امنیت برنامه های وب (بخش سوم)

در این مقاله به بررسی Authentication Forms 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 :

Windows Authentication

Forms Authentication

Authentication Passport

در Forms Authentication ، برنامه IIS مسئولیتی را در ارتباط با تائید کاربران برعهده نگرفته و تنظیمات امنیتی IIS در رابطه با برنامه وب ، دستیابی Anonymous می باشد . فرآیند تائید کاربران در روش فوق، بصورت زیر است :

زمانیکه سرویس گیرنده درخواست یک صفحه ایمن را می نماید ، IIS کاربر را به عنوان Anonymous، تائید و در ادامه درخواست وی را برای ASP.NET ارسال می نماید .

ASP.NET ، بررسی لازم در خصوص وجود یک کوکی خاص بر روی کامپیوتر سرویس گیرنده را انجام خواهد داد .

در صورتیکه کوکی ، موجود نبوده و یا غیرمعتبر باشد ، ASP.NET درخواست کاربر را نادیده گرفته و برای وی یک صفحه Logon را ارسال می نماید (مثلا" Login.aspx).

کاربر اطلاعات لازم (نام و رمز عبور) را در صفحه Logon.aspx (به عنوان نمونه) درج و در ادامه دکمه Submit موجود بر روی فرم را به منظور ارسال اطلاعات برای سرویس دهنده ، فعال می نماید.

IIS ، مجددا" کاربر را به عنوان Anonymous، تائید و درخواست وی را برای ASP.NET ارسال می نماید .

ASP.NET ، تائید کاربر را بر اساس اطلاعات ارسالی (نام و رمز عبور) انجام و یک کوکی را ایجاد می نماید .

در نهایت ، صفحه وب ایمن درخواست شده به همراه کوکی جدید برای سرویس گیرنده ارسال می گردد.

مادامیکه کوکی معتبر باشد ، کاربر قادر به درخواست و مشاهده سایر صفحات وب می باشد.

فرآیند فوق را می توان به دو حالت متفاوت تعمیم و مورد توجه قرار داد :

حالت اول : درخواست یک صفحه ایمن از سرویس دهنده ، توسط یک کاربر غیرمجاز و تائید نشده

مرحله اول : پس از درخواست یک سرویس گیرنده برای دستیابی به یک صفحه ایمن ، درخواست ارسالی وی در ابتدا توسط IIS بررسی و با توجه به اینکه تنظیمات IIS بصورت Anonymous پیکربندی شده تا امکان استفاده از Forms Authentication فراهم گردد ، درخواست کاربر ، مستقیما" برای ماژول ASP.NET Forms Authentication ارسال می گردد .

مرحله دوم : ASP.NET ، بررسی لازم در خصوص وجود (داشتن) یک کوکی Authentication را انجام خواهد داد . با توجه به اینکه کاربر اولین مرتبه است که درخواست اطلاعاتی را نموده و دارای یک کوکی نمی باشد ، سرویس گیرنده به صفحه Logon ، هدایت می گردد .

مرحله سوم : کاربراطلاعات ضروری (نام و رمز عبور) خود را در صفحه Logon درج و پس ازارسال آنان ،فرآیند بررسی اطلاعات ارسالی آغاز می گردد. در یک برنامه بزرگ ، بررسی اطلاعات کاربر از طریق یک بانک اطلاعاتی شامل مشخصات کاربران انجام می شود .

مرحله چهارم : در صورتیکه اطلاعات ارسالی کاربر (نام و رمز عبور) ، پس از بررسی توسط برنامه وب ، معتبر شناخته نگردند ، مجوز دستیابی برای کاربر صادر نشده و امکان دستیابی وی سلب می گردد .

مرحله پنجم : در صورتیکه پس از بررسی اطلاعات ارسالی، اعتبار وصحت آنان تائید گردد ، یک کوکی تائید ایجاد و در ادامه به کاربر مجوز لازم به منظور دستیابی به صفحه ، اعطاء می گردد .(هدایت کاربر به صفحه درخواست اولیه) .

حالت دوم : درخواست یک صفحه ایمن از سرویس دهنده ، توسط یک کاربر مجاز و تائید شده

مرحله اول : پس از درخواست یک صفحه ایمن توسط سرویس گیرنده ، کوکی Authentication بهمراه درخواست وی برای سرویس دهنده ، ارسال می گردد.

مرحله دوم :درخواست ارسالی توسط سرویس گیرنده در ابتدا توسط IIS دریافت و با توجه به تنظیمات انجام شده (دسیتابی Anonymous) ، درخواست وی مستقیما" برای ASP.NET Forms Authentication ارسال می گردد .

مرحله سوم : ماژول ASP.NET Forms Authentication ، بررسی لازم در خصوص کوکی را انجام و در صورتیکه کوکی معتبر باشد ، سرویس گیرنده تائید و امکان دستیابی و مشاهده صفحه وب درخواستی برای وی ، فراهم می گردد .

در روش Forms Authentication ، بصورت اتوماتیک یک فرم وب طراحی شده به منظور اخذ اطلاعات مربوط به نام و رمز عبور کاربران ، نمایش داده می شود . کد مرتبط با فرم وب ، عملیات تائید و معتبرسازی کاربر را بر اساس لیست ذخیره شده در فایل Web.Config برنامه و یا از طریق یک بانک اطلاعاتی جداگانه ، انجام می دهد. مزیت مهم Forms Authentication ، عدم ضرورت عضویت کاربران در Domain شبکه به منظور دستیابی به برنامه وب ، می باشد .

