QDABRA DATABASE ACCELERATOR V2.3

Using Single Sign-On with SharePoint Form Libraries in DBXL 2.3

INTRODUCTION – SHAREPOINT FORM LIBRARIES AND DBXL 2.3

DBXL 2.3 introduces a new feature: the ability to post documents to SharePoint form libraries as well as storing them in the internal database. Each document type (configuration) can have zero or more associated SharePoint form libraries, specified by URL. The DBXL 2.3 web service will add documents to these form libraries as it receives them, update the documents in the form libraries as updates are received, and delete documents from the form libraries as they are deleted from the web service.

DBXL 2.3 can access the form libraries via SharePoint Object Model if they reside on the same machine. Or, DBXL 2.3 can use the SharePoint Web Services to access form libraries on any machine. When using the SharePoint Web Services, authentication can be an issue. Accessing a form library on the same machine usually poses no problem. However, Windows security limitations may be encountered accessing a form library on a different machine. The biggest limitation is the “one-hop” problem: Windows authentication provides a mechanism where the user can be impersonated on the server for access to resources on that server, but the server is prevented from accessing another server under the user’s credentials.

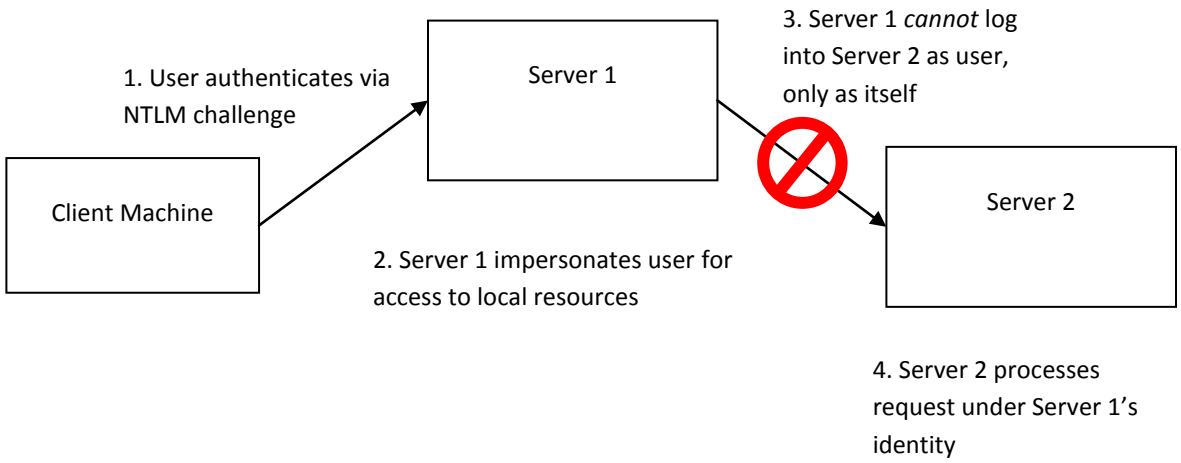
There are a few approaches to work around this. The primary purpose of this document is to explain how DBXL 2.3 can use Single Sign-On (SSO) services to log into the target SharePoint server with credentials configured via SSO. Other approaches discussed briefly include enabling Kerberos constrained delegation or configuring the DBXL 2.3 IIS App-Pool to use a domain user account (rather than a machine-local user account) and granting access permissions to that domain user account on the target SharePoint site.

