
        $ValueOf1stArgument
    )
    Write-Host $ValueOf3rdArgument
    } -ArgumentList $NameOfProcess
```

POWERSHELL SESSIONS

- First, define session

```
New-PSSession -ComputerName dc -Credential (Get-Credential) -Name ea
```

- Second, use session

```
Enter-PSSession -Name ea
```

```
Invoke-Command -Session (Get-PSSession -Name ea) -ScriptBlock {$env:username}
```

- Disconnect session, if not used & Connect session, if needed again

```
Disconnect-PSSession -Name ea
```

```
Connect-PSSession -Name ea
```

- Remove session

```
Remove-PSSession -Name ea
```

LAB

POWERSHELL BASIC SCRIPTING
MODULE 4: POWERSHELL REMOTING

Jan Marek | Cyber Rangers

OSCP | eCPPT | CEH | CHFI | CompTIA Pentest+ | CEI | MVP | MCT | MCC | MCSA | MCSE

Ethical Hacker, Forensic Investigator & Security Engineer

jan@cyber-rangers.com | www.cyber-rangers.com

CYBER RANGERS

ARMORING YOUR BUSINESS