**What is Powershell?**

Powershell là ngôn ngữ kịch bản của Windows và shell được dựng sẵn bằng thư viện NET

Điều này cho phép Powershell thực thi các hàm .NET trực tiếp từ shell. Đa số lệnh Powershell gọi là cmdlets được viết bằng .NET. Không như ngôn ngữ kịch bản và shell khác, đầu ra của cmdlets là các đối tượng - làm cho Powershell hơi hướng đối tượng.

Điều đó cũng đồng nghĩa với việc chạy cmdlets cho phép bạn thực hiện các hành động trên đối tượng đầu ra (giúp truyền đầu ra từ cmdlet này sang cmdlet khác trở nên tiện lợi hơn). Định dạng chung của cmdlet là Verb-Noun, ví dụ, cmdlet để liệt kê các lệnh gọi là Get-Command

Các động từ phổ biến:

* Get
* Start
* Stop
* Read
* Write
* New
* Out

**Basic Powershell Commands**

**Using Get-Help**

Get-Help hiện thị thông tin về 1 cmdlet. Để hiểu thêm về 1 lệnh cụ thể:

Get-Help Command-Name

Bạn có thể hiểu cách dùng lệnh bằng cách truyền -examples



**Using Get-Command**

Get-Command lấy tất cả cmdlet được cài đặt trên máy tính. Điều tuyệt vời ở cmdlet này là nó cho phép tìm dịnh dạng sau:

Get-Command Verb-\* hoặc Get-Command \*-Noun

Chạy lệnh Get-Command New-\* để xem tất cả cmdlet có động từ là new:



**Object Manipulation**

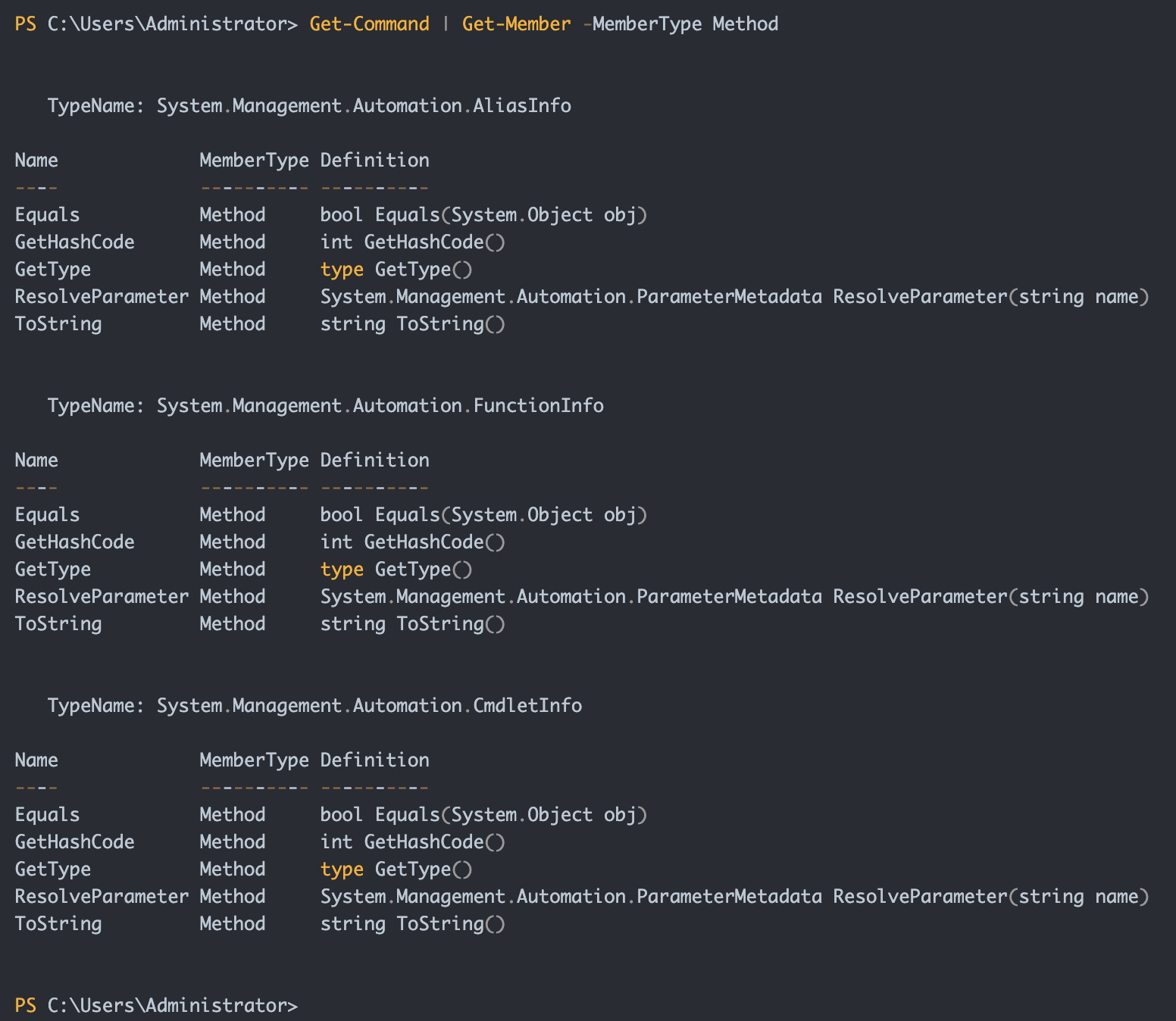
Đầu ra của 1 cmdlet là 1 đối tượng. Nếu muốn thao tác với đầu ra, chúng ta cần tìm hiểu 1 số điều:

