

Re: Single Sign On

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2005-04/msg02971.html>

- *From:* "Dave Fancher" <ejjitek@xxxxxxxxxxxx>
 - *Date:* Tue, 12 Apr 2005 22:43:06 -0500
-

I work for a \$9.7 bil. company (Truck manufacturer) that is using Tivoli Access Manager (from IBM) for single sign on. I have recently inherited a project (traditional ASP) that authenticates against TAM and am developing another (ASP.NET) that will authenticate against it. Take a look at <http://www-306.ibm.com/software/tivoli/products/access-mgr-e-bus/> for more information.

I don't have much first-hand experience with it yet but the documentation I have read makes it sound pretty straight-forward to implement and some other developers I have spoken with seem to like it. TAM is being used heavily by our dealer network and we are slowly migrating to it for our purchasing systems.

HTH

Dave Fancher
<http://davefancher.blogspot.com>

"EricMatz" <EricMatz@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:5A058EEC-FCC2-4E4F-8DE8-FC82F2376176@xxxxxxxxxxxxxxxxxxxx
>I work for a medium-sized insurance company, developing web-based systems
>for
> our independent agents. There are four primary applications we provide –
> one
> that serves as an agent portal (ASP), and three others that serve
> applications specific to each line of business (one ASP, two ASP.NET).
> Each
> of these applications reside on their own server/environment, all of which
> are in the same AD domain. We also have multiple top-level domains that
> different sets of users visit to access the sites.
>
> Currently, this is how we are tackling Single Sign On. The user logs in
> to
> the portal site which is locked down via Basic authentication (okay,
> "locked
> down" and "Basic authentication" in the same sentence...I know...I know).
> Once authenticated, the portal site makes a call to an ASP.NET web service
> that creates and returns a FormsAuthentication ticket for the user that

Re: Single Sign On

> logged in. It then sets a persistent cookie containing that ticket. When
> the user accesses one of the ASP apps on another server, we check for the
> cookie and call in to the same web service to decrypt it and extract the
> user
> – at that point, their session is built and they are logged in. When the
> user accesses one of the ASP.NET apps, it is a bit easier. The machine
> keys
> are all in sync, so enabling Forms Authentication allows the ASP.NET app
> to
> recognize the cookie as a ticket, and the user is logged in. If a user
> visits any of the LOB sites without going through the portal, they are
> presented with that site's login page which essentially does the same
> thing
> as the portal, except the cookie is not persistent.
>
> Here is how I envision it working in a perfect world. The user logs in to
> our portal site using their account which is stored in AD. When they
> visit
> any of the other LOB apps, they do not have to provide credentials a
> second
> time. True Single Sign-On. Granted, this is pretty much how it works
> today,
> but I would prefer something more secure. Now my questions. Given our
> mutli-server environment and the mix of ASP/ASP.NET, what other SSO
> options
> do we have? I've read bits and pieces about Active Directory's ability to
> provide an SSO experience, but I have been coming up short finding out how
> to
> implement this when the bulk of our users are outside the corporate
> firewall.
> Can anyone point me in the right direction? I am not opposed to
> third-party
> solutions if they will meet our needs and are easy to implement/integrate.
> I
> am interested in hearing others' experiences with this as well.
>
> Thanks in advance!
> Eric Matz

• **References:**

◆ **Single Sign On**

◇ *From:* EricMatz

- Prev by Date: **Re: send entire page by email**
- Next by Date: **RE: Changing controls after event fires.**
- Previous by thread: **Single Sign On**

Re: Single Sign On

- Next by thread: *how can i call IE print preview function after i press asp:button?*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*