Autorizar Y Proteger Los Recursos Web Con ASP.NET

Los **sitios web** cuentan normalmente con múltiples secciones diseñadas para los usuarios, manejan diferentes roles, también puede utilizar este tipo de identificación para **administrar el acceso a los diferentes recursos en el sitio web**, los cuales están para cada necesidad de los usuarios.

A modo de ejemplo, en una **intranet corporativa**, pueden haber páginas que contienen los informes y datos confidenciales que sólo deben ser vistos por los departamentos específicos o altos directivos y no por todos los empleados de la empresa. En la mayoría de los casos, sólo ciertos administradores tendrían acceso completo para cambiar la configuración de la página web con otros usuarios tal vez se conceden menos capacidad de cambio. En todos estos casos, tenemos que restringir estas opciones para la mayoría de los usuarios, al tiempo permitir ciertos persmisos a otros usuarios. La autorización le permite asegurar si un usuario puede tener acceso a todo lo que necesita en un sitio web.

Como **desarrollador web**, tienes que manejar de gran manera la seguridad de un sitio web, ya que es un aspecto importante de la seguridad, la cual trata de garantizar que los usuarios no tengan acceso a las acciones que no se deben realizar. Las consecuencias de no protegerla, pueden ser graves. Te invitamos a dar un vistazo a una visión general del manejo de permisos y cómo implementarlo con ASP.NET.

¿Qué es la autorización?

Dar cuentas únicas a las personas nos permite identificar **quien accede a nuestro sitio web**. Una vez que sabemos el usuario, podemos adaptar el sitio para las necesidades de él, y personalizar el sitio web para reflejar la información que sabemos sobre el usuario. Esto puede variar desde aspectos simples tales como un sitio web diciendo: «Hola Bill,» al iniciar sesión.

La mayoría de los sitios de comercio pueden almacenar datos de las tarjetas de crédito, dirección y otra información que necesitan para realizar un pedido. Esto significa que no es necesario introducir la misma información cada vez que se ingrese a dicho sitio. Los sitios web también pueden ir más lejos usando un historial y hábitos para sugerir productos relacionados que podría estar interesado el usuario. También podrías ser capaz de establecer las preferencias en colores, categorías u otros valores del sitio que puede usar.

Eso puede proporcionar beneficios útiles a los usuarios del sitio, pero la autorización se centra en el uso de la identidad única del usuario para determinar las acciones que el usuario puede realizar. Permite que el sitio web determinar si un usuario debe tener la capacidad de acceder a las diferentes secciones de un sitio web, podrá acceder a los datos, o ser capaz de hacer cambios a los datos.

Control del acceso con los Grupos y Roles

Si bien es posible **proporcionar derechos y responsabilidades** únicas para cada usuario de un sitio web, esto se convierte rápidamente inmanejable cuando el número de usuarios crece. Rápidamente, la posibilidad de errores se incrementa con cada nuevo usuario que tenga una configuración personalizada. Si cualquier cambio en el sitio requiere nuevos permisos o ajustes, entonces tendría que ser actualizado posiblemente requiriendo actualizaciones manuales a cientos o miles de cuentas de cada cuenta de usuario.

Por esta razón. los usuarios se agrupan normalmente junto con los que tienen derechos o necesidades similares. Los grupos a menudo se refieren a veces como roles ya que el rol del usuario en un sitio a menudo define los grupos que se irán a usar. Para cada grupo del administrador del sitio puede definir restricciones de acceso y dentro de la aplicación web.

A continuación, asigne usuarios a estos grupos y el usuario tomará sobre los derechos y restricciones asignadas a ese grupo. Los derechos pueden ser quitados simplemente quitando el usuario del grupo.

Como ejemplo, en el primer caso, es posible que tenga los «autores», «editores», «editoriales», etc En la segunda podría tener grupos para «crear el artículo,» «Editar un artículo,» «suprimir el artículo,» «publicar artículo». Este método

proporciona una mayor flexibilidad a cambio de la gestión de más grupos.

Proteger Formularios De Sitios Web Con ASP.NET

El enrutamiento que ofrece **ASP.NET** y ASP.NET Web Forms utilizan el **archivo web.config** para asegurar el acceso a páginas web. Una configuración básica para garantizar el acceso a un recurso en un sitio web podría ser similar a la siguiente:

[xml]

```
<configuration>
<location path="adminhome.aspx">
<system.web>
<authorization>
<allow roles="admin"/>
<deny users="*"/>
</authorization>
</system.web>
</location>
</configuration>
```

[/xml]

El elemento de ubicación de este fragmento de código XML **define la ruta del archivo**, carpeta o ruta que vamos a manejar. Aquí estamos especificando, que esto se aplica a la página **adminhome.aspx**. Esto también podría dar a una carpeta en el sitio y se aplicaría a esa carpeta. Si no se especifica ninguna ruta, se atribuyen los valores de configuración y se aplican al directorio actual del **archivo web.config** y todos los directorios secundarios.

El elemento de la autorización contiene los parámetros utilizados para establecer quién tiene acceso y quién se le niega el acceso al objeto especificado en el elemento de la ruta. Las reglas se comprueban empezando con la primera regla en orden hasta que se encuentra una coincidencia. El elemento de permitir que especifica las funciones y/o los usuarios que tendrán acceso al recurso. Del mismo modo el elemento que niega especifica usuarios y roles que no tienen autorización para acceder al recurso.

