تشخیص هویت، حق دسترسی، اشتراک کاربری و نقش::

امنیت پایه ای و حمایت ASP.NET

مقدمه:

چه چیزی در بین فروم‌ها، سایت‌های تجارت الکترونیک و وب سایت‌های ایمیل آنلاین و شبکه جامعه مجازی، مشترک است؟ همه آنها حساب کاربری را پیشنهاد می‌دهند. سایت‌هایی که حق کاربری را پیشنهاد می‌دهند باید یک تعداد سرویس را تامین کنند. حداقل، بینندگان جدید نیاز دارند تا قادر باشند که یک حساب بگیرند و درمراجعه بیننده‌ها باید قادر باشند تا Log IN کنند. چنین Web Applcationهایی می‌توانند بر اساس ورود کاربر تصمیم بگیرند که بعضی صفحه‌ها یا فعالیت‌ها فقط برای کابران ثبت نام شده انجام بشود، یا برای یک قسمتی از کاربران، صفحات دیگر اطلاعات خاصی به کاربر ثبت نام شده نشان بدهند و به ثبت نام نشده‌ها اطلاعات کمتری نشان بدهد، و به اینکه چه کاربری چه صفحه را مشاهده می‌کند بستگی دارد.

این اولین آموزش در سری آموزش‌هایی است که تکنیک‌هایی را برای تشخیص هویت بینندگان از طریق یک فرم وب، دسترسی به صفحات خاص، و مدیریت حساب‌های کاربری در ASP.NET را نشان می‌آموزش می‌دهد. در انتهای این آموزش‌ها ما بررسی می‌کنیم که چگونه:

* کاربران را تعیین هویت کنیم و به وب سایت وارد کنیم.
* استفاده از Membership framework برای مدیریت حسابهای کاربری
* ایجاد، به روز آوری و حذف کردن حسابهای کاریری
* محدود کردن دسترسی به صفحات وب، پوشه‌ها، یا توابع مخصوص پایه‌ای در ورود کاربر
* استفاده از Roles framework برای ارتباط دادن حسابهای کاربری با نقش آنها به وسیله کاربر مدیر نقش‌ها.
* سفارشی کردن و توسعه دادن وب کنترل‌های امنیتی ASP.NET

این آموزشها مرحله به مرحله دستورالعمل‌ها را با مقدار کافی کد و نمایش آنها به طور مختصر، موارد بالا را فراهم می‌کند. هر مقاله آموزشی مدلهای C# و Visual Basic را و دانلود کامل کدهای استفاده شده در بردارد.(این اولین مقاله آموزشی بر روی مفهوم امنیت از یک دیدگاه سطح بالا تمرکز دارد و بنابراین مرتبط با هیچ کدی نیست.)

در این آموزش ما درباره مفهوم مهم امنیت و اینکه چه امکاناتی در ASP.NET برای کمک به انجام تشخیص هویت در فرم‌ها، دسترسی مجاز، حساب کاربری و نقش‌ها وجود دارد را بحث می‌کنیم. اجازه بدهید شروع کنیم!

توجه: امنیت یکی از جنبه‌های مهم هر کاربردی است که محدودیت‌های فیزیکی، تکنیکی، سیاست‌های تصمیم‌گیری و نیازمندیهای درجه یک امنیتی را دارد. این سری آموزش‌ها برای برنامه نویسان کاربردهای امنیتی مرجع نیست و ترجیحا تمرکز روی تشخیص هویت فرم، دسترسی مجاز، اشتراک کاربر و نقش‌ها تمرکز دارد.

تشخیص هویت، دسترسی، اشتراک کاربری و نقش

تشخیص هویت، دسترسی، اشتراک کاربری و نقش چهار موردی هستند که به طور اغلب در تمام این سری از آموزشها استفاده می‌شوند. بنابراین من می‌خواهم تا لحظاتی را در تعریف این اصطلاحات مستقل از امنیت در وب بگذرانم. در یک مدل مشتری \_ سرور مثل اینترنت، چندین سناریو وجود دارد که در آنها سرور به تشخیص مشتری از طریق درخواست او نیاز دارد. تشخیص هویت پروسه معلوم کردن هویت مشتری است. به یک مشتری که با موفقیت شناسایی شده تشخیص هویت شده یا تصدیق شده گفته می‌شود. به یک مشتری شناخته نشده تشخیص هویت نشده یا غریبه گفته می‌شود.

