عنوان: **SQL Injection چیست؟**
نویسنده: م منفرد
تاریخ: ۱۳۹۲/۱۰/۱۰ ۰:۵
آدرس: www.dotnettips.info
گروه‌ها: امنیت, بانک, Security

برای ایجاد امنیت در نرم افزار، باید ابتدا مشکلات رایج را بدانیم. یکی از رایجترین نقائص امنیتی نرم افزارها SQL Injection می‌باشد.
SQL Injection در لغت به معنی تزریق کد SQL می‌باشد. در اصلاح یعنی تزریق دستوراتی به کد SQL تولیدی یک نرم افزار به نحوی که به جای عمل مورد انتظار برنامه نویس آن، کاری را که ما می‌خواهیم انجام دهد. مثلا به جای اینکه هنگام ورود به برنامه وقتی کاربر مشخصات کاربری خود را وارد می‌کند، مشخصات کاربری را به نحوی وارد کنیم که بتوانیم بعنوان مدیر سامانه و یا یک کاربر معمولی بدون داشتن کلمه عبور وارد سیستم شویم.
البته همیشه از این نوع حمله برای ورود به سیستم استفاده نمی‌شود. یعنی ممکن است هکر به عنوان یک کاربر عادی وارد سیستم شود ولی با به کاربردن دستورات خاص SQL در بخش‌های مختلف، بتواند اطلاعاتی را حذف نماید.
خوب حالا این کار چگونه انجام می‌شود؟
فرض کنید برنامه نویسی کد چک نام کاربری را اینگونه نوشته باشد:

```
SqlCommand cmd=new SqlCommand ("select count(*) from login where user='"+userName+"' and
pass='"+password+"'",con);
```

فکر نکنید خوب این نوع کد نویسی مربوط به زمان تیرکمون شاه است! همین امروز در نظارت از یک پروژه به این نکته برخورد کردم! دلیل نوشتن این مقاله هم همین کد بود.
خوب حالا مگر کد بالا چه مشکلی دارد؟ ;) اگر کاربر در نامه کاربری و کلمه عبور مقادیر معمولی وارد کند (مانند admin, salam123) کد sql تولید شده به شکل زیر خواهد بود:

```
select count(*) from login where user='admin' and pass='salam123'
```

خوب حالا اگر کاربر کمی با ورودی‌ها بازی کند. به عنوان مثال فرض کنید به جای کلمه عبور تایپ کند

```
' or 1=1 --
```

نتیجه حاصله خواهد بود:

```
select count(*) from login where user='admin' and pass='' or 1=1 --'
```

با وارد کردن این دستور کاربر بدون داشتن کلمه عبور خواهد توانست وارد سیستم شود. موردی که توضیح دادم پایه مسئله بود. ما قصد آموزش هک نداریم ولی داشتن اطلاعات پایه لازم است. ممکن است فردی بگوید خوب ما قبل از تولید همچین کدی ' را از رشته کلمه عبور حذف می‌کنیم. خیلی خوب ولی اگر هکر از معادل unicode آن استفاده کرد چه؟ اگر و اگر و اگر...
راه حل‌های متعددی برای این موضوع پیشنهاد شده است. ولی ساده‌ترین و کارآمدترین راه، استفاده از پارامترها می‌باشد که علاوه بر حذف این خطر باعث ایجاد و ذخیره query plan در sql server می‌شود و اجرای این query را در آینده تسریع می‌کند.
بنابراین می‌توان کد فوق را به صورت زیر بازنویسی کرد:

```
SqlCommand cmd=new SqlCommand ("select count(*) from login where user=@u and pass=@p",con);
cmd.Parameters.Add("@u", SqlDbType.Varchar, 10).Value=TextLogin.Text.Trim();
cmd.Parameters.Add("@p", SqlDbType.Varchar,10).Value=TextPwd.Text.Trim();
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۰ ۰:۹

استفاده از ORMها هم می‌تونه مفید باشه.

نویسنده: م منفرد
تاریخ: ۱۳۹۲/۱۰/۱۰ ۰:۱۶

استفاده از ORM هم خوب است ولی استفاده از ORM یا Stored Procedure به صورت مطقن مشکل را حل نمی‌کند. ممکن است در یک Stored Procedure کد sql یه صورت dynamic توید و با sp_executesql اجرا شود که همچنان مشکل پا برجا خواهد بود.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۰ ۰:۱۹

البته کدی که ابزارهای ORM به صورت خودکار از عبارات LINQ تولید می‌کنند دارای dynamic sql نیست؛ مگر اینکه شخصی خودش عمدا این‌کار رو دستی انجام بده و بعد از ORM برای اجرای این SP کمک بگیره.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۰ ۰:۳۱

اگر کسی هنوز می‌خواد SQL بنویسه، استفاده از Micro ORMها (^ و ^) بهتر از این کتابخانه‌های SQL Helper دستی هست (^ و ^).

نویسنده: ساسان عزیزی
تاریخ: ۱۳۹۲/۱۰/۱۱ ۱۰:۲۸

یعنی در صورت استفاده از linq هم باز این مشکل پا برجاست؟!

نویسنده: مصطفی
تاریخ: ۱۳۹۲/۱۰/۱۱ ۱۰:۵۶

وقتی شما از linq استفاده کنید نفوذ از طریق sql injection به شدت کاهش پیدا می‌کنه
این لینک رو که توسط آقای نصیری توضیح داده شده مطالعه کنید
امنیت در LINQ to SQL

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۱ ۱۱:۰

با LINQ نه ولی اگر دستی SP بنویسید ممکنه این SP شما نا امن باشد. یک مثال در اینجا:

خطر SQL injection هنگام استفاده از ORM و SP