

Passing authentication credentials from portal to application using IIS 6

Source:

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2003-04/0354.html>

From: Halcyon Woodward (*kenbeaver_at_pacbell.net*)

Date: 04/30/03

Date: Tue, 29 Apr 2003 15:48:21 -0700

We are developing a Sharepoint portal solution that exposes data and links to custom ASP.Net applications via custom built webparts. These applications reside in separate application pools, on separate virtual and/or physical Windows 2003 servers.

When the user connects to the Sharepoint portal, they are prompted for credentials which are authenticated via Active Directory. The credentials are handled natively by IIS 6, and are transported via clear-text (basic authentication) over SSL.

When the user links out to a custom web application from the portal, they are prompted *_again_* for their credentials (our applications all use the same authentication methods that we've set up for Sharepoint.) We would like to implement a single-sign-on (SSO) solution for these applications instead, making it much easier on our users.

Because of the architecture of our custom webparts, we can programmatically redirect the user from the portal to the applications via back-end code; thus it would be possible to pass the credentials to the application via a redirect URL by examining the server variables sent in the portal's HTTPS request headers:

```
https://[username]:[password]@app1.example.com/default.aspx
```

Once the application has received the initial URL, it can redirect again to a URL that doesn't contain the username/password, as the credentials for that virtual server should now be established. (This is an assumption, rather than fact – I haven't verified this yet). This way the user doesn't freak-out by seeing their username and password in their browser address bar.

Question 1: Is this secure enough to use over the internet? I don't know enough about SSL and redirect header tags to know if the URL with the inline username and password is certificate encrypted or not.

Question 2: Is there another methodology for accomplishing the same objective using all the fancy new IIS 6 & .Net enhancements?

Anyone's help is greatly appreciated,

hb.