امنیت برنامه های وب (بخش دوم)

در این مقاله به بررسی Windows Authentication خواهیم پرداخت .

همانگونه که در بخش اول این مقاله اشاره گردید ، برنامه های وب ASP.NET از سه روش عمده به منظور تائید کاربران استفاده می نمایند :

Authentication Windows

Forms Authentication

Passport Authentication

در Windows Authentication ، برنامه های وب مسئولیتی را در ارتباط با تائید کاربران برعهده نگرفته و این وظیفه تماما" به سیستم عامل ویندوز ، واگذار می گردد. فرآیند تائید کاربران در روش فوق، بصورت زیر است :

کاربر درخواستی مبنی بر دریافت یک صفحه وب ایمن را از برنامه وب ، می نماید .

پس از دریافت درخواست توسط سرویس دهنده وب ، IIS عملیات بررسی صلاحیت کاربر را انجام خواهد داد . در این راستا ، اطلاعات ارائه شده توسط کاربر در زمان logon (نام و رمز عبور) ، با اطلاعات موجود بر روی سرویس دهنده وب و یا Domain ، مقایسه می گردد .

در صورتیکه پس از بررسی مدارک ارائه شده توسط کاربر (نام و رمز عبور) ، وی به عنوان کاربر غیر مجاز تشخیص داده شود ، درخواست وی نادیده گرفته خواهد شد .

کامپیوتر سرویس گیرنده ، یک جعبه محاوره ای Logon را تولید و از کاربر درخواست درج اطلاعات مورد نیاز (نام و رمز عبور) ، می گردد . پس از درج اطلاعات درخواستی توسط کاربر و ارسال آنان برای سرویس دهنده ، مجددا" IIS بررسی لازم در خصوص صحت آنان را انجام خواهد داد . در صورتیکه صحت اطلاعات ارسالی کاربر (نام و رمز عبور) تائید گردد ، IIS درخواست اولیه کاربر را به سمت برنامه وب هدایت می نماید .

در آخرین مرحله و پس از بررسی و تائید صلاحیت کاربر ، صفحه وب درخواستی برای کاربر ارسال می گردد .

مهمترین مزیت روش Authentication Windows ، استفاده مشترک از یک مدل امنیتی به منظور دستیابی به منابع موجود در شبکه و برنامه های وب است . پس از تعریف و اعطای مجوزهای لازم به کاربر ، امکان دستیابی وی به منابع موجود در شبکه و برنامه های وب بر اساس یک سیستم امنیتی مشابه و یکسان ، فراهم می گردد .

در زمان ایجاد یک پروژه جدید برنامه وب توسط ویژوال استودیو دات نت ، از روش Windows Authentication بصورت پیش فرض به منظور تائید کاربران استفاده می گردد .پس از ایجاد یک پروژه جدید برنامه وب در ویژوال استودیو دات نت ، فایل Web.Config بصورت اتوماتیک ایجاد می گردد . (یک فایل XML که اطلاعات متفاوتی را در ارتباط با پیکربندی رنامه وب در خود ذخیره می نماید) . محتوی پیش فرض این فایل بصورت زیر است (صرفا" بخشی که با موضوع این مقاله ارتباط دارد ، منعکس می گردد) :

Web.Config default setting

```
<authentication mode="Windows" />
<authorization>
<allow users="*" /> <!-- تمامی کاربران -->
</authorization>
```

در بخش مربوط به عنصر authentication ، سیاست تائید کاربران برنامه های وب مخشص می گردد . برای

مشخص نمودن سیاست فوق از خصلت mode مربوط به عنصر authentication ، استفاده شده که می تواند یکی از مقادیر : Passport, Forms , Windows و یا None را دارا باشد . در بخش authorization ، سیاست های مربوط به کاربران مجاز برنامه وب مشخص می گردد . در این رابطه می توان ، امکان دستیابی و یا عدم دستیابی به برنامه های وب را با مشخص نمودن کاربران و یا با توجه به وظایف آنان ، فراهم نمود. (استفاده از کاراکتر " * " ، به معنی همه کاربران بوده و کاراکتر "؟" به منزله کاربران ناشناس و غیرمجاز است) . برای آشنائی با عملکرد روش Windows Authentication ، مراحل زیر را دنبال می نمائیم :

بخش authorization در فایل Web.Config را بصورت زیر تغییر می نمائیم :

Authorization element

```
<authorization>
<deny users="?" />
</authorization>
```

تگ های زیر را که یک جدول HTML را تعریف می نمایند ، در فرم وب شروع برنامه وب ، قرار می دهیم :