En este ejemplo, la regla **<allow admin role/>** se comprobará en primer lugar. Si el usuario está en el rol de administrador, a continuación, se les concede el acceso y nada más necesita ser comprobada. Si el usuario no está en ese rol, a continuación, ASP.NET sigue la siguiente regla. Aquí, la regla <deny users=»*»/> negaría todos los usuarios. Por tanto, este ejemplo podría permitir a los usuarios en el acceso rol de administrador. El resto de usuarios se les niega el acceso.

Hay unos pocos roles especiales para especificar grupos comunes. Vimos el usuario * anterior, que especifica todos los usuarios. El usuario se refiere a los usuarios anónimos, es decir, cualquier usuario que no ha iniciado la sesión. Múltiples usuarios y roles pueden especificarse separándolos con una coma. Usuarios y roles se pueden mezclar en la misma norma, tales como:

[xml]<allow roles="siteadmin,editors" users="bob">[/xml]

Protegiendo Sitios MVC Con ASP.NET

ASP.NET MVC se centra en los controladores y eventos sobre esos controladores en lugar de los archivos. Esto cambia el método de asegurar el acceso a un sitio de **ASP.NET MVC**. Por defecto, todos los eventos y los controladores se puede acceder a través de todos los usuarios, al igual que en WebForms. Sigues usando los mismos atributos y roles de usuario, pero no se establece estos dentro del archivo web.config.

En lugar de aplicar un atributo [Authorize] a los controladores y eventos directamente. Como ejemplo, si tienes un **AdminController** que sólo debe ser accesible por los miembros del rol de administrador, puedes hacer eso mediante la adición de los usuarios y/o roles de la etiqueta.

[xml]

```
[Authorize(Roles = "siteadmin")]
public class AdminController : Controller
{
…
```

[/xml]

El mismo * ? Puedes aplicar las reglas específicamente para una acción individual en el controlador para restringir sólo esas acciones. Los atributos especificados en una acción anularán las especificadas para todo lo que haya dispuesto el programador.

[xml]

```
[Authorize(Roles = "siteadmin")]
public ActionResult AdminView()
{
…
```

[/xml]

Si no se especifica ninguna función ó usuarios con el atributo [Authorize], entonces se permitirá a cualquier usuario autenticado iniciar sesión. Esto le permite el acceso a las acciones ó controladores para los usuarios que se registran específicamente en el sistema.

ASP.NET 4 agrega un atributo [**AllowAnonymous**] que le permite anular este para una acción dentro de un controlador.

Aspectos de seguridad de

sesiones de usuario

Mientras que el aspecto de autenticación difiere del elemento de autorización discutido, están relacionados entre sí. En primer lugar, en la sesión de inicio de sesión por lo general se establecen con un tiempo de espera en la configuración. En **ASP.NET**, se establece en el archivo web.config en la sección <**authentication**>.

[xml]<forms loginUrl="~/Auth/LogOn.aspx" timeout="30" slidingExpiration="true" />[/xml]

Esto establecería el tiempo de espera en 30 minutos, antes de finalizar automáticamente la sesión. El atributo **slidingExpiration** determina si una solicitud se restablece el contador a cero. Cuando se establece en false, un usuario tendría que volver a iniciar sesión en cada treinta minutos, incluso si se utiliza activamente el sitio todo ese tiempo.

También debemos ser conscientes del riesgo del robo de sesión. La mayoría de los **frameworks web** utilizan un identificador único para el usuario, una vez autenticado, normalmente son almacenados en una cookie. Si esta cookie no está protegida de alguna manera, entonces cualquiera que pueda ver el tráfico del usuario se puede utilizar la cookie de hacerse pasar como el usuario original.

Puedes proteger el robo de estas usando SSL, no sólo para las cookies, sino también para la autenticación del token que representa al usuario conectado. Esto asegura que la cookie

sólo se enviará cuando se accede a la página por SSL.

En **ASP.NET**, puede cumplir esta estableciendo el atributo **requireSSL = «true»** en las etiquetas </> que están en web.config cuando se utiliza la autenticación de formularios. Para mayor protección, también puede establecer el <httpCookies requireSSL = «true» /> en su web.config para establecer todas las cookies a por SSL sólo por defecto.

Finalmente

El uso de los sitios web por muchos usuarios con diferentes necesidades y responsabilidades requiere de métodos para prevenir el acceso no autorizado a datos y funciones sensibles. Puedes utilizar la identidad de un usuario para determinar qué permisos tiene el usuario y hacer cumplir esos **permisos dentro de la aplicación web**.

Se empieza por asegurar que las páginas y los eventos dentro de la aplicación web se limitan sólo a los usuarios que deben tener la capacidad de trabajar con ellos. Para que accedan distintos usuarios a las páginas en múltiples roles, se debe tomar cuidado para validar el usuario tiene el derecho de realizar las acciones solicitadas antes de realizarlos. Dado que la identidad del usuario define su acceso, también se debe tener cuidado de asegurarse de que los demás no pueden suplantar a un usuario con más derechos.