
PowerShell -With PowerView

---

powershell -ep bypass
. .\PowerView.ps1

- gets a list of all operating systems on the domain
  Get-NetComputer -fulldata | select operatingsystem

- gets a list of all users on the domain
  Get-NetUser | select cn
  Get-NetUser exampleusername

- gets a list of all groups on the domain
  *Get-NetGroup -GroupName **

- gets a list of all share on the domain
  Get-NetShare
  Invoke-ShareFinder

- gets a list of all computers on the domain
  Get-NetComputer

- When was the password last set for the SQLService user?
  Get-NetUser -SPN | ?{$_.memberof -match 'Domain Admins'}

PowerShell -No PowerView

- Display information about a particular command "Get-Command"
  Get-Help | Get-Command

- Search for all possible command by characters, i.e search for the Get-FileHash:
  *Get-Command *hash**

- list all the parameters that we can use with: Get-FileHash or Get-LocalUser
  (Get-Command Get-FileHash).Parameters
  (Get-Command Get-LocalUser).Parameters

- List all the members for the "Get-LocalUser" or "GEt-NetTCPConnection" or "Get-NetTCPConnection"
  Get-FileHash | Get-member
  Get-LocalUser | Get-member
  Get-NetTCPConnection | Get-member

- List all possibility format value for the "State" member of the "Get-NetTCPConnection"
  Get-NetTCPConnection | Get-member
  Get-NetTCPConnection | Format-List -Property State

- Get all Methode
  Get-Command | Get-Member -MemberType Methode
  or we can use it in this way:
  Get-Command | Get-Member
  Get-Command -MemberType Methode

- Creating Objects From Previous cmdlets
  Get-Command | Get-ChildItem
  Get-ChildItem | Select-Object -Property Mode, Name

- Filtering Objects
  Get-Service | Where-Object -Property Status -eq Stopped

- use sort to extract the information more efficiently for the Get-Children
  Get-Children | Sort-Object

Basic Powershell command

- Search for a specific file, i.e interesting-file.txt :

```
Get-ChildItem -Path C:\ -Include *interesting-file.txt* -File -Recurse -ErrorAction SilentlyContinue
```

- Get the content of file, i.e interesting-file.txt:

```
Get-Content "C:\Program Files\interesting-file.txt.txt"
```

- Count Cmdlet on a system:

```
Get-Command | Get-member
Get-Command | Where-Object -Parameter CommandType -eq Cmdlet | measure
or we can use:
(Get-Command Get-Command).Parameters
Get-Command -CommandType Cmdlet | measure
```

- Get the MD5 hash of a file.txt, i.e interesting-file.txt

```
(Get-Command Get-FileHash).Parameters
Get-FileHash -Path "C:\Program Files\interesting-file.txt" -Algorithm MD5
```

- Get the current working directory:

```
Get-Location
Get-Location -Path "C:\Users\Administrator\Documents\Passwords"
```

- Make a request to a web server:

```
Invoke-Webrequest
```

- Decode base 64 bit file, i.e b64.txt:

```
Get-ChildItem -Path C:/ -Include *b64.txt* -Recurse -File
certutil -decode "C:\Users\Administrator\Desktop\b64.txt" out.txt
Get-Content out.txt
```

Enumeration

- Check number of users on the machine:

```
Get-LocalUser
```

- Check user with a specific SID

```
(Get-Command Get-LocalUser).Parameters
Get-LocalUser -SID "S-1-5-21-1394777289-3961777894-1791813945-501"
```

- How many users have their password required values set to False?

```
Get-LocalUser | Get-member
Get-LocalUser | Where-Object -Property PasswordRequired -Match false
```

- How many local groups exist?

```
Get-LocalGroup | measure
```

- get the IP address info:

```
Get-Command *IPadd*
Get-NetIPAddress
```

- How many ports are listed as listening?

```
GEt-NetTCPConnection | Where-Object -Property State -Match Listen | measure
GEt-NetTCPConnection -State "Listen" | measure
```

- How many patches have been applied?

```
Get-Hotfix | measure
```

- When was the patch with ID KB4023834 installed?

```
Get-Hotfix -Id KB4023834
Get-Hotfix -Id "KB4023834"
```

- Find the contents of a backup file.

```
Get-ChildItem -Path C:\ -Include *.bak* -File -Recurse -ErrorAction SilentlyContinue
Get-Content "C:\Program Files (x86)\Internet Explorer\passwords.bak.txt"
```

- Search for all files containing API_KEY

```
Get-ChildItem C:\* -Recurse | Select-String -pattern API_KEY
```

- What command do you do to list all the running processes?

```
Get-Process
```

- What is the path of the scheduled task called new-sched-task?

```
Get-ScheduleTask
Get-ScheduleTask -TaskName new-sched-task
```

- Who is the owner of the C:\

Get-Acl c:/

Basic Scripting

- Check password in email folder:

Get-ChildItem -Path "C:\Users\Administrator\Desktop\emails" -Recurse | Select-String -Pattern password*

Or we can use a script:

```
$path = "C:\Users\Administrator\Desktop\emails\*"
$string_pattern = "password"
$command = Get-ChildItem -Path $path -Recurse | Select-String -Pattern $String_pattern
echo $command
```

- What files contains an HTTPS link?

```
$path = "C:\Users\Administrator\Desktop\emails\*"
$string_pattern = "https://"
$command = Get-ChildItem -Path $path -Recurse | Select-String -Pattern $String_pattern
echo $command
```

Intermediate Scripting

---

https://medium.com/@nallamuthu/powershell-port-scan-bf27fc754585