Descrição Evento Online

Powershell

H2G2
Consultoria & TI

Sumário

Treinamentos H2G2

Windows Powershell (12hs)

Módulo 1 – Iniciando com Windows Powershell
Visão Geral
Encontrando os comandos
Executando os comandos

Módulo 2 – Trabalhando com Pipeline
Entendendo o Pipeline
Selecionando e Ordenando os objetos
Convertendo, exportando e importando os objetos
Filtrando os objetos no pipeline
Enumerando os objetos no pipeline

Módulo 3 – Entendendo como o Pipeline Funciona
Passagem de dados no Pipeline por valor
Passagem de dados no Pipeline por nome de propriedade

Módulo 4 – Utilizando PSProviders e PSDrives
Usando PSProviders
Usando PSDrives

Módulo 5 – Formatando a Saída
Usando a formatação básica
Usando a formatação avançada
Redirecionando a saída

Módulo 6 – Consultando WMI e CIM
Entendendo WMI e CIM
Consultando informações com WMI e CIM
Alterando informações com WMI e CIM

Módulo 7 – Preparando para Scripts
Usando variáveis
Segurança nos scripts

Módulo 8 – Iniciando com Scripts
Preparando para scripts
Preparando para módulos
Implementando tratamento de erros
Usando scripts
Explorando funcionalidades de scripts

Módulo 9 – Administrando Computadores Remotos
Usando administração remota
Técnicas avançadas de administração remota
Usando sessões

Módulo 10 – Juntando tudo
Provisionamento de um novo servidor

Módulo 11 – Utilizando Jobs Agendados e em Background
Usando Jobs em background
Usando Jobs agendados

Módulo 12 – Utilizando Profiles e Técnicas Avançadas em Powershell
Usando técnicas avançadas em Powershell
Criando Profile Scripts
Trabalhando com credenciais

Metodologia dos Treinamentos de Especialização Profissional

A nossa experiência em capacitação profissional nos permitiu desenvolver uma metodologia diferenciada, que engloba o conhecimento teórico com a aplicação prática de tecnologias para um melhor aproveitamento e resultado final tanto para os profissionais quanto para as empresas.

Os programas de capacitação são desenvolvidos com estudos de caso ou implementação prática, que permitem situações reais do dia-a-dia sejam abordadas durante o treinamento, resultando um melhor aproveitamento do conteúdo abordado.

O objetivo do treinamento é entregar o conhecimento teórico e também a entrega da tecnologia implantada de acordo com as melhores práticas de mercado e do fabricante.

Descrição dos Comandos Powershell do Evento

Consultando o EventViewer (Logs)

Este comando mostra as 10 entradas mais novas do log de Application do Windows.

Get-EventLog –LogName Application –Newest 10

Gerando uma Lista de Serviços

Este comando pega a lista de serviços do Windows e joga a saída para um arquivo texto.

Get-Service | Out-File ServiceList.txt

Exibindo Métodos e Atributos de um Objeto

Este comando exibe a lista de atributos e métodos de um objeto.

Get-Service | Get-Member

Gerando uma Lista de Processos em Execução

Este comando exibe a lista dos processos em execução do Windows.

Get-Process | Select-Object –First 10

Filtrando a Saída de um CmdLet

Este comando exibe a lista dos processos em execução do Windows, filtrando para exibir somente o Nome, ID, Virtual Memory e Paged Memory.

Get-Process | Select-Object –Property Name,ID,VM,PM

Trabalhando com Propriedades Calculadas

Este comando exibe a lista dos processos em execução do Windows, filtrando para exibir somente o Nome, ID, Virtual Memory e Paged Memory utilizando propriedades calculadas.

Get-Process | Select-Object Name,ID, @{n='VirtualMemory';e={$PSItem.PM}}, @{n='PagedMemory';e={$PSItem.PM}}

Trabalhando com Propriedades Calculadas e Formatando a Saída

Este comando exibe a lista dos processos em execução do Windows, filtrando para exibir somente o Nome, ID, Virtual Memory e Paged Memory utilizando propriedades calculadas formatando a saída.

Get-Process | Select-Object Name,ID, @{n='VirtualMemory(MB)';e={'{0:N2}' –f ($PSItem.VM / 1MB)}}, @{n='PagedMemory(MB)';e={'{0:N2}' –f ($PSItem.PM / 1MB) }}

Exportando e Convertendo a Saída

Este comando exibe os CmdLets que começam com o VERBO "Export" e "ConvertTo".