* Truyền đầu ra đến cmdlets khác
* Dùng cmdlets đối tượng cụ thể để trích xuất thông tin

Pipeline truyền đầu ra từ 1 cmdlet đến 1 cmdlet khác. Điểm khác nhau với các shell khác là Powershell truyền 1 đối tượng thay vì truyền văn bản hay chuỗi đến lệnh sau pipe. Một đối tượng sẽ chứa các phương thức và thuộc tính. Để xem chi tiết, truyền đầu ra của 1 cmdlet đến cmdlet Get-Member:

Verb-Noun | Get-Member

Ví dụ: Get-Command | Get-Member -MemberType Method

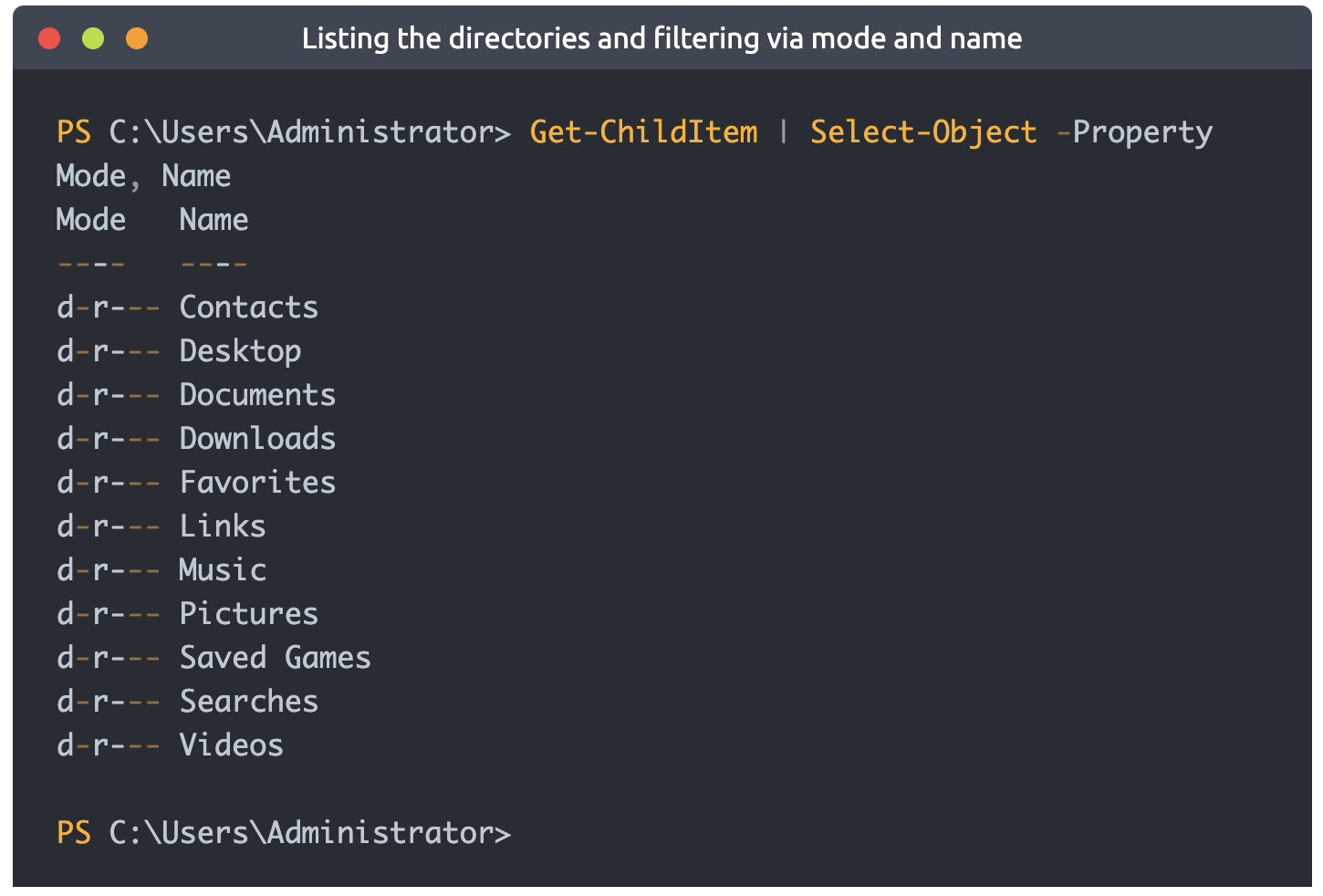


(Get-Command).Version (Version: property)

**Tạo các đối tượng từ cmdlets trước**

Một cách để thao tác với các đối tượng là lấy các thuộc tính từ đầu ra của 1 cmdlet và tạo 1 đối tượng mới. Có thể làm bằng cmdlet Select-Object

Đây là một ví dụ liệt kê thư mục và chỉ chọn thuộc tính mode và name:



Bạn có thể dùng flag sau để chọn thông tin bạn muốn:

* first - gets the first x object
* last - gets the last x object
* unique - shows the unique objects
* skip - skips x objects

**Filtering Objects**

Khi truy suất một đối tượng, bạn có thể muốn các đối tượng phải phù hợp với định dạng bạn muốn. Bạn có thể làm điều này bằng lệnh Where-Object để lọc dựa vào giá trị của thuộc tính.