CONTENTS

SINGLE SIGN-ON THEORY

The Trouble with NTLM

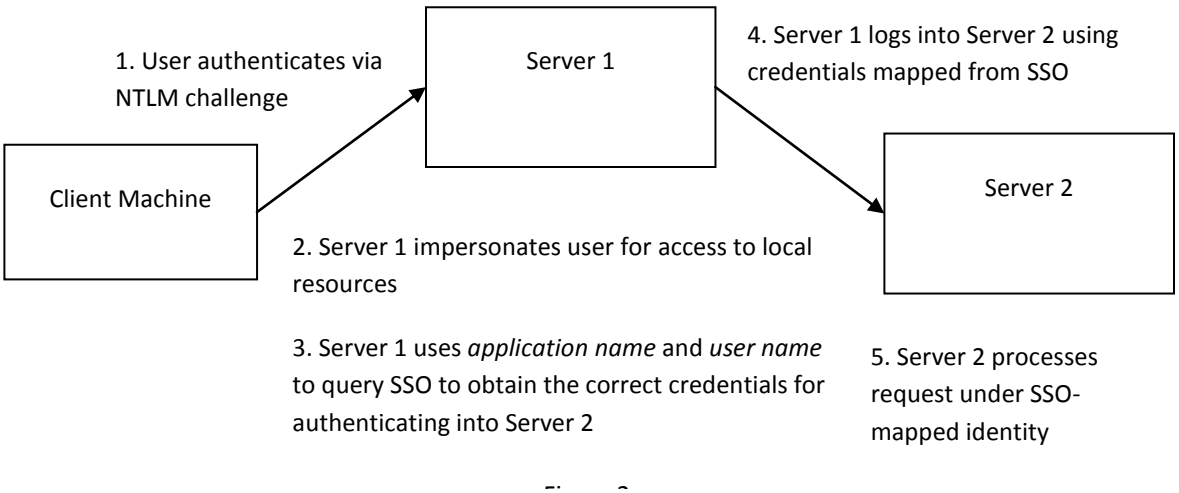
NTLM has a well-known restriction that allows a user's credentials to move only "one hop" on the network. This permits a user to log into a server and access resources local to that server, without the server actually knowing the user's password. But without actually knowing the user's password, the server cannot log into another server and authenticate as the user – it can only authenticate as a server account. This is actually a design point in NTLM which prevents untrusted or compromised server software from having network-wide access to the user's securables. But it is inconvenient.



The first cut at a solution to the problem would be to provide Server 1 with access to the user's password. In general this is not a good solution, since software on Server 1 could then access resource on the available network as the user. If Server 1 is compromised or the software is not trusted, this provides unacceptable access to the network. We shall see that some (though not all) SSO services provide a way to mitigate this. Another problem is that relying on a user's credentials implies that all servers are in the same domain and use Windows authentication. In general there are scenarios where Server 2 may be on an unrelated Windows domain, or even use its own authentication scheme. For example, Microsoft SQL Server can use Windows user accounts and authentication, but also provides its own proprietary management of users and passwords, with its own authentication scheme. Other scenarios arise in Internet or Extranet environments.

How Single Sign-On Solves the Problem

SSO is designed to solve problems in these environments as well. Let's take a look at a typical SSO scenario:



In this scenario, Server 1 uses SSO to obtain credentials to use for logging into Server 2. Examining the steps in detail will demonstrate the various options available in an SSO scenario.

1. The user is authenticated into Server 1 via NTLM challenge. Note that in NTLM challenge authentication, *Server 1 never knows the user's password*. Instead, cryptographic one-way hashes are exchanged between the client, Server 1, and the domain controller that allow trust to be established in both directions without exchanging the password.
2. Software on Server 1 can impersonate the user during access to local resources to enforce permissions with respect to the user.
3. When Server 1 needs to access Server 2 (e.g. via a web service call), it uses SSO to obtain credentials. There are some details to be aware of:
    a. SSO credential lookup is done in the context of an *application*. The ability to create multiple applications allows for the creation of multiple sets of user credential mappings. For example, one application might be created for mapping to a SharePoint server and a second unrelated application might be created for mapping to a SQL server.
    b. Credential mapping can be done on an individual user-to-user basis or be determined by group membership:
        i. For individual users, the user name is used as a key to query a dictionary that returns credentials (typically a user name and password) to be used for logging into the second server. Each individual user account can be mapped to a different user account on the target server. The need to configure per-user credentials mapping makes this more complicated to set up and limits the number of situations it can be used.
        ii. For access that does not require per-user permissions, group membership can be used. In this case, the user is checked for membership in a specific domain group. If the user is a member, then the dictionary is queried for credentials (user name

and password) to be used for logging into the second server. In this case, all users who are members of a domain group will log into the second server using the *same* credentials/account.

