ایجاد یک ارتباط امن با SQL Server در ASP.NET

نگهداري صريح رمزعبور و Hard Code نمودن آن در برنامه هاي كاربردي و وب سايتها قابل قبول نمي باشد. SQL Server داري امكاني با عنوان Connection Trusted مي باشد كه از طريق آن اعتبار سنجي لازم براي دسترسي به منابع Server SQL از روي نام Login انجام مي شود. در اين حالت هيچ رمز عبوري به SQL Server فرستاده نمي شود و تنها نام Login و يک Authentication Token به آن ارسال مي گردد. در صورتيكه شما خواسته باشيد از اين امكان در ASP.NET استفاده نماييد در پاره اي از موارد تا حدودي با مشكل مواجه خواهيد شد. زيرا كاربري كه ASP.NET تحت حقوق دسترسي آن اجرا مي شود نقش تعيين كننده اي را ايفا مي كند. در حالت پيش فرض ASP.NET با استفاده از كاربر ASPNET در ماشين محلي اجرا مي گردد. در صورتيكه ASP.NET و Server SQL هر دو برروي يک ماشين اجرا شوند كار بسيار ساده مي باشد. در اين حالت بايد به كاربر ASPNET در SQL Server حقوق دسترسي و مجوزهاي لازمه اعطا گرديده و در نهايت در رشته Connection لازم است عبارت Integrated Security=SSPI

يا Trusted_Connection=true بر حسب سبک رشته Connection در نظر گرفته شود.

مشكل زماني رخ مي دهد كه ASP.NET و SQL Server برروي ماشينهاي جداگانه اي اجرا گردند كه در واقع در اكثر موارد نيز چنين مي باشد. زيرا كاربر ASPNET در SQL Server قابل دسترسي نمي باشد.

پنج راه اصلي براي غلبه بر اين مشكل موجود مي باشد.

- استفاده از IIS 6 در حالت Native Application
- انطباق كاربر ASP.NET برروي IIS و SQL Server و مشخص كردن رمز عبور.
- استفاده از جعل هويت (Impersonation) بمنظور تغيير كاربر ASP.NET.
- كد كردن (Encrypt) رشته Connection و قرار دادن آن در رجيستري و فراموش كردن ارتباط امن!
- تغيير كاربر ASP.NET به يک Domain User.

اجراي هر نوع سرويس وب به عنوان Domain User بسيار خطرناک مي باشد. زيرا در اين صورت هكرها با استفاده از حقوق دسترسي كاربر دامنه قابليت دسترسي به تمامي منابع داخل و خارج سرويس دهنده وب را دارا مي باشند.

كد كردن داده ها و دسترسي به آنها بطور كامل در مقاله شماره Knowledge Base 329290 Microsoft و

مقاله Applicartion Building Secure ASP.NET در MSDN بررسي شده است.

هر دو حالت انطباق کاربر و جعل هویت نیازمند این مي باشد که شما داري حسابهاي Mirror در IIS و SQL Server باشید. (در صورتیکه شما در یک محیط Active Directory و Domain قرار نداشته باشید)

جعل هویت

جعل هویت به شما این امکان را مي دهد که به ASP.NET بگویید تحت عنوان یک کاربر بخصوص اجرا شود. برروي هر دو ماشین مورد نظر کاربري همنام و با رمز عبور مطمئن ایجاد نمایید. برروي سرویس دهنده IIS کاربر ایجاد شده باید داراي توانایي اجرا به عنوان کاربر ASP.NET را دارا باشد (براي اطلاعات بیشتر مي توانید به MSDN مراجعه نمایید). برروي سرویس دهنده SQL Server نیز کاربر مورد نظر باید قابلیت دسترسي به منابع مورد نظر از جمله بانک اطلاعاتي، جداول، دیدگاهها، روالهاي ذخیره شده و ... را داشته باشد.

اکنون شما مي بایست ASP.NET را بمنظور اجرا تحت عنوان این کاربر پیکره بندي نمایید. دو راه حل بمنظور انجام این عمل در پیش روي شما مي باشد. راه اول قرار دادن رمز عبور در فایل Web.Config (که راه نامطلوبي است) و دومین راه استفاده از ابزار IIS Administration همراه تنظیماتي در فایل Web.Config.

براي Hard Code کردن رمز عبور و نام کاربر فایل Web.Config را ویرایش کرده و حالت عنصر identity را بصورت زیر به آن اضافه نمایید.

```
<system.web>
  ...
  <identity impersonate="true"
    userName="yourNewUsername"
    password="yourStrongPassword" />
  ...
</system.web>
```

اين حالت در واقع تمامي تلاش شما براي قرار ندادن رمز عبور در كد را بي نتيجه خواهد كرد.

در صورتيكه شما نياز داشته باشيد كه رمز عبور را در كد قرار ندهد (كه بايد همين طور باشد) مي توانيد از قرار دادن نام كاربر و رمز عبور در فايل Web.Config صرفه نظر كرده و تنظيمات مربوطه را در IIS انجام دهيد.