Định dạng chung cho cmdlet này là:

Verb-Noun | Where-Object -Property PropertyName -operator Value

Verb-Noun | Where-Object {$\_.PropertyName -operator Value}

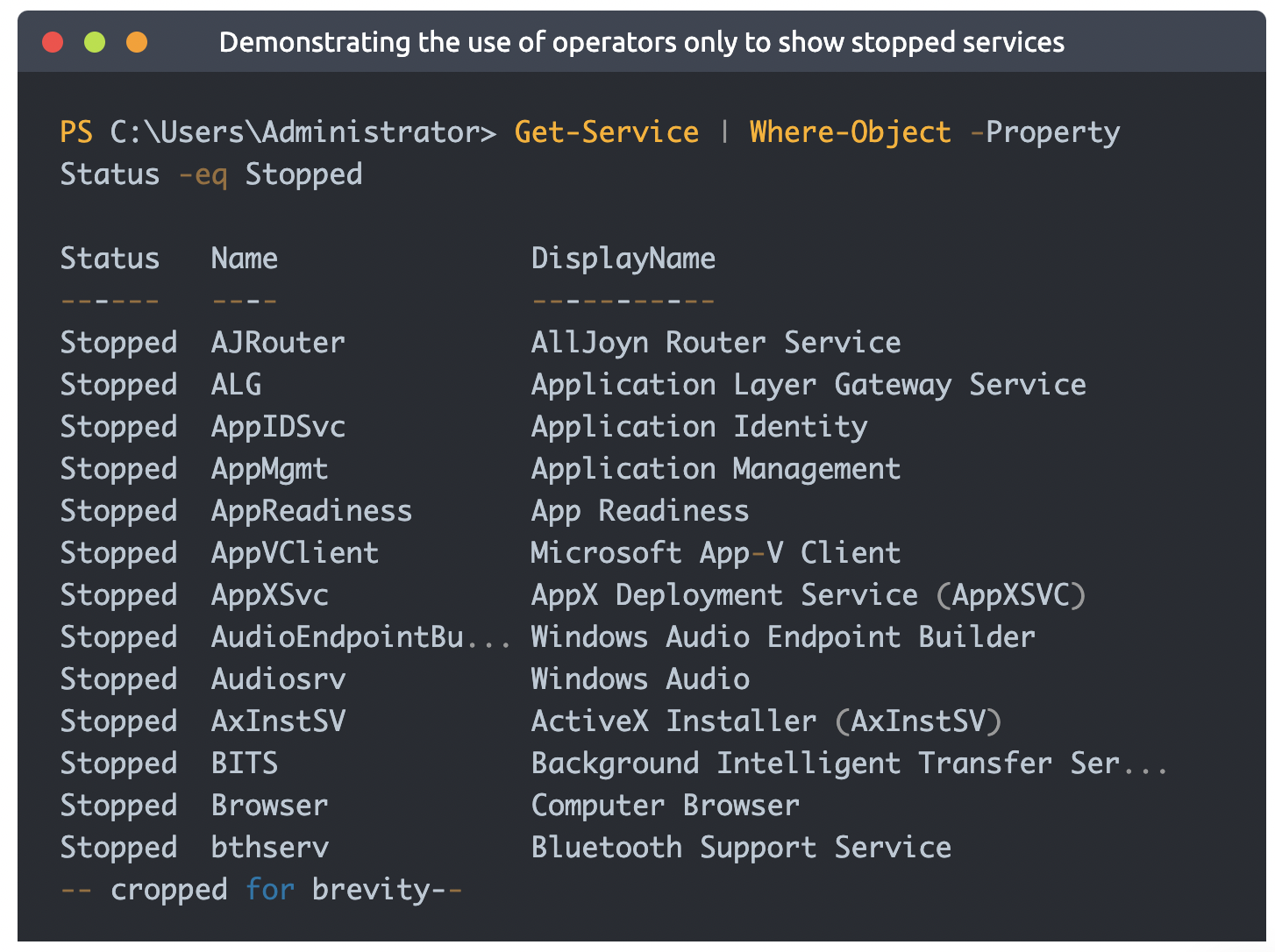
Phiên bản thứ hai dùng toán tử $\_ để lặp lại qua mỗi đối tượng được truyền vào cmdlet Where-Object

Powershell có phân biệt chữ hoa thường, vì vậy không đặt dấu ngoặc kép xung quanh lệnh.

