Written by Zack MIlls
Tuesday, 09 February 2010 15:04 -

.Net Authentication

http://support.microsoft.com/kb/891028#43

The mode is set to one of the authentication modes: **Windows**, **Forms**, **Passport**, or **None**. The default is Windows. If the mode is **None**, ASP.NET does not apply any additional authentication to the request. This can be useful when you want to implement a custom authentication scheme, or if you are solely using anonymous authentication and want the highest possible level of performance.

The authentication mode cannot be set at a level below the application root directory. As is the case with other ASP.NET modules, subdirectories in the URL space inherit authentication modules unless explicitly overridden. Back to the top

Forms-based authentication

Forms authentication is a system by which unauthenticated requests are redirected to an HTML form using HTTP client-side redirection. The user provides credentials and submits the form. If the application authenticates the request, the system issues a cookie that contains the credentials or a key for reacquiring the identity. Subsequent requests are issued with the cookie in the request headers. They are authenticated and authorized by an ASP.NET event handler using whatever validation method the application developer specifies.

Protecting static file types using forms authentication

By default, forms authentication protects only ASPX pages and any other .NET extensions. You can configure forms authentication to protect other static extensions such as .jpg, .gif, .html, .pdf, etc. To do this, map these extensions to aspnet_isapi.dll using IIS Manager as follows:

1. Open IIS Manager. To do so, click **Start**, click **Program Files**, point to **Administrative Tools**, and then click **Internet Information Services Manager**.
2. Find your application’s virtual folder, and right-click it (your application must be forms authentication-enabled).
3. Click **Properties**.
4. Click **Configuration**.
5. On the **Mappings** tab, click **Add**.
6. In the **Executable** box, click **aspnet_isapi.dll**, which will be located in the %windows folder%\Microsoft.NET\Framework\FrameworkVersion folder.

In the **Extension** box, type your extension (for example, .jpg).
Provide at-least “GET” verb.
Click to clear the **Check that File Exists** check box.
Click **OK** for rest of the dialog boxes.

Protecting classic ASP pages using forms authentication

Protecting classic ASP pages with forms authentication is not supported by design because ASP and ASP.NET use different handlers. However, you can make it work using the help of COM-Interop and Web services.

The following sample should work. This would have been pretty easy using simple COM Interop to call into the **FormsAutentication** utility functions. However, the functions require an HttpContext, which is only available in an ASP.NET application.

As a workaround, create an ASP.NET Web service that does the forms authentication ticket validation.