Single Sign-On PowerCAMPUS Portal Web Parts

Presented by: Chad Sexton, Technology Strategy

April 4, 2006
Tuesday, 10:30 am
Evaluation Code 510
Session Rules of Etiquette

- Please turn off your cell phone/pager
- If you must leave the session early, please do so as discreetly as possible
- Please avoid side conversation during the session

Thank you for your cooperation!
Overview

- SunGard Higher Education’s vision for higher education is to help every institution create the Unified Digital Campus (UDC), an environment in which systems, individuals, and communities interact seamlessly for learning, teaching, administration, and achievement. A Portal serves as a very important piece of the UDC because it provides institutions with the ability to unify and manage their core academic, administrative and community web applications in a common platform that ultimately provides a seamless, secure, and personalized atmosphere.

- The portal solution chosen by the PowerCAMPUS line of business is Microsoft Sharepoint Portal Server. All non-anonymous access to Sharepoint Portal Server sites and areas requires that the user possess an Active Directory account. An important part of the value added to the Sharepoint Portal Server solution the PowerCAMPUS line of business will offer is the automatic provisioning of Active Directory accounts based on pre-existing IQ.Web accounts. The current design is a user-driven method of account provisioning.
Topics of Discussion

• What is Single Sign-On?
• Current SSO methods used in PowerCAMPUS Portal
• SharePoint infrastructure for SSO
• Caveats
• Q&A
What is Single Sign-On?
What is Single Sign-On?

- After user authenticates to SharePoint, the user is not prompted for id and password when accessing integrated, external systems
- Dictionary.com: Any user authentication system permitting users to access multiple data sources through a single point of entry. Part of an integrated access management framework.
Authentication Primer

- **Native Localized Authentication**
  User Directly Authenticates on the System

- **Trusted Authentication**
  User provides a Ticket to the System issued by a Trusted Authentication Service hosted by another System

- **Proxy Authentication**
  System (Portal) acts as a user’s proxy in authenticating users on other systems.
  - Requires keeping credentials on the proxy system synchronized
  - SharePoint Portal Server uses the Microsoft Single Sign-on Service
SSO Schemes

- **HTTP Scheme**
  - Web part authenticates and communicates to external system over HTTP(S) as a client.
  - No modifications required on external system.
  - Web part custom code must extract (page-scraping) content from web pages to render web part content.

- **Native Scheme**
  - Web part uses supplied API to connect to external system
  - Examples, COM/DCOM libraries, direct database connections, web services, .NET Remoting, etc.
SSO Methods used by the PowerCAMPUS Portal
Current SSO methods used by PowerCAMPUS Portal

- XAPI
  - ISAPI filter that enables single sign-on from PowerCAMPUS Portal to any application that delegates its authentication to IIS.
- Custom SSO from PowerCAMPUS Portal to IQ.Web
  - Each custom Web Part that connects to IQ.Web uses custom classes that facilitate the creation of an IQ.Web session and evaluate users’ roles
XAPI Overview

User logs in to Portal with AD credentials

Active Directory

XAPI checks headers and cookie

Portal

IIS Virtual Server

XAPI checks headers and cookie

Participating application (e.g., OWA)
A lot of text about XAPI

- When a non-domain user* connects to one of the participating web applications, the following steps occur:
  - If any authentication header is present, the caller is already authenticated to the domain; in this case the XAPI filter acts as a pass-through filter without altering the request or response states.
  - If the basic authentication header is present and XAPI cookie is not present, the caller has been authenticated by the hosting application; in this case XAPI sets an XAPI Cookie, which contains the basic authentication header in the response object.
  - If authentication header is not present and XAPI cookie is present, the caller has been authenticated by one of the other participating web applications; in this case XAPI uses the XAPI Cookie contents to construct a basic authentication header.

*A user who is not logged in the domain; this includes a non-IE browser on workstation in the domain.
Portal-to-IQ.Web SSO Overview

1. Web Part checks for existing IQ session for this authenticated user
2. If no IQ session exists, generate one. If a session exists, refresh its timeout
3. Web Part checks for existing IQ session for this authenticated user
4. If the user clicks a link in the Web Part that opens a new browser window, IQ validates the session id before allowing access
SharePoint Portal Server and Single Sign-On
Sharepoint SSO Overview

User Logs in To Portal With AD credentials

Pass Credentials to External Application (EA)

Retrieves EA Credentials

Retrieves EA page, render summary in WebPart
SharePoint SSO Overview (continued)

1. User login into SharePoint using Active Directory credentials.
2. When a user loads a portal page containing an SSO web part, code for this web part is triggered.
   1. Web part code checks whether the user credentials for the external application are stored in the single sign-on database (SSODB). If they are stored, process continues from Step 3.
   2. If user credentials are not stored, the user’s browser is redirected to the logon form for this application (SharePoint page). User supplies credentials for the external application, which are then stored in SSODB. Process continues on Step 3.
3. Web part extracts credentials from SSODB and connects to the external application using HTTP(S).
4. Web part retrieves required information from the external application, processes it, and renders it in SharePoint web part content area.
Prompt for External Password
Saving your credentials for the external application

Provide IQ.Web Account Information

This page is not encrypted for secure communication. User names, passwords and any other information will be sent in clear text. For more information, please contact your administrator.

Use this page to provide the information specified to access the enterprise application.

Logon Information
Type the account information for the enterprise application.

User ID: cseilton
Password: ********

Sign in automatically.

