

Injection trong MySQL và SQL

Nếu bạn nhận User Input thông qua một Webpage và chèn nó vào trong một SQL Database, thì tình cờ, bạn đã mở rộng cửa bảo mật ra bên ngoài, mà được biết đến với tên gọi là SQL Injection.

Chương này sẽ hướng dẫn bạn cách ngăn cản tình huống này xảy ra và giúp bạn bảo vệ Script của bạn và các lệnh SQL trong Server-Side Script như PERL Script.

Injection thường xảy ra khi bạn yêu cầu input từ một người dùng, như tên của họ, và thay vì cung cấp tên, họ cung cấp cho bạn một lệnh SQL mà bạn sẽ chạy trên Database của mình mà không hay biết.

Đừng bao giờ tin vào dữ liệu được cung cấp bởi người dùng, xử lý dữ liệu này, và như một qui tắc, điều này được thực hiện bởi Pattern Matching (so khớp mẫu).

Trong ví dụ dưới, name bị giới hạn là các ký tự chữ-số cộng với dấu gạch dưới và có độ dài từ 8 đến 20 ký tự (bạn có thể sửa đổi nếu thấy cần thiết).

```
if (preg_match("/^\w{8,20}$/", $_GET['username'], $matches)) {    $result =
mysql_query("SELECT * FROM nhanvienIT                              WHERE
username=$matches[0]"); }  else  {    echo "Ten su dung khong duoc chap nhan";
}
```

Để minh họa vấn đề, bạn xem phần trích sau:

```
// gia su co input nhu sau: $name = "Thanh'; DELETE FROM nhanvienIT;";
mysql_query("SELECT * FROM nhanvienIT WHERE name='{$name}'");
```

Lời gọi hàm được xem như để lấy một bản ghi từ bảng NHANVIEN, với cột name so khớp với name đã được xác định bởi người dùng. Thông thường, $name sẽ chỉ chứa các ký tự chữ-số và có thể có khoảng trống. Nhưng ở đây, bằng việc phụ thêm một truy vấn hoàn toàn mới tới $name, lời gọi tới Database sẽ gây ra vấn đề lớn: truy vấn DELETE bị tiêm vào sẽ xóa tất cả bản ghi từ bảng NHANVIEN.

May mắn là, nếu bạn sử dụng MySQL, hàm mysql_query() không cho phép Query Stacking hoặc thực thi nhiều truy vấn SQL trong một lời gọi hàm đơn. Nếu bạn nỗ lực để thực hiện nhiều truy vấn, lời gọi hàm sẽ thất bại.

Tuy nhiên, với PHP Database, ví dụ như SQLite và PostgreSQL, lại cho thực hiện nhiều truy vấn, thực thi tất cả truy vấn được cung cấp trong một chuỗi và điều này tạo ra một vấn đề rất nghiêm trọng.

Ngăn chặn SQL Injection

Bạn có thể xử lý tất cả Escape Character một cách khéo léo trong các ngôn ngữ Scripting như PERL và PHP. MySQL extension cho PHP cung cấp hàm mysql_real_escape_string() để tránh các ký tự được nhập vào mà có ý nghĩa đặc biệt với MySQL.

```
if (get_magic_quotes_gpc())  {   $name = stripslashes($name); } $name =
mysql_real_escape_string($name); mysql_query("SELECT * FROM nhanvienIT WHERE
name='{$name}'");
```

LIKE Quandary trong MySQL

Để định vị một LIKE Quandary, một kỹ thuật do người dùng tạo phải chuyển đổi các ký tự '%' và '_' do người dùng cung cấp thành literal (hằng). Sử dụng hàm addcslashes(), một hàm mà giúp bạn xác định một dãy ký tự để thoát.

```
$sub = addcslashes(mysql_real_escape_string("%something_"), "%_"); // $sub ==
\%something\_ mysql_query("SELECT * FROM messages WHERE subject LIKE
'{$sub}%'");
```