SQL Serverでどーんといってみよう!

必ず役立つ現場のノウハウ

百田 昌馬
HYAKUTA, Shoma
Supported by 松本 美穂
http://www.sqlquality.com/

第6回 SQLインジェクション対策（ADO後編）

Level

1 2 **3** 4 5

Technology Tools

- ☑ Visual Basic
- ☐ Visual C#
- ☐ Visual C++
- ☑ SQL Server
- ☐ Oracle
- ☐ Access
- ☑ ASP.NET
- ☑ Other:
  - MSDE
  - Visual Studio 6.0

Samples

はじめに

前回は、ASP／ADOにおけるSQLインジェクション対策として、Parameterオブジェクトを使ったSQLのパラメータ化を説明した。今回は、TOPやORDER BY句などパラメータ化を利用できない場合や、Windowsアプリケーションにおける ADODC（ADOデータコントロール）を利用した場合、LIKE演算子を使った場合のSQLインジェクション対策について説明する。

TOP、ORDER BYなどパラメータ化を利用できない場合

Parameterオブジェクトによる SQLのパラメータ化は、WHERE句の条件式における定数値にしか利用できない。たとえば、前回の説明で利用した次のようなケースでは、条件式における定数値なのでパラメータ化を利用できる。

ケース1
```
SELECT * FROM 会員
  WHERE emailアドレス=?
    AND パスワード=?
```

ケース2
```
SELECT * FROM Products
  WHERE CategoryID=?
```

しかし、次のようにTOP句やORDER BY句、テーブル名、列名などに対しては、パラメータ化は利用できないのである。

例1 SELECT TOP ? * FROM t1
例2 SELECT * FROM t1 ORDER BY ?
例3 SELECT * FROM ?
例4 SELECT ?, ? FROM t1

したがって、図1のようにアプリケーション上で並べ替えや取得件数を絞り込ませるようなケースでは、パラメータ化を利用できない。

具体的には、リスト1のようなコードを記述できない。これを実行すると「構文エラーまたはアクセス違反です」エラーになってしまう。

このようにパラメータ化が利用できない場合は、次のようにSQLを文字列として組み立てなければならない。

```
"SELECT TOP " & topN & _
    " ProductID, ProductName, UnitPrice" _
 & " FROM Products ORDER BY " & sort
```

しかしこのままでは、前回説明したように“半角スペース”や“;”“--”を使って任意のSQLコマンドを埋め込み、SQLインジェクション攻撃ができてしまう（TOPのところは“GETDATE() 任意のSQL--”のようにGETDATEなど適当な関数を埋め込めば、構文エラーにならずに任意のSQLを実行できる）。

✷TOP句での数値チェック

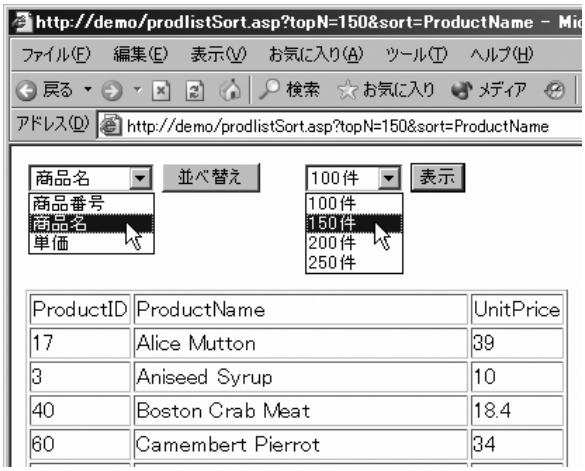
TOP句でのSQLインジェクション対策は、入力値が数値かどうかをチェックするだけでよい。また、入力値の長さチェックも念のため行なっておくとよい。たとえば、表示件数を500件まで（TOP 500が上限）としている場合は、文字列の長さが3桁以下であることをチェックする。これらは、Len関数とIsNumeric関数を使って、次のように記述できる。

```
If Len(topN) <= 3 And IsNumeric(topN) Then
  ' 3桁以下の数値（OK）
Else
  ' 不正な文字あり
End If
```

また、次のように正規表現（RegExpオブジェクト）を使ってもよい。

```
Set r = Server.CreateObject("VBScript.Regexp")
r.Pattern = "^[0-9]{1,3}$"
If r.Test(topN) Then
  ' 3桁以下の数値（OK）
Else
  ' 不正な文字あり
End If
```

図1：並べ替えや取得件数の制限



リスト1：TOPやORDER BYをパラメータ化？

```
topN = Request.QueryString("topN")
sort = Request.QueryString("sort")

Set cmd = Server.CreateObject("ADODB.Command")
cmd.ActiveConnection = cn

cmd.CommandText = "SELECT TOP ? " _
                & " ProductID, ProductName, UnitPrice" _
                & " FROM Products ORDER BY ?"

cmd.Parameters(0).Value = topN
cmd.Parameters(1).Value = sort

rs.Open cmd
```

✷ORDER BY句での文字列チェック

ORDER BY句に指定するのは、テーブルの列名になる。したがって、ORDER BY句でのSQLインジェクション対策は、入力値が正しい列名かどうかをチェックすればよい。たとえば、図1のように商品番号（ProductID）、商品名（ProductName）、単価（UnitPrice）のいずれかで並べ替えをさせる場合は、リスト2のようにチェックする。

このように、パラメータ化が利用できない場合は、数値チェックや文字列チェックを明示的に行なわなければならない。また、ORDER BY句のケースでは、並べ替えに指定してもよい列数が多い場合には、列名を列挙するのが大変になるが、安全で確実な方法である。たとえば、“半角スペース”や“;”“--”“UNION”“xp_cmdshell”