OK  Cancel
SSO Configuration

- To configure single sign-on for the first time, you must complete the following tasks:
  1. Determine and set up necessary Windows accounts.
  2. Enable the single sign-on service on the portal server.
  3. Configure the single sign-on settings.
  4. Create a new application definition.
  5. Provide account information for the application definition.
Configure the SSO settings

Manage Settings for Single Sign-On for MALSHAREPOINT5

Server Settings
Use these links to manage settings for single sign-on.
- Manage server settings
- Manage encryption key

Enterprise Application Definition Settings
Use these links to manage settings for enterprise application definitions.
- Manage settings for enterprise application definitions
- Manage account information for enterprise application definitions
Configure the SSO settings

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Sign-On Settings</td>
<td>In the <strong>Account name</strong> box, type the name of the group or user account that can set up and manage the single sign-on service. Example: DOMAIN\group name or DOMAIN\user name</td>
</tr>
<tr>
<td>Enterprise Application Definition Settings</td>
<td>In the <strong>Account name</strong> box, type the name of the group or user account that can set up and manage enterprise application definitions. Example: powercampus\sharepointservice</td>
</tr>
<tr>
<td>Database Settings</td>
<td>In the <strong>Server name</strong> box, type the name of the database server that stores the settings and account information for single sign-on. Example: computer name or computer name\SQL Server instance</td>
</tr>
<tr>
<td>Time Out Settings</td>
<td>In the <strong>Ticket time out</strong> box, type the number of minutes to wait before allowing a ticket to time out. Example: 2</td>
</tr>
<tr>
<td></td>
<td>In the <strong>Delete audit log records older than</strong> box, type the number of days to hold records in the audit log before deleting. Example: 10</td>
</tr>
</tbody>
</table>

**Account name:**
- powercampus\ssouservice
- powercampus\sharepointservice

**Server name:**
- BENCHMARK
- example: computer name or computer name\SQL Server instance

**Ticket time out (in minutes):**
- 2

**Delete audit log records older than (in days):**
- 10
Configure the SSO settings
Configure the SSO settings
Configure the SSO settings
Configure the SSO settings
Building the Web Part

- Get the Credentials
  - Call the GetCredentials method of the Credentials class. Specify the application name for which you want to retrieve credentials.
  - Example:
    ```csharp
    Credentials.GetCredentials(1, "SampleApp", ref rgGetCredentialData);
    ```
Building the Web Part, continued

- If SSOSrv can’t find credentials for the user for the enterprise application definition, the GetCredentials method throws a SingleSignonException. If the LastErrorCode property of the SingleSignonException is SSO_E_CREDS_NOT_FOUND, call the GetCredentialEntryUrl(String) method of the SingleSignonLocator class to build the URL to the Single sign-on logon form.
Building the Web Part, continued

- When you retrieve the URL for the logon form, redirect the browser to the URL. The logon form is displayed and prompts the user for the account name and password to use with the enterprise application definition. After SSOSrv saves the credentials, the form redirects control back to the original Web Part.
protected override void RenderWebPart(HtmlTextWriter writer) //RenderWebPart
{
    string[] rgGetCredentialData = null;
    try
    {
        //Try to get the credentials for this application.
        //Before running this code, make sure that an individual
        //application definition for application called
        "SampleApp"
        //has been added.
        Credentials.GetCredentials(1,"SampleAPP", ref
rgGetCredentialData);
    }
}
catch (SingleSignonException ssoe)
{
    // This exception will be thrown if this user does not have
    // credentials for the "SampleApp" application.
    if(SSOReturnCodes.SSO_E_CREDS_NOT_FOUND ==
      ssoe.LastErrorCode)
    {
        // Send the user to the single sign-on logon form.
        // The logon form will:
        //  - Prompt the user for credentials for this application
        //  - Save credentials for this user for this application
        //  - Then redirect the user back to this Web Part
        string strSSOLogonFormUrl =
            SingleSignonLocator.GetCredentialEntryUrl
            ("MyIndividualApplicationID");
        writer.Write("<a href=" + strSSOLogonFormUrl +"">Click
        here to save your credentials for the Enterprise
        Application.</a>");
        writer.WriteLine();
    }
}
Caveats

- **Types of SSO**
  - **Authenticating Web Part**: Web part authenticates with external system for rendering contents
  - **Authenticating User’s Browser**: Providing user a link within SharePoint that will direct user to external system website by setting cookie in the user’s browser. This can become complex if external system is not on the same domains.

- **Password Synchronization**: Understanding associated operations overhead of password synchronization between systems.
  - Change password events must be trapped and exchange password between system over a secure medium
  - Mandate that users provide changed password when SSO fails

- **Session Management**
  - **Session Pooling**: Every time web part rendering takes place it will authenticate with external system. Session must be lightweight and closed or pooled for reuse.
  - **Single Sign-Off**: Single Sign-On does not address session co-ordination and Single Sign-Off aspects of system integrations.
Online articles to reference

- **Sharepoint Resource Kit article**
  - [http://www.microsoft.com/technet/prodtechnol/sppt/reskit/c2661881x.mspx](http://www.microsoft.com/technet/prodtechnol/sppt/reskit/c2661881x.mspx)

- **Integrating 3rd-party Single Sign-On in Sharepoint Portal Server**
  - [http://www.theserverside.net/articles/showarticle.tss?id=ImplementingSSO](http://www.theserverside.net/articles/showarticle.tss?id=ImplementingSSO)
Summary

- You can use XAPI for IIS-based applications
- The PowerCAMPUS group intends to document the use of the shared libraries for enabling SSO to IQ.Web for you to use in your custom web parts to IQ.Web
- You can use MS SSO Service for other web applications
  - Get, send, save external application credentials using the SSO database
- The next version of SharePoint Portal Server promises to enhance its enterprise single sign-on features
Questions & Answers
Thank You!

Chad Sexton
Chad.Sexton@sungardhe.com

Please complete the on-line Evaluation Form
Evaluation Code 510