سیستم‌های تشخیص هویت ایمن حداقل شامل یکی از سه شکل زیراند: چیزهایی که شما می دانید، چیزهایی که شما دارید یا چیزهایی که شما هستید. بیشتر برنامه های کاربردی وب به چیزهایی از قبیل کلمه عبور که مشتری می‌داند متکی است. اطلاعات استفاده شده برای شناختن یک کاربر، نام کاربری و کلمه عبور است که به اعتبارنامه او منسوب می‌شوند. این مجموعه آموزشی روی فرم‌های تشخیص هویت تمرکز دارد، که یک مدل تشخیص هویت برای کاربرانی است که وارد می شوند به سایت مشروط بر اعتبار نامه شان. بروید به هر سایت تجارت الکترونیک. وقتی شما آماده بشوید برای تصفیه حساب، از شما خواسته می شود تا وارد بشوید به وسیله وارد کردن نام کاربری و کلمه عبور در textbox روی صفحه وب.

علاوه بر شناسایی مشتری، یک سرور نیاز دارد به محدود کردن اینکه چه منابعی یا توابعی باید در دسترس باشند منوط به اینکه چه کاربری اجازه دسترسی به منابع یا توابع مخصوص را دارد.

یک حساب کاربری هست یک محل ذخیره اطلاعات برای به یادآوری درباره یک کاربر خاص. حساب کاربری حداقل باید شامل اطلاعات که به طور منجصر به فرد مشخص کند هویت کاربر را، از قبیل نام و کلمه عبور ورود به سیستم. علاوه بر این اطلاعات ضروری و واجب، حساب کاربری باید شامل مواردی مثل اینها باشد: نام پست الکترونیکی کاربر، تاریخ و زمانی که حساب کاربری ایجاد شده است، تاریخ و زمانی که آخرین بار به سیستم وارد شده است، نام و نام خانوادگی، شماره تلفن، و آدرس پستی. وقتی استفاده می کنید از فرم تشخیص هویت، اطلاعات حساب کاربری عموما ذخیره می شود در پایگاه داده ای مثل Microsoft SQL Server.

برنامه های کاربردی وب که از حساب کاربری پشتیبانی می‌کنند، به طور انتخابی گروه کاربران را به نقش ها تبدیل می‌کنند. یک نقش یک برچسب ساده است که به یک کاربر الحاق می شود و یک ایزوله سازی را برای معین کردن نقش‌های دسترسی و صفحات سطح بندی شده فراهم می‌کند. برای مثال، یک وب سایت شاید شامل یک نقش Administrator با قوانین دسترسی‌ای باشد که غریبه ها را منع کرده باشد، اما یک Administator به تمام مجموعه از صفحات وب دسترسی دارد. علاوه بر این، صفحات متنوعی در دسترس برای همه کاربران(شامل غیر از Administrator) وجود دارند که شاید اطلاعات اضافی را نمایش بدهد یا عرضه کند توابع اضافی را وقتی که مشاهده می شوند به وسیله کاربران در نقش Administator. ما می‌توانیم این قوانین دسترسی مجاز را بر اساس نقش افراد به جای نام کاربران تعریف کنیم.

تشخیص هویت کاربران در کاربردهای ASP.NET

وقتی یک کاربر وارد در پنجره آدرس مرورگر یک آدرس وارد می‌کند یا روی یک لینک کلیک می‌کند، مرورگر یک درخواست HTTP(پروتکل انتقال اطلاعات حساس به متن) می‌سازد و برای بدست آوردن محتوای تعیین شده که هست یک صفحه ASP.NET یا یک تصویر یا یک فایل جاوا یا هرگونه محتوایی به سرور می‌فرستد. وب سرور محتوای درخواست را برمی‌گرداند. در حین انجام کار، باید یک تعداد مواردی را درباره آن درخواست مشخص کند، شامل اینکه چه کسی آن درخواست کرده و اینکه آیا هویت شناسایی شده حق دسترسی و بازیابی اطلاعات درخواست شده را دارد.

