BAB XVI
SQL INJECTION

Pada bab ini akan dibahas mengenai apa itu SQL injection, cara melakukannya dan cara pencegahannya. Meskipun dalam bab ini dijelaskan cara melakukan teknik ini, akan tetapi penulis bukan bertujuan untuk mendidik orang untuk menjadi penjahat dalam dunia virtual, melainkan memberikan pemahaman supaya orang yang membaca menjadi *eling lan waspodo*.

Apa itu SQL Injection

SQL injection merupakan perbuatan orang yang memberikan perintah SQL untuk dijalankan di mesin server SQL tanpa sepengetahuan Anda sebagai administrator. Proses injection biasanya dilakukan orang ketika memasukkan input melalui form dengan perintah atau kode tertentu. Berikut ini adalah contohnya:

```
<?
$nama = "agus";
$query = "SELECT * FROM customers WHERE username = '$name'";
echo "Query Normal: " . $query . "<br>";

// input user yang menggunakan SQL injection
$name = "' OR 1 OR '1' = '1";

$query = "SELECT * FROM customers WHERE username = '$name'";

echo "Query Injection: " . $query;
?>
```

Tampilan script di atas adalah

```
Query Normal: SELECT * FROM customers WHERE username = 'agus'
Query Injection: SELECT * FROM customers WHERE username = '' OR 1' OR
'1' = '1'
```

Apabila query normal dijalankan di server SQL tentu saja tidak ada masalah karena akan mencari record dari tabel `customers` yang terkait dengan username `agus`. Query tersebut akan mendapatkan data yang diinginkan apabila memang terdapat user `agus` dan tidak akan mendapatkan data yang diinginkan apabila tidak terdapat user tersebut.

Sedangkan untuk query injection pasti akan selalu mendapatkan paling sedikit satu record/ data dari tabel customers. Mengapa hal ini bisa terjadi? Hal ini dikarenakan dalam query injection terdapat perintah OR 1 yang selalu bernilai TRUE.

Berikut ini adalah contoh mekanisme SQL injection yang dapat terjadi pada proses login.

Misalkan terdapat tabel `user` yang digunakan untuk menyimpan data user, yang di dalamnya terdapat 2 field yaitu username dan password. Misalkan juga terdapat 2 user yaitu JOKO (password : JOKO) dan AMIR (password : AMIR).

Diberikan form login sbb.

Login.php

```
<form method=post action=submit.php>
Nama user <input type=text name=username><br>
Password <input type=text name=password>
<input type=submit name=submit value=Submit>
</form>
```

Dan file **submit.php** sbb.

```
<?

mysql_connect("localhost","root","root");
mysql_select_db("test");

$username = $_POST['username'];
$password = $_POST['password'];

$query = "SELECT * FROM user WHERE username = '$username'
          AND password = '$password'";

echo "$query<br>";

$hasil = mysql_query($query);
$jmldata = mysql_num_rows($hasil);

if ($jmldata != 0) echo "<h1>Login SUKSES</h1>";
else echo "<h1>Login GAGAL</h1>";

?>
```

Ide dari proses login di atas adalah mencari namauser dan password dalam tabel user berdasarkan username dan password yang dimasukkan melalui form. Perintah `mysql_num_rows()` akan menghasilkan jumlah record hasil pencarian. Apabila ditemukan namauser dan password yang sesuai maka `mysql_num_rows()` akan menghasilkan jumlah baris record yang tidak sama dengan 0. Sedangkan apabila tidak ditemukan, jumlah baris record adalah 0. Selanjutnya jumlah baris record yang muncul tersebut dicek. Login akan berhasil apabila ada record yang ditemukan atau jumlah baris recordnya tidak sama dengan 0, dan akan gagal jika jumlah baris recordnya 0.

Sekarang, perhatikan apa yang terjadi apabila dalam form dimasukkan input sebarang nama user dan password berbentuk `' OR 1 OR '1' = '1`. Pastilah login akan sukses karena perintah SQL yang dijalankan adalah

```
SELECT * FROM user WHERE username = 'xxx'
         AND password = '' OR 1 OR '1' = '1'
```

yang akan menghasilkan TRUE

Cara Pencegahan SQL Injection

Beruntunglah … sekarang PHP sudah dilengkapi dengan security (PHP rilis terbaru) dengan mengubah karakter single quote (‘) menjadi (\’). Dengan kata lain karakter single quote yang diinputkan melalui form secara otomatis akan diubah menjadi backslash-single quote (\’).

Sehingga untuk kasus login di atas perintah SQL yang dijalankan adalah

```
SELECT * FROM user WHERE username = 'xxx'
         AND password = '\' OR 1 OR \'1\' = \'1'
```

sehingga akan menghasilkan FALSE karena password yang dicari adalah `\' OR 1 OR \'1\' = \'1`. Dengan demikian login akan gagal.

Catatan:
Dalam MySQL, tanda single quote digunakan sebagai pengapit data/value yang berupa string. Sedangkan backslash-single quote merupakan tanda yang digunakan untuk menyatakan karakter single quote.

Apabila PHP Anda tidak mensupport metode tersebut (biasanya PHP rilis lama), cara lain adalah menggunakan perintah `mysql_real_escape_string()`. Peran perintah ini sama dengan metode di atas yaitu mengubah single quote menjadi backslash-single quote.
Script berikut ini adalah modifikasi dari submit.php pada kasus login di atas yang ditambahkan perintah `mysql_real_escape_string()`.

```
<?

mysql_connect("localhost","root","root");
mysql_select_db("test");

$username = mysql_real_escape_string($_POST['username']);
$password = mysql_real_escape_string($_POST['password']);

$query = "SELECT * FROM user WHERE username = '$username'
          AND password = '$password'";

echo "$query<br>";

$hasil = mysql_query($query);
$jmldata = mysql_num_rows($hasil);

if ($jmldata != 0) echo "<h1>Login SUKSES</h1>";
else echo "<h1>Login GAGAL</h1>";

?>
```