4. Server 1 logs into Server 2 using the credentials obtained from SSO. In general, SSO does not mandate any particular authentication scheme, so the protocol could be NTLM, forms authentication, or some other mechanism. For SharePoint web services, it is typically one of the two mechanisms just mentioned.
5. Server 2 can impersonate the user account obtained from the SSO credentials lookup to enforce permissions with respect to the SSO application user or group.

CONFIGURING SHAREPOINT SINGLE SIGN-ON SERVICE

DBXL 2.3 supports two SSO services that are useful and commonly available in SharePoint scenarios. The first is SharePoint Portal SSO, which is available if DBXL is installed on a machine that also has Microsoft Office SharePoint Server (MOSS) 2007 installed on it. MOSS 2007 is required; the SSO service is not available in WSS 3.0 without MOSS. The other alternative is the Host Integration Server (BizTalk) Enterprise SSO service. This may be available to you for Windows Server 2003 R2 under certain licensing agreements.

The SharePoint Portal SSO Service

SharePoint SSO is not the easiest thing to set up. Microsoft has provided extensive documentation, for example http://technet.microsoft.com/en-us/library/cc262932.aspx and http://blogs.msdn.com/sharepointdesigner/archive/2007/08/27/an-introduction-to-single-sign-on-sso-with-data-views.aspx. I have summarized some of the common issues below to try to help make it easier for DBXL scenarios. This will configure your SharePoint/DBXL server to act as an SSO server, to help you get started. For more advanced configuration options (such as working with a server farm), please consult the preceding Microsoft links for more details.

Note that in order to use SharePoint Portal SSO, MOSS 2007 *must* be installed on the same machine as DBXL 2.3, since DBXL will be calling into the SSO service via the Microsoft.SharePoint.Portal.dll assembly.

1. Configure and start the Microsoft Single Sign-On service
    a. Create a user account in the domain to be used for this service. (A local account on the machine will not work.)
        i. Make the account a member of Local Administrators on the machine
        ii. Ensure the account is a Farm Administrator in SharePoint. For a typical "basic" install of MOSS 2007 this is usually already the case for a Local Administrator.
        iii. Ensure the account has Security Administrators and db_creator roles on the SQL server where the SSO database will be stored.
    b. Set this user's credentials as the identity for the Microsoft Single Sign-On service in the Services administration tool for the local computer.
    c. Configure the service to *Automatic* start and ensure that it is started.
2. Configure SharePoint Central Admin SSO settings

a. Navigate to the SharePoint Central Administration web site for the machine (typically found in the Start menu under *Administrative Tools* → *SharePoint 3.0 Central Administration*)
b. Log in as the user created in 1.a by clicking the username in the upper-right corner of the page and choosing *Sign in as a Different User*
c. Select the *Operations* page, and then under the *Security Configuration* section, click the *Manage settings for single sign-on* link.
d. Click *Manage server settings*
    i. Specify the user account from 1.a. as both the *Single Sign-On Administrator Account* and the *Enterprise Application Definition Administrator Account*.
        1. In a production scenario, it is more secure to create an SSO group and create separate accounts for these functions. See http://technet.microsoft.com/en-us/library/cc262932.aspx#Section2 for specific details.
    ii. Ensure the location for the SSO database is valid. With SharePoint on the box, it usually defaults to *<servername>\OfficeServers* which is a good option. However, note that SSO does not support SQL Express, so make sure you are pointing it to a SQL 2005 or 2008 instance (or similar).
e. If you attempt to configure SSO without sufficient database permissions, it may be necessary to stop and restart the SSO service and possibly log off and log back onto the machine before attempting to configure the SSO settings in SharePoint Central Administration again. (for example, see http://social.technet.microsoft.com/Forums/en-US/sharepointadmin/thread/d36b60b3-d17f-429f-8f27-ca3504d2cee7/)

The Host Integration Server (BizTalk) Enterprise SSO Service

If MOSS 2007 is not available on the DBXL server machine and it is running Windows Server 2003 R2, the Host Integration Server SSO service provides an alternative. The Microsoft home page for Host Integration Server is located here: http://www.microsoft.com/hiserver/default.mspx.