به صورت پیش فرض، مرورگر درخواست HTTP که فاقد تقریبا هرگونه اطلاعات شناسایی است را می‌فرستد. اما اگر مرورگر شامل اطلاعات تشخیص هویت فرد باشد، سپس وب سرور شروع می کند به گردش کار تشخیص هویت که تلاش می‌کند تا مشتری‌ای را که درخواست را صادر کرده است شناسایی کند. مراحل کاری تشخیص هویت کاربر بستگی به نوع تشخیص هویتی که برنامه کاربردی وب استفاده از آن می‌کند دارد. ASP.NET از سه نوع تشخیص هویت ویندوز، گذرنامه و فرم پشتیبانی می‌کند. این سری آموزش تمرکز دارد روی تشخیص هویت فرم، اما اجازه بدهید چند دقیقه آن را با تشخیص هویت ویندوزی مقایسه کنیم.

تشخیص هویت به وسیله تشخیص هویت ویندوزی

گردش کار تشخیص هویت ویندوزی از یکی از تکنیک‌های تشخیص هویت زیر استفاده می‌کند:

* تشخیص هویت پایه ای
* تشخیص هویت خلاصه
* تشخیص هویت جامع ویندوزی

همه این سه تکنیک با راهی تقریبا مثل هم کار می‌کنند: وقتی فرد بدون اجازه یک درخواست ناآشنا می‌فرستد، وب‌سرور یک پاسخ HTTP که نشان می‌دهد اجازه برای ادامه نیاز است را برمی‌گرداند. سپس مرورگر یک نمونه جعبه تبادل دوطرفه را نمایش می‌دهد که از کاربر نام کاربری و کلمه عبور او را درخواست می‌کند (ببینید در شکل 1). این اطلاعات سپس برگردانده می شود به وب سرور به وسیله یک بلوک HTTP.

شکل 1: یک مدل از دیالوگ تبادل دو طرفه که از کاربر اعتبارنامه او را می‌خواهد.

اعتبارنامه‌ها در وب سرور ویندوز تایید می شوند. این به معنی این است که هر کاربر تشخیص هویت شده در برنامه کاربردی وب باید یک حساب کاربری در سازمان شما داشته باشد. این در سناریوی اینترانت پیش پا افتاده است. در حقیقت وقتی از تشخیص هویت جامع ویندوز در یک اینترانت استفاده می‌کنید، مرورگر به طور خودکار وب سرور را با اعتبارنامه‌ای برای ورود به شبکه آماده می‌کند، که جعبه دیالوگ آن در شکل 1 نشان داده شده است. در حین اینکه تشخیص هویت ویندوزی برای کاربردهای اینترانت مهم است، این معمولا برای برنامه کاربردی اینترنت ناشدنی است، زیرا شما نمی‌خواهید تا یک حساب کاربری برای هر کاربری که وارد وارد سایت شما می‌شود ایجاد کنید.

تشخیص هویت به وسیله فرم تشخیص هویت

تشخیص هویت فرم، از طرف دیگر برای برنامه های کاربردی وب مطلوب هست. فراخوانی می کند آن فرمهای تشخیص هویت تایید شده کاربر را بوسیله فراخوانی آنها به وارد نمودن اعتبارشان از طریق یک فرم وب. در نتیجه وقتی یک کاربر تلاش می کند تا دسترسی پیدا کند به یک منبع غیرمجاز در وب، آنها به صورت اتوماتیک راهنمایی می شوند به صفحه ورود، جایی که آنها می توانند وارد کنند اعتبارنامه شان را. اعتبارنامه ارائه شده و سپس تایید اعتبار می شود با یک انباری از کاربران عرف – معمولا یک پایگاه داده.

بعد از بررسی کردن اعتبار نامه ارائه شده، یک بلیط برای کاربر صادر می‌شود. این بلیط به اینکه چه کاربری تشخیص هویت شده است و اینکه شامل چه اطلاعات شناسایی از کاربر، از قبیل نام کاربری است اشاره دارد. بلیط فرم تشخیص هویت در یک کوکی روی کامپیوتر مشتری ذخیره می‌شود. بنابراین، در انتهای یک دیدار از یک سایت که شامل بلیط تشخیص هویت است وب سرور قادر خواهد بود تا دفعه بعد که مشتری از سایت دیدن می کند او را شناسایی و وارد کند.