HTML Table in Startup web form

```
<TABLE id="tblUser">
<tr>
<TD><STRONG>آیا کاربر تائید شده ؟ است</STRONG></TD>
<TD><Span runat="server" id="spnAuthenticated"></Span></TD>
</tr>
<tr>
<TD><STRONG>نام کاربر </STRONG></TD>
<TD><Span runat="server" id="spnUserName"></Span></TD>
</tr>
<tr>
<TD><STRONG>نوع کاربر تائید</STRONG></TD>
<TD><Span runat="server" id="spnAuthenticationtype"></Span></TD>
```

```
</tr>
</TABLE>
```

به حالت Design view سوئیچ نموده و کد زیر را در فایل Code Behind فرم وب شروع برنامه ، قرار می دهیم :

```
Web form's code-behind file

Private Sub Page_Load( ByVal sender As System.Object,ByVal e As
System.EventArgs ) Handles Mybase.Load
spnAuthenticated.InnerText = User.Identity.IsAuthenticated
spnUserName .InnerText = User.Identity.Name
spnAuthenticationType.InnerText = User.Identity.AuthenticationType
End Sub
```

پس از اجرای پروژه بصورت محلی ، ASP.NET تائید کاربر را بر اساس نام و رمز عبوری که برای ورود به ویندوز استفاده شده است ، انجام خواهد د اد .

پس از اجرای پروژه از راه دور (مثلا" دستیابی از طریق اینترنت) ، ASP.NET یک جعبه محاوره ای رادر مرورگر نمایش داده تا از طریق آن نام و رمز عبور کاربر دریافت گردد .

در صورتیکه نام و رمز عبور درج شده توسط کاربر با تعاریف انجام شده در Domain شبکه ، مطابقت نماید ، ASP.NET کاربر را تائید و مجوز لازم به منظور استفاده از برنامه وب صادر خواهد شد . در این رابطه ASP.NET ، یک authorization certificate را به شکل یک کوکی صادر که در حین Session کاربر ، نگهداری و از آن استفاده می گردد. Session کاربر، پس از اتمام زمان Time out و یا بستن مرورگر ، خاتمه می یابد . برنامه وب اجرای خود را متناسب با مجوزهای تعریف شده در ارتباط با Account آغاز می نماید .
روش Windows integrated authentication در یک شبکه مبتنی بر Domain بهتر کار خواهد کرد .
شبکه هائی که از Workgroup استفاده می نمایند (در مقابل استفاده از Domain) دارای محدودیت های

خاص خود به منظور استفاده از ویژگی های امنیتی ، می باشند. شبکه های مبتنی بر Domain ، از یک کنترل کننده Domain به منظور تائید و معتبرسازی کاربران شبکه ، استفاده می نماید .

با استفاده از امکانات ارائه شده در فایل Web.Config می توان یک لایه امنیتی مضاعف را ایجاد نمود . دراین راستا ، می توان تنظیمات لازم به منظور دستیابی و یا عدم دستیابی کاربران و یا گروه های خاصی از کاربران را نیز انجام داد .

اعمال محدودیت برای کاربران خاص (دستیابی و یا عدم دستیابی)

در مواردیکه از روش Windows integrated authentication استفاده می گردد ، ASP.NET ، لیست تائید موجود در فایل Web.Config را به منظور آگاهی از صلاحیت کاربران شبکه برای استفاده ازبرنامه وب ، بررسی می نماید. کاراکترهای "*" و "؟" دارای معانی خاصی در لیست تائید می باشند : کاراکتر "* " ، نشاندهنده تمامی کاربران و کاراکتر "؟"، نشاندهنده کاربران غیر مجاز(ناشناس) می باشد . مثلا" لیست تائید زیر در Web.Config ، امکان دستیابی تمامی کاربران ناشناس به برنامه وب را حذف و می بایست تمامی کاربران به منظور استفاده از برنامه وب ،تائید گردند .

Authorization element

```
<authorization>
<deny users="?" />
</authorization>
```

به منظور اعمال محدودیت در دستیابی کاربرانی خاص ،می توان از عنصر <allow> استفاده و اسامی تمامی کاربران مجاز را با صراحت مشخص نمود (اسامی توسط ویرگول از یکدیگر تفکیک می گردند) . پس از معرفی کاربران مجاز با استفاده از عنصر <allow> ، می بایست با بکارگیری عنصر <deny> ، امکان دستیابی به برنامه توسط کاربران غیر مجاز، سلب می گردد .