Get-Command –Verb ConvertTo,Export

Estes comandos geram uma lista de serviços e jogam a saída para um arquivo no formato CSV.

Get-Service | ConvertTo-CSV | Out-File Services.csv

Get-Service | Export-Csv Services.csv

Filtrando a Saída por um Atributo

Este comando exibe os serviços do Windows que estão com o status de "Running" (em execução).

Get-Service | Where-Object –FilterScript { $_.Status –eq 'Running'}

Filtrando a Saída por um Atributo

Estes comandos exibem os serviços do Windows que estão com o status de "Running" (em execução).

Get-Service | Where-Object –FilterScript { $_.Status –eq 'Running'}

Get-Service | ? Status –eq Running

Descobrindo os PSDrives

Este comando exibe os PSDrives disponíveis no Powershell.

Get-PSDrive

Consultando usuários desabilitados no AD

Este comando exibe os usuários que estão desabilitados no AD.

Search-ADAccount -AccountDisabled

Consultando último logon do usuário no AD

Este comando exibe a última vez que o usuário fez logon no AD.

Get-ADUser -Identity <username> -properties LastLogOnDate

Consultando usuários que não fazem logon há mais de 90 dias

Este comando exibe os usuários que não fazem logon há mais de 90 dias, filtrando e formatando a saída utilizando propriedades calculadas.

```
$domain = ”ADATUM.COM“
$DaysInactive = 90
$time = (Get-Date).Adddays(-($DaysInactive))

$Users = Get-ADUser -Filter {LastLogonTimeStamp -lt $time} -Properties LastLogonTimeStamp

$Users |
 Select-object Name,
@{n=“Data"; e={[DateTime]::FromFileTime($_.lastLogonTimestamp).ToString(‘yyyy-MM-dd’)}}
```

Consultando informações com WMI e CIM

Estes comandos exibem informações que podem ser consultadas de um computador através do WMI e CIM.

Get-WmiObject -Namespace root\cimv2 –List

Get-CimClass –Namespace root\CIMv2

Consultando informações de hardware com WMI e CIM

Este comando exibe o Service Tag de um computador através do CIM.

Get-CimInstance Win32_SystemEnclosure | Select SerialNumber

Este comando exibe informações da placa de vídeo de um computador através do CIM.

Get-CimInstance Win32_VideoController | select Caption

Este comando exibe informações do disco rídigo de um computador através do CIM.

Get-CimInstance Win32_LogicalDisk –Filter "DriveType=3" |
Select DeviceID,@{n='Size';e={'{0:N2}' –f ($_.Size/1GB)}}

Consultando informações de hardware com WMI e CIM

Este comando altera a política de execução de scripts do Powershell.

Set-ExecutionPolicy RemoteSigned –Force

Restricted (Default) – Previne execução de scripts
AllSigned – Executa apenas scripts assinados
RemoteSigned – Executa todos scripts locais, mas remotos somente assinados
Unrestricted – Executa todos os scripts
Bypass – Segurança gerenciada pelo script

Trabalhando com Scripts

Este script consulta os logs de um computador. O computador é um variável que deve ser passada como parâmetro obrigatório na execução do script.

```
[CmdletBinding()]
Param(
    [Parameter(Mandatory=$True)]
    [string]$ComputerName,

    [int]$EventID = 4624
   )
```

Get-EventLog -LogName Security -ComputerName $ComputerName | Where EventID -eq $EventID | Select -First 50

Script para Gerar Relatório de Logon/Logoff

Este script gera um relatório de usuários que fizeram logon em um computador através da consulta em logs do Windows. Formatando a saída para uma visualização mais amigável e exportando a saída para um arquivo no formato CSV.

```
[CmdletBinding()]
param(
   $ComputerName="localhost"
)

$UserProperty = @{n="User";e={(New-Object System.Security.Principal.SecurityIdentifier $_.ReplacementStrings[1]).Translate([System.Security.Principal.NTAccount])}}
$TypeProperty = @{n="Action";e={if($_.EventID -eq 7001) {"Logon"} else {"Logoff"}}}
$TimeProperty = @{n="Time";e={$_.TimeGenerated}}
$MachineNameProperty = @{n="MachinenName";e={$_.MachineName}}

foreach ($computer in $ComputerName) {
   Get-EventLog System -Source Microsoft-Windows-Winlogon | select $UserProperty,$TypeProperty,$TimeProperty,$MachineNameProperty |
Export-CSV logins.csv
}
```