شکل 2 به صورت مصور نشان می دهد گردش کاری فرم تشخیص هویت را از نقطه مناسبی از سطح بالا. توجه کنیدکه چگونه قطعه های تشخیص هویت و حق دسترسی در ASP.NET به صورت دو موجودیت جداگانه عمل می کنند. سیستم تشخیص هویت فرم شناسایی می کند کاربر را(یا گزارش می کند که غریبه است). سیستم حق دسترسی تصمیم می گیرد که آیا کاربر حق درخواست منابع را دارد. وقتی که کاربر حق دسترسی ندارد(همانطوری که در شکل 2 نشان داده شده وقتی یک کاربر غریبه تلاش می کند تا ببیند ProtectedPage.aspx)، سیستم تشخیص هویت گزارش می دهد که کاربر هست غیرمجاز، و فرم تشخیص هویت فرد را به صفحه ورود هدایت می کند.

یکبار که فرد موفق شد تا با موفقیت وارد شود، درخواست بعدی HTTP شامل بلیط فرم تشخیص هویت خواهد بود. سیستم تشخیص هویت فرم صرفا شناسایی کاربر – این سیستم حق دسترسی است که تصمیم می گیرد که کاربر می تواند دسترسی داشته باشد به منبع درخواست شده.



شکل 2: گردش کاری فرم تشخیص هویت

ما کاوش می کنیم در تشخیص هویت فرم با جزئیات بیشتری در دو آموزش بعدی، یک مروری بر تنظیمات تشخیص هویت فرم و حق دسترسی فرم و مباحث پیشرفته. برای انتخابهای بیشتری در تشخیص هویت ASP.NET، ببینید [ASP.NET Authentication](http://msdn2.microsoft.com/en-us/library/eeyk640h.aspx).

محدودیت دسترسی به صفحات وب، پوشه ها و صفحات توابع

ASP.NET دو راه برای تصمیم گیری دارد در مورد اینکه آیا کاربر اجازه نفوذ برای دسترسی به یک فایل یا پوشه مخصوص را دارد:

* حق دسترسی به فایل – صفحات ASP.NET و سرویس های وب، اجرا می شوند روی فایل هایی که قرار می گیرند روی سیستم فایل وب سرور، دسترسی به این فایل ها می تواند مشخص بشود از طریق لیست کنترل دسترسی(ACLs). دسترسی به فایل به طور عمومی استفاده می‌کند از تشخیص هویت ویندوزی، زیرا ACLs هست اجازه ای که اجرا می شود از روی حساب ویندوزی. وقتی استفاده می‌کنید از تشخیص هویت فرمی، همه سیستم عامل ها و سیستم های فایل برای درخواست ها اشتراک ویندوزی را اجرا می کنند، قطع نظر از اینکه چه کاربری سایت را مشاهده می‌کند.
* حق دسترسی به آدرس – صفحات گسترش دهنده مشخص می کنند قوانین حق دسترسی را در Web.config. این قوانین دسترسی مشخص می کند که چه کاربری یا نقشی اجازه دارد یا غیرمجاز است تا دسترسی پیدا کند به دستیابی مسلم به صفحات یا پوشه ای در برنامه کاربردی ما.

بر اساس حق دسترسی به فایل یا آدرس، تعریف می شود قوانین حق دسترسی برای دستیابی به یکسری صفحات خاص یا همه صفحات موجود در یک دایرکتوری خاص. با استفاده از این تکنیک ها ما می توانیم دستور بدهیم به ASP.NET تا رد کند درخواست را برای دسترسی به یک صفحه خاصی برای یک کاربر خاص، یا اجازه بدهیم به مجموعه ای از کاربران و دسترسی همه افراد دیگر را غیرمجاز کنیم. در مورد صفحاتی که همه کاربران به آن دسترسی دارند، اما کارکرد صفحات بستگی به کاربر دارد؟ برای مثال، تعدادی از سایتهایی که حمایت می کنند از اشتراک کاربری، دارند صفحاتی را که نمایش می دهند محتوای یا اطلاعات مختلفی را برای کاربر تشخیص هویت شده، در مقابل کاربر میهمان. یک کاربر غریبه شاید ببیند یک لینک را برای ورود به سایت، در حالی که کاربر تشخیص هویت شده می بیند یک پیغام مثل "خوش آمدید کاربر ..." همراه با یک لینک برای خروج از سایت. مثال دیگر: وقتی شما می بینید یک جنسی را در یک سایت حراج، شما می بینید اطلاعات مختلفی را در مورد آن، بسته به اینکه شما هستید یک شرکت کننده در یک مناقصه یا یک برگزار کننده حراج هستید.