Authorization element

```
<authorization>
<allow users="Ali Reaz , Reza Ali " />
```

```
<deny users="*" />
</authorization>
```

لیست مجاز فوق ، امکان دستیابی دو کاربر که اسامی آنان با صراحت مشخص شده است را به برنامه وب خواهد داد. سایر کاربران ، امکان دستیابی به برنامه وب را دارا نخواهند بود (نقش عنصر deny در مثال فوق) علاوه بر لیست مجاز فوق که اسامی دو کاربر را مشخص و آنان را برای استفاده از برنامه وب مجاز می نماید ، دو کاربر فوق ، می بایست دارای Account لازم در Domain شبکه نیز باشند .

تائید کاربران بر اساس نوع وظیفه

برای تائید کاربران به منظور استفاده از یک برنامه می توان ، مجوزهای لازم را بر اساس وظیفه آنان در سازمان ، صادر و امکان دستیابی و یا عدم دستیابی را برای آنان فراهم نمود. در ویندوز NT و XP ، وظایف به اسامی مپ شده تا از این طریق امکان شناسائی گروه های کاربران ، فراهم گردد. ویندوز، چندین گروه را بصورت اتوماتیک از قبل ایجاد می نماید : Users, Administrators و Guests . در این رابطه می توان از عنصر <roles> در لیست استفاده کنندگان مجاز برنامه وب در فایل Web.Config استفاده و امکان دستیابی به یک برنامه را با توجه به وظایف کاربر ، فراهم نمود. مثلا" لیست زیر، امکان دستیابی به برنامه وب را صرفا" برای کاربرانی که به عنوان Administrator به شبکه وارد می شوند ، فراهم می نماید.

Authorization element

```
<authorization>
<allow roles ="Administrators" />
<deny users="*" />
</authorization>
```

پس از تائید کاربر و صدور مجوز لازم به منظور استفاده از برنامه وب ، می توان با استفاده از خصلت Identity مربوط به شی User ، هویت کاربر (نام و نوع وظیفه) را از طریق برنامه شناسائی نمود.خصلت فوق، یک شی را که شامل اطلاعات مربوط به نام و وظیفه کاربراست را برمی گرداند .

Web form's code-behind file

```
Private Sub Page_Load( ByVal sender As System.Object,ByVal e As
System.EventArgs ) Handles Mybase.Load
spnAuthenticated.InnerText = User.Identity.IsAuthenticated
spnUserName .InnerText = User.Identity.Name
spnAuthenticationType.InnerText = User.Identity.AuthenticationType
End Sub
```

به منظور آگاهی و انجام عملیات لازم با توجه به نوع وظیفه کاربر که از برنامه وب استفاده می نماید ، می توان از متد IsInRole شی User ، استفاده نمود .

IsInRole method

```
If User.IsInRole("Administrators") Then
'انجام عملیات دلخواه
End If
```

استفاده از تنظیمات IIS به همراه Authentication Windows

تنظیمات Authorization در فایل Web.Config با تنظیمات انجام شده در IIS با یکدیگر Overlap می شوند . در صورتیکه Authorization هم در فایل Web.Config و هم توسط IIS تنظیم شده باشد ، در ابتدا تنظیمات IIS بررسی و در ادامه تنظیمات موجود در فایل Web.Config ، مورد توجه قرار خواهند گرفت. به منظور مشاهده تنظیمات authorization در IIS مراحل زیر را دنبال می نمائیم :

در IIS بر روی فولدر برنامه وب کلیک سمت راست نموده و در ادامه گزینه Properties را انتخاب می نمائیم . برنامه IIS در ادامه جعبه محاوره ای Properties مربوط به فولدر را نمایش خواهد داد .

بر روی Directory Security Tab کلیک و در ادامه دکمه Edit را در گروه Anonymous Access And Authentication Control کلیک می نمائیم . IIS ، جعبه محاوره ای Authentication Methods

را نمایش خواهد داد .

اولین گروه از تنظیمات در جعبه محاوره ای ، کنترل دستیابی Anonymous را انجام می دهد (همه کاربران). غیر فعال نمودن گزینه فوق ، معادل <deny User = "?"> در فایل Web.config است.

Check Box های موجود در قسمت دوم جعبه محاوره ای ، مجاز بودن برنامه به منظور استفاده از Basic و یا Digest Authentication را علاوه بر Windows Authentication ، مشخص می نماید. روش های فوق ، ایمنی بمراتب کمتری را نسبت به Windows Integrated ارائه می نمایند .می توان چندین روش authentication را در IIS فعال نمود . در صورتیکه چندین روش فعال شده باشد ، می توان با استفاده از متد AuthenticationType مربوط به شی Identity ، از روش استفاده شده به منظور تائید کاربر ، آگاهی یافت .

AuthenticationType method

```
Response.Write(User.Identity.AuthenticationType)
```