**Thiết Lập Cho xp\_cmdshell**

[Vũ Huy Tâm](http://www.sqlviet.com/blog)

SQL Server là hệ quản trị dữ liệu, khả năng chính của nó là lưu trữ và truy xuất dữ liệu. Tuy nhiên đôi lúc bạn cần thực hiện một số thao tác của windows từ SQL Server, ví dụ khi bạn cần viết sql script để copy file từ folder này sang folder khác; hoặc tự động restore từ file backup mới nhất dựa theo ngày tạo (tên file không biết trước). SQL Server cung cấp thủ tục xp\_cmdshell trong master database để làm việc này.  
Để có thể gọi đến xp\_cmdshell đòi hỏi một vài bước thiết lập.  
Nếu bạn là DBA với quyền sysadmin, bạn cần cấu hình lại server để bật xp\_cmdshell (khi SQL Server mới cài đặt tính năng này bị tắt):

EXEC SP\_CONFIGURE 'show advanced option', 1

RECONFIGURE

GO

EXEC SP\_CONFIGURE 'xp\_cmdshell', 1

RECONFIGURE

GO

Sau đó, với quyền sysadmin bạn có thể dùng xp\_cmdshell:

EXEC xp\_cmdshell 'dir c:\'

Với sysadmin như vậy là xong. Nhưng nếu một người khác (giả sử với sql login là xeko) không có quyền sysadmin mà đăng nhập vào và thử xp\_cmdshell, kể cả sau khi bạn đã cấu hình lại server hệ thống vẫn tiếp tục báo lỗi. Có hai vấn đề bạn cần giải quyết.  
Thứ nhất, đây là một tác vụ nằm ngoài SQL Server, vì thế mỗi khi có yêu cầu như vậy SQL Server sẽ cần tạo 1 tiến trình trong window để window thực hiện. Tiến trình này cần một window account để có thể chạy, vậy là account nào? Nếu bạn là sysadmin, SQL Server sẽ lấy luôn account đang chạy SQL Service. Nhưng với một non-sysadmin như xeko, SQL Server không đủ tin tưởng để dùng account đó, và như vậy là không có account nào, tất nhiên window sẽ từ chối thực hiện. Để lấp vào khoảng trống đó, bạn cần tạo một proxy account và ánh xạ đến một window account thực, để chỉ thị cho SQL Server khi chạy xp\_cmdshell hãy lôi account này ra dùng:

EXEC sp\_xp\_cmdshell\_proxy\_account 'thu-pc\thu','password' --thay bằng domain account và password thật

Sau khi chạy lệnh trên, trong Management Studio nếu bạn mở rộng node Security/Credentials, bạn sẽ thấy một dòng ##xp\_cmdshell\_proxy\_account##.  
Thứ hai, bạn phải cấp quyền thực hiện thủ tục xp\_cmdshell cho xeko:

USE master

GO

--Tạo user xeko trong master database nếu chưa có

CREATE USER xeko FROM LOGIN xeko WITH default\_schema = dbo

GO

GRANT EXECUTE ON xp\_cmdshell TO xeko

Và từ đó xeko có thể chạy xp\_cmdshell. Một vài lưu ý sau cùng:

* Bạn có thể truyền tham số cho xp\_cmdshell như một thủ tục thông thường:
* DECLARE @cmd VARCHAR(100), @drive CHAR(2)
* SET @drive = 'C:'
* SET @cmd = 'dir '+@drive

EXEC xp\_cmdshell @cmd

* Các lệnh bạn chạy trong xp\_cmdshell đều được thực hiện bằng account 'thu-pc\thu và tuân theo quyền hạn của account này. Ví dụ nếu thu không có quyền ghi lên ổ D, lệnh sau sẽ trả về thông báo không thực hiện được:

EXEC xp\_cmdshell 'copy C:\file.txt D:\'

* xp\_cmdshell tạo ra một lỗ hổng về bảo mật, theo đó hacker nếu chiếm được quyền vào SQL Server và thực hiện xp\_cmdshell (tất nhiên phải thông qua một login có quyền exec xp\_cmdshell như xeko) có thể tiếp tục phá hoại hoặc đánh cắp dữ liệu nằm ngoài SQL Server. Đây là một đe dọa đáng kể bạn cần lưu ý và cân nhắc giữa sự tiện lợi và an ninh hệ thống. Bạn có thể giảm bớt nguy cơ này bằng cách chỉ bật xp\_cmdshell trên các SQL Server không tương tác trực tiếp với web server, và hạn chế tối đa quyền trên mạng cho 'thu-pc\thu.