برای نشان دادن اینکه محتوای صفحات برای یک کاربر غریبه هست متفاوت تر از کاربر تشخیص هویت شده، یک کنترل LginView روی صفحه بیندازید و وارد کنید محتوای متناسب با کاربر، در AnonymousTemplate برای کاربر غریبه و LoggedInTemplate برای کاربر وارد شده به سایت. متناوبا، شما می توانید به صورت اتوماتیک تعیین کنید که درخواست جاری چه کاربری است، و چه نقشی دارد(اگر وجود دارد). شما می توانید همچنین استفاده کنید از این اطلاعات تا سرانجام نمایش بدهید یا مخفی کنید ستونهایی در یک جدول یا پنل در صفحه.

این مجموعه شامل سه مقاله آموزشی در تمرکز روی تشخیص هویت است. تشخیص هویت بر اساس کاربر مشخص می کند که چگونه محدود کنیم دسترسی به یک صفحه یا یک صفحاتی در یک پوشه برای یک سری اشتراک های خاصی. تشخیص هویت بر اساس جستجو می کند کاربران تشخیص هویت شده را براساس نقش آنها. در آخر، محتوای نشان داده شده برپایه کاربر وارد شده نشان می دهد یک محتوای صفحه خاص و طبقه بندی بر اساس کاربری که صفحه را مشاهده می کند. برای مشاهده گزینه های بیشتر در تشخیص هویت ASP.NET، ببینید [ASP.NET Authentication](http://msdn2.microsoft.com/en-us/library/wce3kxhd.aspx).

اشتراک های کاربری و نقش ها

تشخیص هویت فرمی ASP.NET تهییه می کند یک زیربنایی برای کاربران برای ورود به سایت و نگه می دارد حالت تشخیص هویت شده آنها را برای مشاهده شده تمام صفحات. و حق دسترسی از طریق آدرس به ما پیشنهاد می دهد یک چهارچوب برای محدود کردن دسترسی به فایلها یا پوشه ها در یک اشتراک کاربری و مدیریت نقش ها.

پیش از این در ASP.NET 2.0، برنامه نویسان مسوول بودن برای ایجاد اشتراک کاربری و مدیریت نقش ها. آنها همچنین گیج بودن برای طراحی واسط کاربری و نوشتن کدهای اساسی برای صفحات وابسته به اشتراک کاربری مثل صفحه ورود کاربر و صفحه ایجاد یک کاربر جدید. بدون هر گونه Frame work داخلی اشتراک کاربری در ASP.NET، هر برنامه نویس اجرا می کند اشتراک کاربری را طبق نظر خود برای طراحی در مورد سوال هایی مثل این، "چگونه کلمه عبور یا اطلاعات مهم دیگر را ذخیره کنم؟" و "چگونه باید من رهبری کنم اعمال نفوذ بر طول کلمه عبور و استحکام آن را؟"

امروزه، اجرا کردن اشتراک کاربری در کاربردهای ASP.NET هست ساده هست در Membership framework و بوسیله Login Web controls. Membership framework هست یک پر از کلاسهایی در System.Web.Security که تهییه می کند توابعی برای اجرا کردن وظایف وابسته به اشتراک کاربری. کلاسهای کلیدی در Membership framework هست کلاسهای Membership که دارد متدهایی مانند زیر:

* CreateUser
* DeleteUser
* GetAllUser
* GetUser
* UpdateUser
* ValidateUser

Membership framework استفاده می کند از یک مدل سرویس دهنده ای که به طور واضحی جدا می کند APIهای Membership framework را در اجرا. این قادر می سازد برنامه نویسان را تا استفاده کنند از APIهای عمومی، اما قدرت می دهد به کاربر تا استفاده کند از اجرای آنهایی که برآورده می کنند نیازهای سفارشی او را. به عبارت ساده تر، کلاس Membership تعریف میکند توابع اساسی Framework(متدها، صفات و رخدادها)، اما واقعا هیچ گونه از جزئیات اجرا را فراهم نمی کند. در عوض متدهای کلاس Membership طلب می کنند تنظیمات سرویس را، که این ایفا می کند نقش واقعی را. برای مثال، وقتی وقتی متد کلاس Membership درخواست می شود، کلاس Membership نمی داند جزئیات ذخیره کاربر را. کلاس Membership بررسی می کند که تنظیمات برنامه کاربردی وب برای تعیین کردن اینکه چه سرویس دهنده ای نمایندگی دارد تا فراخوانی بشود و کلاس آن سرویس دهنده هست مسوول برای ایجاد یک اشتراک کاربری جدید در یک جایگاه ذخیره کاربری مناسب. این فعل و انفعال نشان داده شده در شکل 3.

