10 lệnh PowerShell cơ bản hữu ích dành cho Windows

Trong những năm vừa qua, Microsoft đã cố gắng để biến PowerShell thành 1 trong những công cụ quản lý toàn diện dành cho Windows. Hầu hết các hệ thống server của Microsoft đều khuyến cáo mọi người sử dụng *PowerShell*, có thể thực hiện được rất nhiều chức năng mà không cần phải can thiệp bằng *Command Prompt* như trước kia. Đối với mỗi người quản trị hệ thống Windows, họ cần phải làm quen và sử dụng *PowerShell* từ những bước cơ bản ban đầu. Sau đây, chúng tôi sẽ giới thiệu với các bạn 10 lệnh không thể thiếu khi bắt tay làm quen với PowerShell.

1. Get-Help:

Đầu tiên và trước tiên, tất cả mọi người cần tìm hiểu về bất cứ câu lệnh, cú pháp nào là Get-Help. Ví dụ nếu muốn kiểm tra về *Get-Process* thì gõ lệnh như sau:

```
Get-Help -Name Get-Process
```

và Windows sẽ hiển thị đầy đủ cú pháp. Bên cạnh đó, *Get-Help* còn

được sử dụng đi kèm với danh từ và động từ riêng rẽ, ví dụ với lệnh động từ Get:

```
Get-Help -Name Get-*
```

2. Set-ExecutionPolicy:

Mặc dù bạn có thể tạo và thực thi các đoạn mã PowerShell khác nhau, nhưng ở chế độ mặc định Microsoft đã tắt bỏ tính năng này để phòng tránh các loại mã độc khác nhau khi xâm nhập vào hệ thống có thể tự kích hoạt và khởi động trong môi trường PowerShell. Người sử dụng có thể áp dụng lệnh *Set-ExecutionPolicy* để thiết lập các mức bảo mật khác nhau, cụ thể có 4 lựa chọn phù hợp:

- **Restricted**: đây là chính sách mặc định của hệ thống, các câu lệnh PowerShell đều bị khóa, người sử dụng chỉ có thể nhập lệnh nhưng không thực thi được.

- **All Signed**: nếu bạn hoặc người quản trị thiết lập mức All Signed thì các đoạn mã sẽ được thực thi, nhưng chỉ áp dụng với những thành phần được chỉ định rõ ràng.

- **Remote Signed**: chính sách bảo mật khi ở mức này, bất cứ đoạn mã PowerShell được tạo bên trong hệ thống local sẽ được phép hoạt động. Còn những mã tạo qua remote thì chỉ được phép chạy khi gán

thuộc tính đầy đủ.

- **Unrestricted**: không áp dụng bất cứ hình thức ngăn cấm nào trong hệ thống.

Cú pháp chung của lệnh này bao gồm tên của lệnh Set-ExecutionPolicy đứng sau chính sách. Ví dụ như sau:

```
Set-ExecutionPolicy Unrestricted
```

3. Get-ExecutionPolicy:

Nếu bạn phải làm việc trên hệ thống server không quen thuộc, thì cần phải biết chính sách mức chính sách bảo mật nào đang được áp dụng trên đó trước khi thực thi bất cứ câu lệnh hoặc đoạn mã nào đó. Để làm việc này, các bạn sử dụng lệnh *Get-ExecutionPolicy*.

4. Get-Service:

Câu lệnh này sẽ liệt kê tất cả các dịch vụ đã được cài đặt trên hệ thống. Nếu cần tìm hiểu kỹ hơn về 1 dịch vụ bất kỳ nào đó, hãy thêm **-Name** và tên của dịch vụ đó, Windows sẽ hiển thị đầy đủ chi tiết, tình trạng liên quan.

5. ConvertTo-HTML:

Khi cần xem hoặc tạo báo cáo đầy đủ về thông tin, tình trạng hiện

thời của toàn bộ hệ thống, hãy sử dụng chức năng chuyển đổi định dạng *ConvertTo-HTML*. Trước tiên, các bạn cần chỉ định đường dẫn file chuyển đổi sau khi dùng *ConvertTo-HTML*, tham số **-Property** có nhiệm vụ khởi tạo thuộc tính trong file HTML, sau cùng là đặt tên cho file chuyển đổi. Cú pháp chung của lệnh này như sau:

```
Get-Service | ConvertTo-HTML -Property Name, Status > C:\services.htm
```

6. Export-CSV:

Sau khi tạo báo cáo bằng HTML dựa trên dữ liệu của PowerShell, bạn cũng có thể trích xuất dữ liệu PowerShell thành file CSV để sử dụng với Microsoft Excel. Cú pháp chung cũng tương tự như câu lệnh trên:

```
Get-Service | Export-CSV c:\service.csv
```

7. Select-Object:

Việc sử dụng các lệnh trên để tìm hiểu về hệ thống, bạn sẽ phát hiện ra rằng có rất nhiều thuộc tính kèm trong file CSV. Tính năng này tỏ ra thực sự hữu ích khi cho phép người sử dụng chỉ định những thuộc tính cố định trong các mối liên kết. Ví dụ, để tạo file CSV có chứa tên của các dịch vụ riêng biệt trong hệ thống và tình trạng đi kèm, các

bạn có thể sử dụng cú pháp chung như sau:

```
Get-Service | Select-Object Name, Status | Export-CSV c:\service.csv
```

8. Get-EventLog:

Người sử dụng hoàn toàn có thể dùng *PowerShell* để phân tích các sự kiện xảy ra trong hệ thông qua file log. Có 1 vài tham số cụ thể đối với các dịch vụ khác nhau, nhưng hãy thử nghiệm bằng cách thêm -Log ở phía trước tên file log. Ví dụ, để xem file log *Application* thì các bạn sử dụng lệnh sau:

```
Get-EventLog -Log "Application"
```

Tuy nhiên, cú pháp này không thực sự phổ biến trong các hoàn cảnh làm việc, khi mà người sử dụng có thể lựa chọn giữa phương pháp lưu báo cáo thành định dạng HTML hoặc CSV.

9. Get-Process:

Đi kèm với lệnh *Get-Service* để hiển thị danh sách các dịch vụ hiện thời của hệ thống, cú pháp *Get-Process* được dùng để liệt kê toàn bộ các tiến trình đang hoạt động.

10. Stop-Process:

Đôi khi, có những dịch vụ trong hệ thống bị rơi vào trạng thái “treo”.

Đối với những trường hợp như vậy, hãy dùng lệnh *Get-Process* để xác định tên hoặc ID chính xác của tiến trình đó, và tắt tiến trình này bằng lệnh *Stop-Process*. Ví dụ, để tắt hoạt động của chương trình *NotePad* thì gõ lệnh như sau:

```
Stop-Process -Name notepad
```

```
Stop-Process -ID 2668
```

Nhưng hãy lưu ý vì ID của các tiến trình sẽ thay đổi theo hệ thống.

T.Anh (theo Tech Republic)