-operator là 1 danh sách các toán tử:

* -EQ: nếu giá trị của thuộc tính bằng với giá trị được chỉ định
* -GT: nếu giá trị của thuộc tính lớn hơn giá trị được chỉ định

Kiểm tra các tiến trình đang bị dừng:

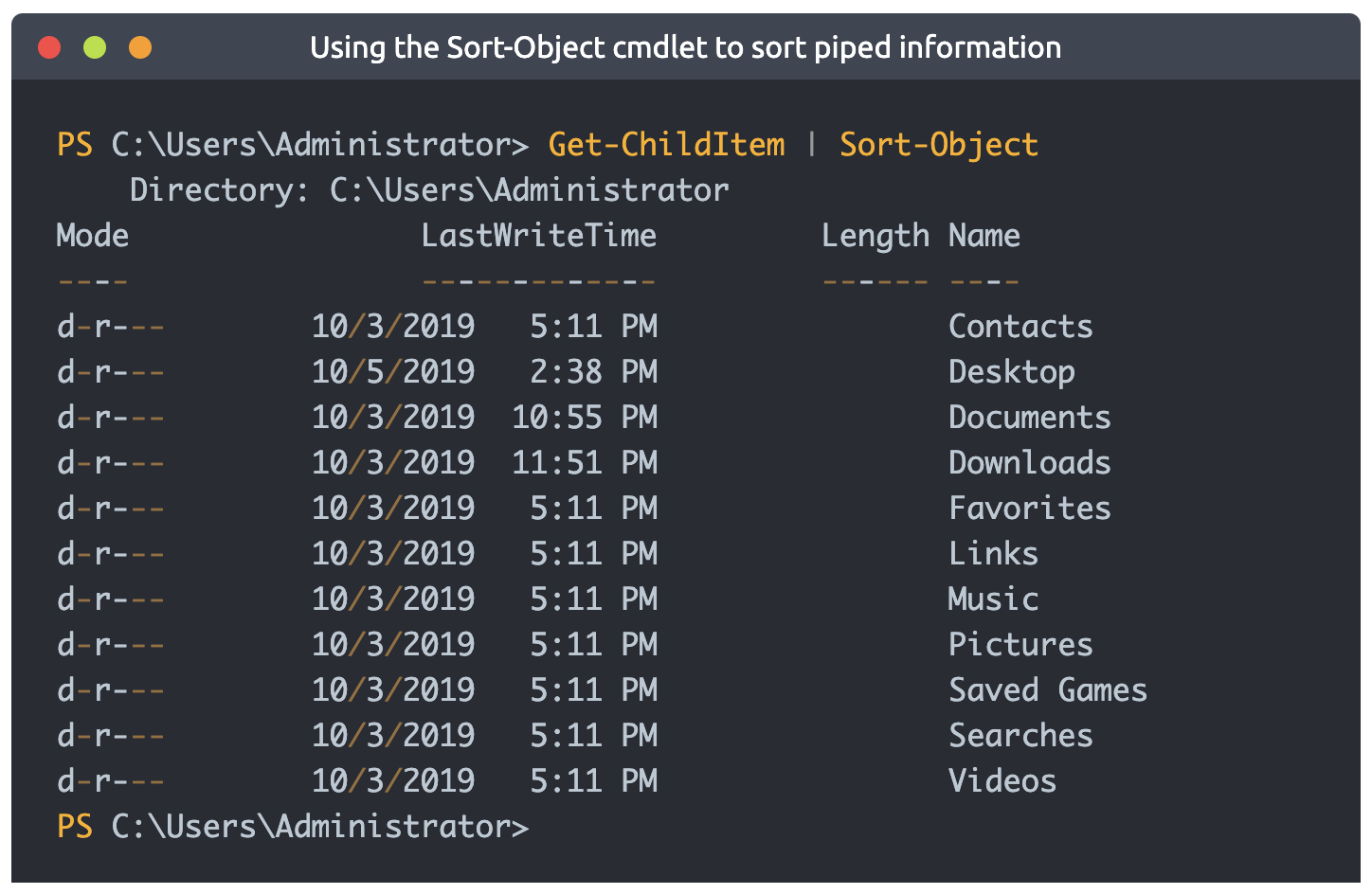


**Sort-Object**

Khi đầu ra của 1 cmdlet có quá nhiều thông tin, bạn cần sắp xếp lại để trích xuất thông tin hiệu quả hơn. Bạn có thể dùng lệnh Sort-Object.

Verb-Noun | Sort-Object

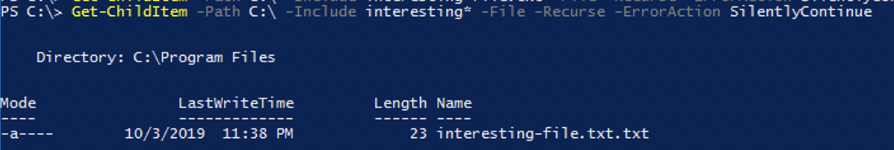
Ví dụ sắp xếp các thư mục:



**Bài Tập**

1. Tìm kiếm file “interesting-file.txt”

Get-ChildItem -Path C:\ -Include interesting\* -File -Recurse -ErrorAction SilentlyContinue

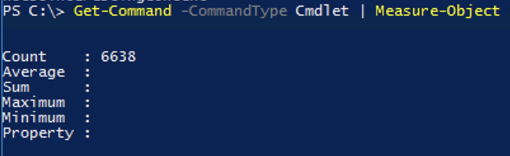


Cách đọc nội dung file:

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

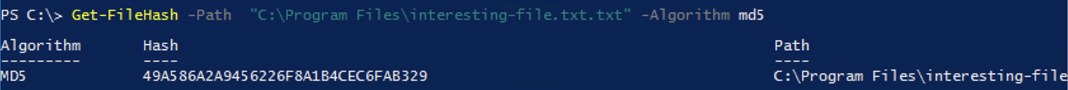


Cách đếm số lệnh được cài đặt trên hệ thống:



Cách lấy hash md5 của 1 file:

Get-FileHash -Path "C:\Program Files\interesting-file.txt.txt" -Algorithm md5

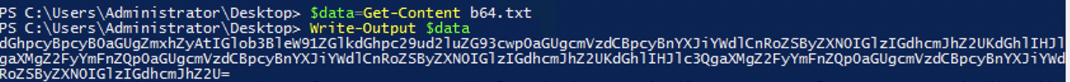


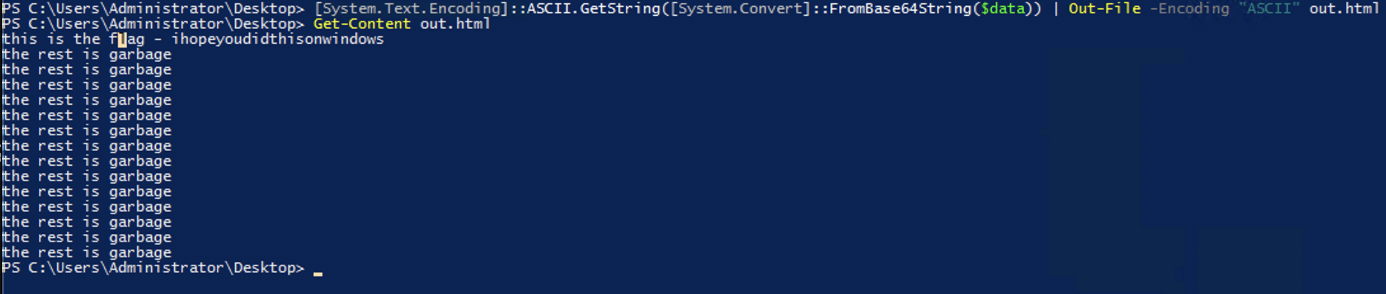
Lệnh để lấy thư mục hiện tại: Get-Location

Kiểm tra đường dẫn tồn tại: Test-Path



Cách giải mã base64 trong Powershell:



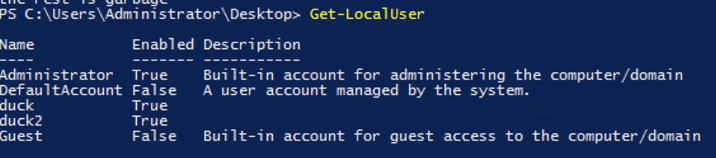


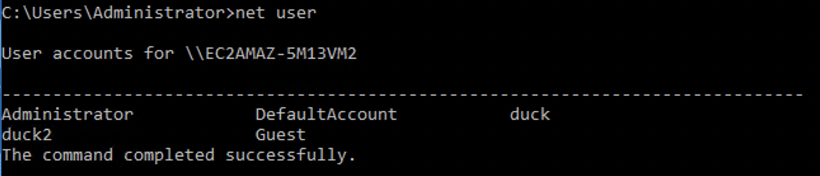
**Enumeration**

Bước đầu tiên khi bạn truy cập vào bất kỳ máy ảo nào là liệt kê. Chúng ta sẽ liệt kê những thông tin sau:

* users
* basic networking information
* file permissions
* registry permissions
* scheduled and running tasks
* insecure files

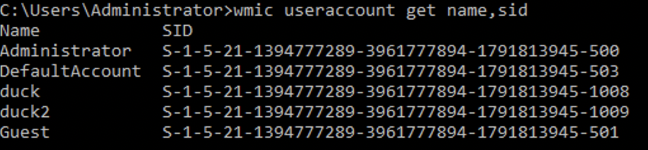
Có bao nhiêu người dùng trên máy ảo:



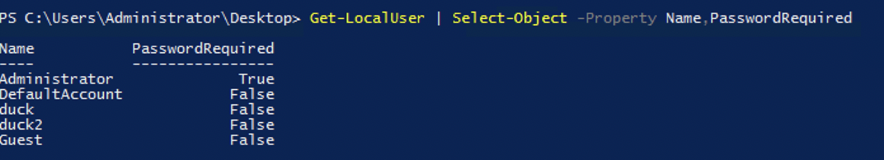


Kiểm tra SID của người dùng:

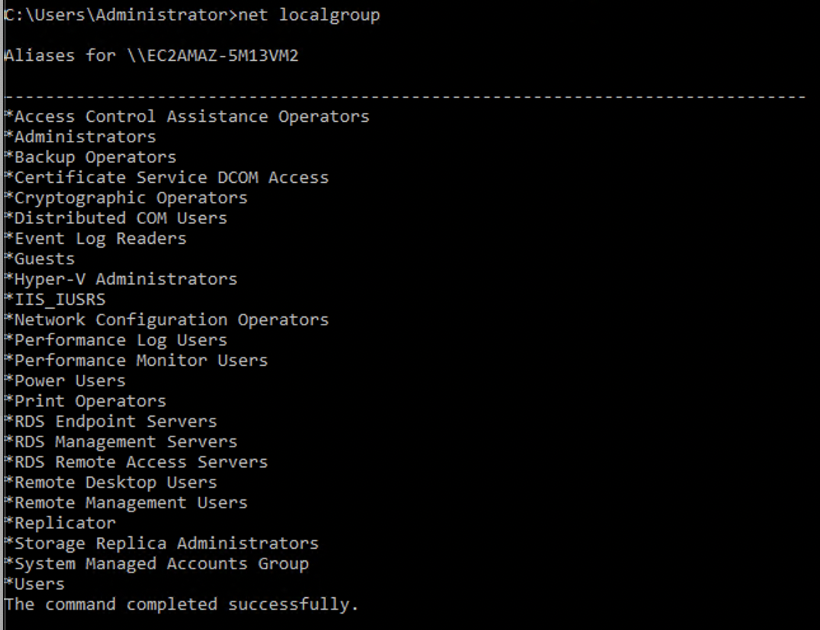


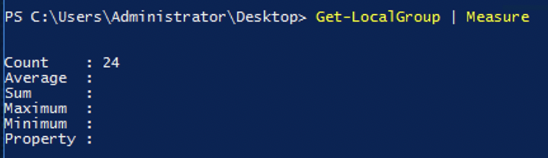


User with NoPassword



Có bao nhiêu local group tồn tại?





Lệnh bạn dùng để lấy địa chỉ Ip:

Get-NetIPAddress

ipconfig

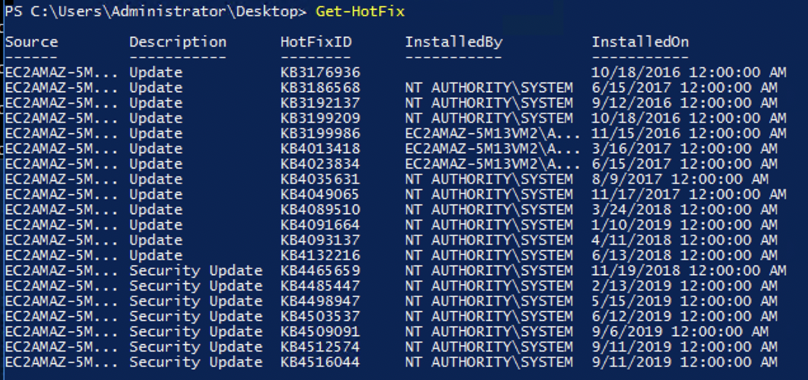
Số lượng cổng đang lắng nghe:



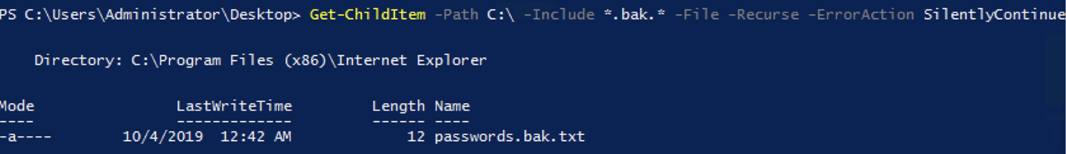
netstat -a

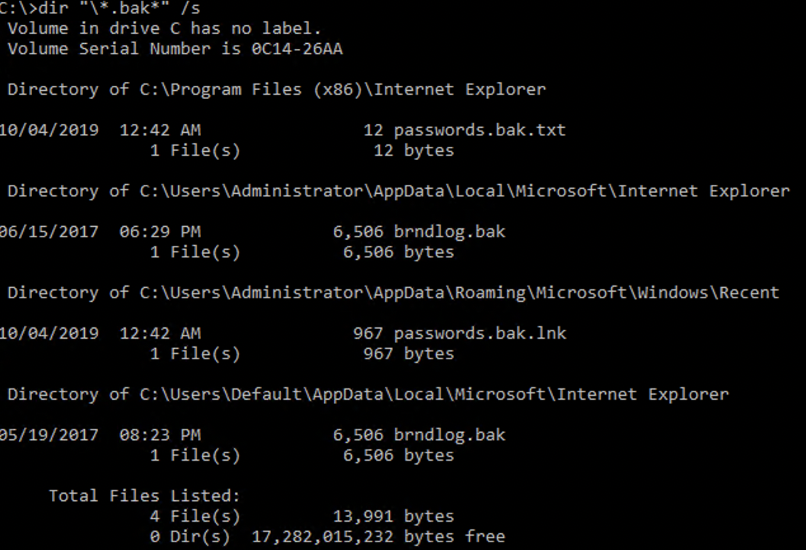
Kiểm tra số bản vá được cài đặt:

wmic qge list



Đọc backup file

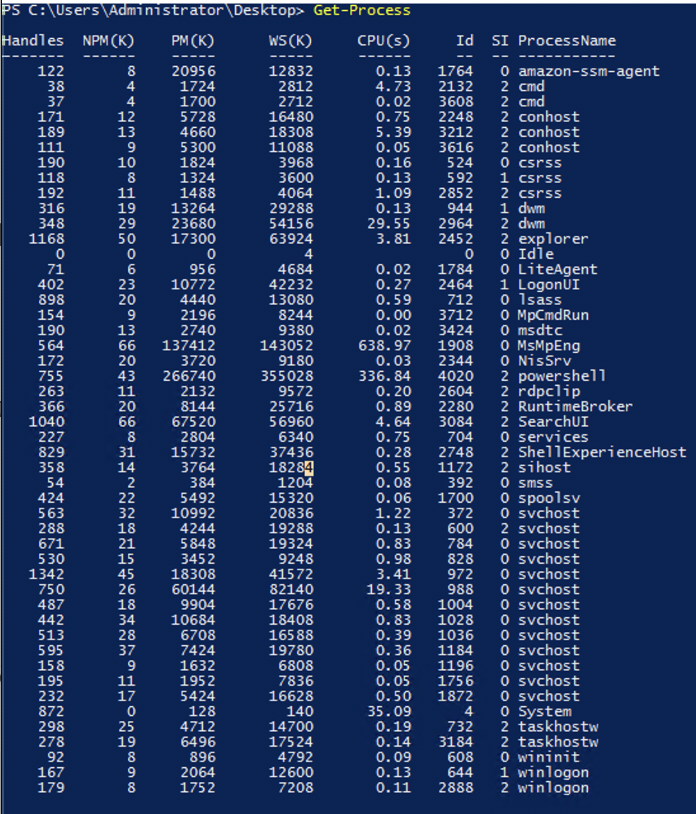




Tìm kiếm tất cả files chứa API\_KEY

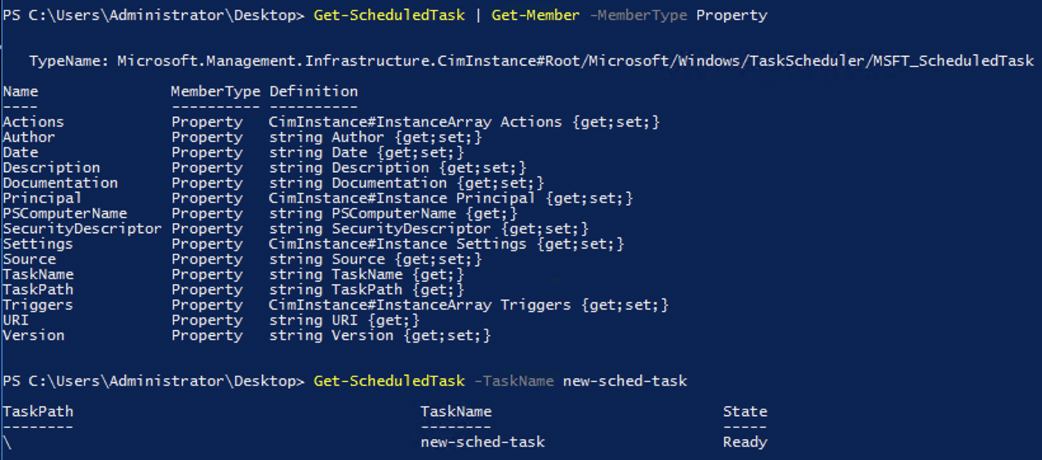
Bạn phải dùng lệnh grep. Sự thay thế của Powershell cho grep là: Select-String, Cmd là findstr

Liệt kê các tiến trình đang chạy:

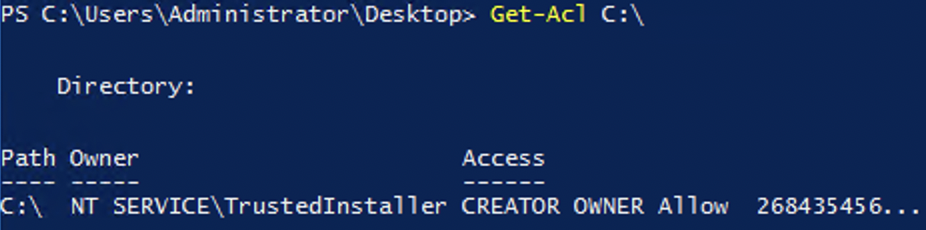


tasklist (Cmd)

Đường dẫn của scheduled task gọi là new-sched-task:



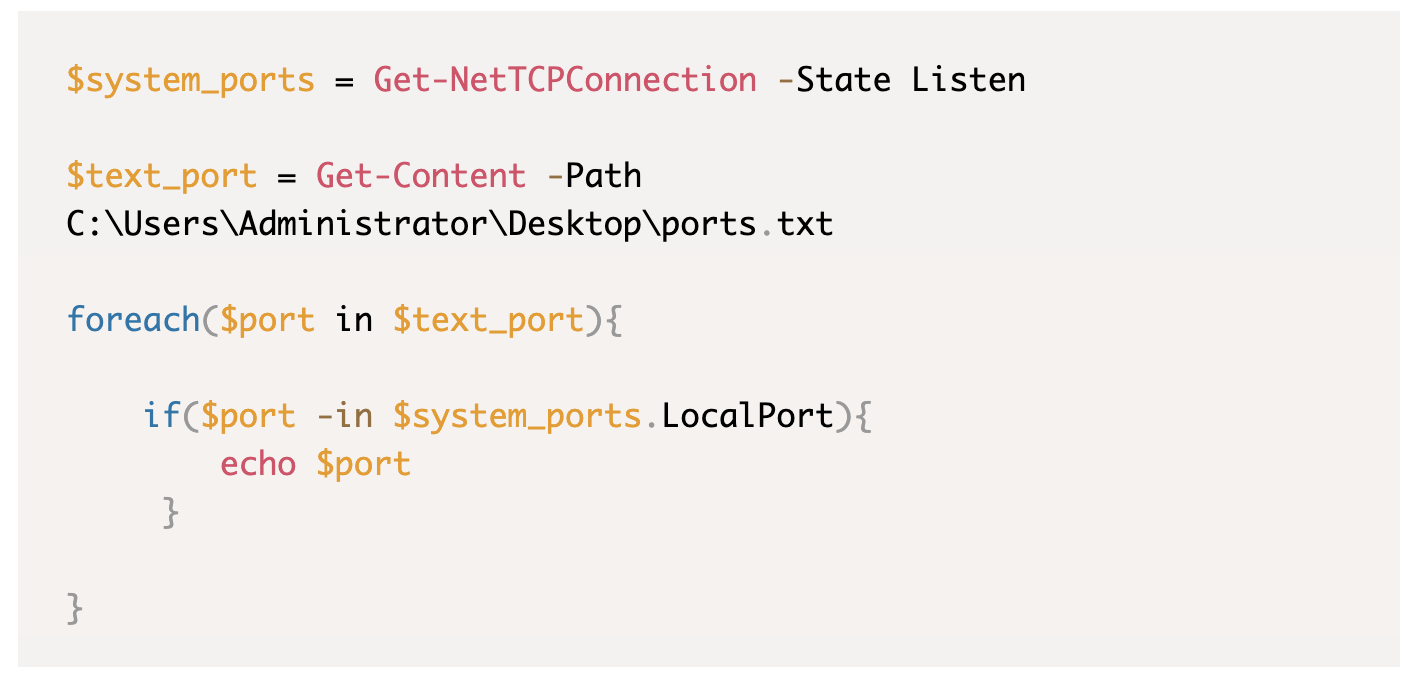
Ai là chủ nhân của thư mục C:\



**Basic Scripting Challenge**

Chúng ta đã biết thực thi lệnh Powershell, hãy cố viết và chạy 1 script để thực hiện các hành động phức tạp và mạnh mẽ hơn.

Ta sẽ dùng Powershell ISE (Powershell Text Editor) để viết script. Chúng ta sẽ dùng danh sách các cổng để kiểm tra cổng nào đang lắng nghe. Mở script listening-ports.ps1 trên Desktop bằng Powershell ISE. Script Powershell có phần mở rộng là .ps1



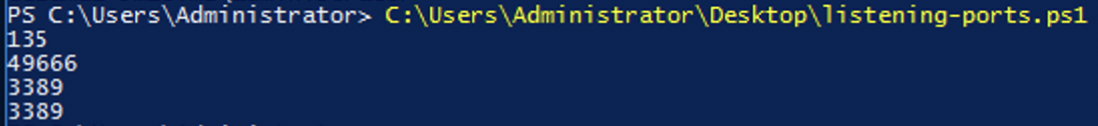
Dòng đầu tiên, chúng ta muốn có 1 danh sách các cổng đang lắng nghe. Chúng ta lưu vào biến $system\_ports

Ở dòng tiếp theo chúng ta muốn đọc 1 danh sách các cổng từ file. Chúng ta lưu vào biến $text\_port

Bước tiếp theo là lặp qua tất cả port đọc trong file:

foreach($new\_var in $existing\_var{}

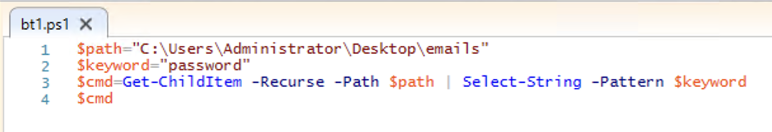
Ta dùng câu lệnh if với toán tử -in để kiểm tra cổng trong $text\_port có tồn tại trong $system\_ports không.

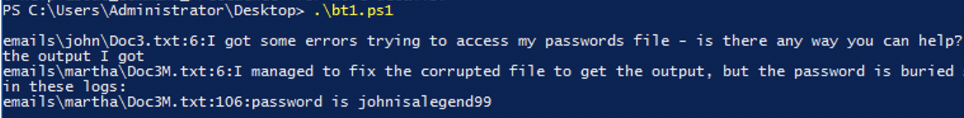


**Bài tập:**

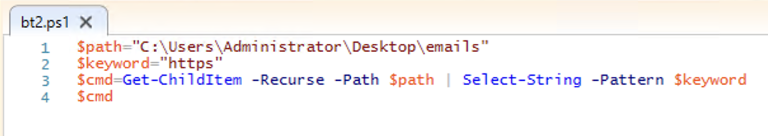
Thư mục emails trên Desktop chứa bản sao email của John, Martha và Mary gửi cho nhau.

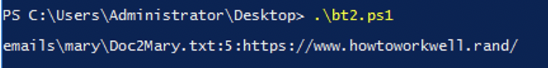
File nào chứa mật khẩu:





File nào chứa đường dẫn HTTPS:





**Intermediate Scripting**

Bạn tìm thấy bao nhiêu cổng đang mở giữa 130 và 140?