فعال نمودن Forms Authentication

به منظور استفاده از روش فوق ، می بایست مراحل زیر را دنبال نمود :

مقداردهی Authentication mode در فایل Web.config به Forms

ایجاد یک فرم وب به منظور اخذ اطلاعات کاربران (Logon Page)

ایجاد یک فایل و یا بانک اطلاعاتی به منظور ذخیره نام و رمز عبور کاربران

نوشتن کد لازم به منظور افزودن کاربر جدید به فایل و یا بانک اطلاعاتی کاربران

نوشتن کد لازم به منظور تائید کاربران با استناد به فایل و یا بانک اطلاعاتی کاربران

Forms Authentication ، از کلاس های موجود در namespace با نام System.Web.Security استفاده می نماید . به منظور استفاده از کلاس های فوق، می بایست در ویژوال بیسک دات نت از عبارت Imports و در ویژوال سی شارپ از Using استفاده گردد (در ابتدای هر ماژول که عملیات تائید را انجام خواهد داد : System.Web.Security Imports) .

مقداردهی Authentication mode

نوع تائید کاربران در یک برنامه وب ، می بایست با استفاده از عنصر <authentication> در فایل Web.config مشخص گردد. به منظور تنظیم برنامه مورد نظر خود برای استفاده از Forms Authentication ، تغییرات زیر را در فایل Web.Config ، اعمال می نمائیم :

Web.Config setting for Forms Authentication

```
<authentication mode="Forms">
<forms loginUrl = Login.aspx" >
```

```
<credentials passwordFormat = "Clear" >
<user name = "Ali" Password ="110" />
<user name = "Kaveh" Password ="111" />
</credentials>
</forms>
</authentication>
```

کد فوق، یک نوع ساده از تائید کاربران به روش Forms را نشان می دهد . در این رابطه ، اغلب از تعاریف و تنظیمات پیش فرض و یک لیست کاربران مجاز، استفاده شده است. از عناصر متفاوتی در ارتباط با Forms Authentication در فایل Web.Config استفاده می گردد.هر یک از عناصر دارای خصلت های خاص خود می باشند :

عنصر <authentication>

خصلت Mode ، با استفاده از خصلت فوق ، روش تائید و شناسائی کاربران مشخص می گردد. با مقدار دهی خصلت فوق به Forms ، روش Authentication Forms انتخاب خواهد شد.

عنصر <forms>

خصلت name . از خصلت فوق به منظور مشخص نمودن نام کوکی که اطلاعات مربوط به نام و رمز عبور را ذخیره می نماید ، استفاده می شود . مقدار پیش فرض ، authaspx . می باشد . در صورتیکه بیش از یک برنامه بر روی سرویس دهنده از روش Forms Authentication استفاده می نمایند ، می بایست برای هر یک از آنان نام منحصربفردی در نظر گرفته شود .

خصلت loginUrl .از خصلت فوق به منظور مشخص نمودن نام فرم وب Login برای کاربران تائید نشده ، استفاده می گردد . مقدار پیش فرض خصلت فوق، Default.aspx است .

خصلت protection . با استفاده از خصلت فوق روش حفاظت کوکی Authentication که بر روی کامپیوتر سرویس گیرنده ذخیره می گردد ، مشخص خواهد شد. مقدار پیش فرض خصلت فوق ، All بوده که عملیات رمزنگاری و بررسی اعتبار و صحت داده در رابطه با آن اعمال می گردد. سایر گزینه های موجود در این راستا ، Encryption,Validation و None می باشد .

خصلت timeout . با استفاده از خصلت فوق ، مدت زمان نگهداری کوکی Authentication بر روی ماشین کاربر مشخص می گردد . مقدار پیش فرض ۳۰ دقیقه است . ASP.NET ، پس از دریافت یک درخواست جدید توسط کاربر و مشروط به گذشت بیش از نصف زمان تعریف شده ، کوکی را تجدید (Renew) خواهد کرد .

خصلت path . با استفاده از خصلت فوق ، مسیر مورد نظر به منظور ذخیره سازی کوکی بر روی ماشین کاربر مشخص می گردد . مقدار پیش فرض ، "\" است .

عنصر <credentials>

خصلت passwordFormat ، با استفاده از خصلت فوق ، الگوریتم لازم به منظور رمزنگاری رمز عبور کاربر ، مشخص می گردد . مقدار پیش فرض ، SHA1 می باشد . سایر گزینه های موجود در این رابطه ، MD5 و Clear (بدون رمزنگاری) می باشد .

عنصر <users>

خصلت name ، با استفاده از خصلت فوق ، نام کاربر مشخص می گردد.

خصلت password ، با استفاده از خصلت فوق ، رمز عبور کاربر مشخص می گردد.

عنصر <credentilas> ، امکان ذخیره سازی لیست کاربران را در Web.Config فراهم می نماید . رویکرد فوق ، روشی ساده به منظور تعریف کاربران مجاز یک برنامه وب می باشد . در چنین مواردی ، مدیریت سیستم می تواند بسادگی و در صورت لزوم نام و رمز عبور کاربران دیگری را به لیست مجاز کاربران ، اضافه نماید . مکانیزم فوق ، در مواردی که قصد داشته باشیم ، امکان تعریف نام و رمز عبور را در اختیار کاربران قرار دهیم ، گزینه مناسبی نبوده و می بایست از یک فایل و یا بانک اطلاعاتی به منظور ذخیره سازی اطلاعات کاربران ، استفاده گردد