مایکروسافت دارد دو کلاس سرویس دهنده در .NET Framework

* ActiveDirectoryMembershipProvider

اجرا می کند توابع کاربردی API را در Active Directory و ADAM.

* SqlMembershipProvider

اجرا می کند توابع کاربردی API را در پایگاه داده SQL Server.

این مجموعه آموزشی منحصرا تمرکز دارد روی sqlMembershipProvider.



شکل 3: مدل سرویس دهنده فعال می کند اجراهای متفاوتی را در چهارچوب کاری.

منفعت مدل های سرویس دهنده هست جای گزین کردن اجراهایی که می تواند نوشته شود توسط مایکروسافت. برای مثال، مایکروسافت حاصل کرده یک Membership provider for Microsoft Access databases. برای اطلاعات بیشتر در مورد سرویس دهنده ها، مراجعه کنید به [Provider Toolkit](http://msdn2.microsoft.com/en-us/asp.net/aa336558.aspx)، که شامل آموزش(walkthrough) Membership providers است، سرویس دهنده های سفارشی ساده، بالای 100 صفحه مقاله در مورد مدل سرویس دهنده، . سورس کدهای کامل برای Membership provider طراحی شده (به نام های ActiveDirectoryMembershipProvider و SqlMembershipProvider).

ASP.NET 2.0 همچنین معرفی کرده است Roles framework را. مثل Membership framework، Roles framework نیز ساخته شده بر روی مدل سرویس دهنده. توابع کاربردی آن آشکار می شود بوسیله Roles Class و اعضای .NET Framework با سه کلاس سرویس دهنده:

* AuthorizationStoreRoleProvider

مدیریت می کند اطلاعات نقش ها را بر اساس قوانین مدیریت Authorization ذخیره شده، از قبیل Active Directory یا ADAM.

* SqlRoleProvider

اجرا می کند نقش ها را در پایگاه داده SQL Server.

* WindowsTokenRoleProvider

مرتبط می کند اطلاعات را بر اساس گروههای ملاقات کننده ویندوز. این متدها نوعا استفاده می شود با تشخیص هویت ویندوزی.

این مجموعه از آموزش ها تمرکز دارد بر روی SqlRoleProvider.

بیشتر بخوانید

برای اطلاعات بیشتر در مورد بحث های بالا در این مقاله آموزشی، مراجعه کنید به منابع زیر:

 ASP.NET 2.0 Membership, Roles, Forms Authentication, and Security Resources

 ASP.NET 2.0 Security Guidelines

 ASP.NET Authentication

 ASP.NET Authorization

 ASP.NET Login Controls Overview

 Examining ASP.NET 2.0’s Membership, Roles, and Profile

 How Do I: Secure My Site Using Membership and Roles? (Video)

 Introduction to Membership

 MSDN Security Developer Center

 Professional ASP.NET 2.0 Security, Membership, and Role Management (ISBN: 978-0-7645-9698-8)

 Provider Toolkit

 Securing Your Web Site with Membership and Login Controls (Video)

درباره مولف

Scott Mitchell، مولف چندین کتاب در مورد ASP/ASP.NET و موسس 4GuysFromRolla.com و کار می کند با تکنولوژی مایکروسافت از سال 1998. Scott Mitchell کار می کند به عنوان یک مشاور مستقل، مربی و یک نویسنده. آخرین کتاب او هست [Sams Teach Yourself ASP.NET 2.0 in 24 Hours](http://www.amazon.com/exec/obidos/asin/0672327384/a4uysfromrollaco).

ایمیل: [mitchell@4guysfromrolla.com](mailto:mitchell@4guysfromrolla.com) یا از طریق http://ScottOnWriting.NET.