اما نياز است شما عنصر idenity را بصورت زير در فايل Web.Config قرار دهيد.

```
<system.web>
  ...
  <identity impersonate="true" />
  ...
</system.web>
```

اكنون ابزار IIS Administration را باز كرده و روي شاخه اي كه برنامه شما در آن قرار دارد كليک راست نماييد. پنجره خصوصيات را باز كرده و در برگه Directory Security دكمه Edit قسمت Autentication and access control را كليک نماييد. قسمت Enable anonymous access را تيک زده و نام كاربر و رمز عبور آنرا مشخص نماييد.

ويرايش محتوي پيش فرض

در صورتيكه شما با IIS 5 كار مي كنيد و انجام اين عمل ممكن است اثرات جانبي داشته باشد و يا شما داري چندين برنامه مي باشيد كه تعيين كاربر براي تک تک آنها وقت گير مي باشد، در اين حالت مي توانيد مستقيما محتوي كاربر ASP.NET را تغيير دهيد. همانطور كه قبلا ذكر شد ASP.NET تحت كاربر ASPNET اجرا مي گردد و اين كاربر مجوزهاي لازم را براي اجراي صفحات ASP.NET را دارا مي باشد. اين كاربر در زمان نصب Framework بطور خودكار ايجاد مي گردد و رمز عبور آن براي ما شناخته شده نيست. تنها عمل ممكن براي ما Rest كردن رمز عبور آن (فراموش نشود كاربر ASPNET را در SQL Server با همان رمز عبور

ثبت نماييد) و سپس ويرايش فايل machine.config و اطلاع به Framework بمنظور استفاده از اين رمز عبور جديد. متاسفانه شما نياز است كه رمز عبور را بصورت صريح در اين فايل ذكر نماييد. اين فايل در مسير زير قرار دارد.

```
c:\windows\microsoft.net\framework\versionNumber\config
```

بمنظور پيكره بندي مجدد و تعيين رمز عبور جديد عنصر processModel را در فايل فوق پيدا كرده و آنرا بصورت زير با جايگزيني هاي لازم تغيير دهيد.

```
<processmodel
  ...
  userName="ASPNET" password="ASPNETpassword"
  ...
/>
```

پس از انجام اين عمل IIS را راه اندازي مجدد نماييد.

IIS 6 در حالت محلي

آخرين راه و در واقع بهينه ترين راه استفاده از IIS 6 و اجراي برنامه هاي ASP.NET در حالت محلي مي باشد. IIS 6 به شما امكان ايجاد Pool Application را مي دهيد. يک Pool، كنترل كارايي، چرخه پروسه ها و مهمتر از همه امنيت لازم براي ايجاد ارتباط امن را فراهم مي كند. در اين حالت شما همانند حالت قبل نيازمند ايجاد يک كاربر برروي IIS و SQL Server مي باشيد. براي ايجاد يک Pool ابزار IIS Administration را

فعال کرده برروي Application Pool کليک راست کرده و گزينه New Application Pool را انتخاب نماييد. نامي براي Pool خود انتخاب کرده و دکمه OK را کليک نماييد. سپس برروي نام Pool ايجاد شده راست کليک کرده و گزينه Properties را انتخاب نماييد. با انتخاب برگه Identity نام کاربر و رمز عبور آنرا مشخص نماييد. در ادامه شما نيازمند تغيير تنظيمات برنامه خود براي استفاده از Pool ايجاد شده مي باشد. بدين منظور برروي نام برنامه خود در IIS راست کليک کرده و گزينه Properties را انتخاب نماييد. در پنجره Properties برگه Home Directory را انتخاب کرده وسپس Application Pool آنرا مطابق Pool ايجاد شده تغيير دهيد.

تاثيرات جانبي

در استفاده از ارتباط امن با اثراتي جانبي مواجه مي باشيم. اگر هر ارتباط تحت کاربراني مجزا باز شده باشند در اين حالت ارتباطات ايجاد شده بين کاربران به اشتراک گذاشته نخواهند شد. اين عمل سرباري برروي IIS و SQL Server ايجاد خواهد کرد. ارتباط امن در مقايسه با حالت ديگر قدرت پردازشي بيشتري را طلب مي کند زيرا درخواست ها خارج از SQL Server و تحت نظارت NT انجام مي پذيرد. در صورتيکه شما از داخل يک دامنه اين اعمال را انجام دهيد سربار بيشتري برروي کنترل کننده هاي دامنه و SQL Server اعمال خواهد شد.

خلاصه

در واقع يک راه واحد براي ايجاد ارتباط امن با SQL Server در ASP.NET وجود ندارد. Microsoft راههاي مختلفي براي رسيدن به اين منظور فراهم آورده است و انتخاب بهترين راه براي برنامه شما بر عهده خود شما مي باشد.

برگرفته از سايت : http://idunno.org/dotNet/trustedConnections.aspx