1. Download and install the HIS Enterprise SSO service.
    a. Be sure to elect to install *Enterprise Single Sign-On* under the *Server* category in the installer (it is not enabled by default).
2. In the resulting configuration dialog, after the install finishes, the service account user must be specified
    a. Create a user that is *not* administrator (unlike in the SharePoint SSO case). This can be a local machine user. However, if you will be using group mappings, the user must have permissions to query Active Directory for group membership.
    b. Enter this user name and password in the HIS configuration dialog
3. Finish configuration with remaining default options.

CREATING AN APPLICATION MAPPING

Application Mapping in SharePoint Portal SSO Service

The SSO Application encapsulates the mapping from one set of front-end users to one set of back-end users. In general, use a separate SSO Application for each single scenario (e.g. for each SharePoint list mapping in DBXL that requires SSO).

1. Creating the SSO Application
    a. Navigate to the SharePoint Central Administration web site for the machine (typically found in the Start menu under *Administrative Tools* → *SharePoint 3.0 Central Administration*)
    b. Log in as the user created in 1.a by clicking the username in the upper-right corner of the page and choosing *Sign in as a Different User*
    c. Select the *Operations* page, and then under the *Security Configuration* section, click the *Manage settings for single sign-on* link.
    d. Click *Manage settings for enterprise application definitions*
    e. To create a new application, click the *New* button
        i. The *Application name* will be the name you specify in SSO configuration options. The *Display name* and *Contact e-mail address* can be whatever you want.
        ii. To map individual front-end users to individual back-end users, select *Individual* for *Account type*. Or, to map all front-end users to one shared back-end user, select *Group* for *Account type*. If possible, check *Windows authentication* for increased security. In the scenario of accessing SharePoint servers via DBXL, this is desirable.
        iii. The remaining options should be left as the defaults.
        iv. Click *OK* to create the application
    f. To edit an existing SSO Application, click on the application in the list. Some properties cannot be changed.
2. Specifying the user mapping
    a. Navigate to the SharePoint Central Administration web site for the machine (typically found in the Start menu under *Administrative Tools* → *SharePoint 3.0 Central Administration*)
    b. Log in as the user created in 1.a by clicking the username in the upper-right corner of the page and choosing *Sign in as a Different User*
    c. Select the *Operations* page, and then under the *Security Configuration* section, click the *Manage settings for single sign-on* link.
    d. Click *Manage account information for enterprise application definitions*
    e. Select the SSO Application you wish to configure from the drop-down.
    f. If the *Account type* is *Group*:
        i. All front-end users being granted access must be a member of an Active.Directory group. Enter the group name in the *Group account name* box. (To get started in a dev environment, you can enter your *Domain Users* group – but this is not recommended in a production deployment; create a new group specific to the scenario.)
        ii. Click *Set.*
        iii. Enter the user name and password of the back-end user that everyone will get mapped to.

iv. Click *OK.*
v. To make any changes to an existing mapping, follow the same steps, providing the new user identity and credentials at each step.

g. If the *Account type* is *Individual*:
i. For security reasons, SharePoint does not list configured users.
ii. To add a new user mapping (or modify an existing user mapping):
1. Enter the user's name in the *User account name* box.
2. Ensure *Update account information* is selected.
3. Click *Set.*
4. Enter the user name and password of the back-end user that everyone will get mapped to.
5. Click *OK.*
iii. To remove a user mapping:
1. Enter the user name in the *User account name* box.
2. You can elect to remove the user from the selected SSO Application or from all SSO Applications in the system by clicking the appropriate radio button.
3. Click *Set.*

Application Mapping in Host Integration Server (BizTalk) Enterprise SSO Service

1. Creating the SSO Application
a. Open the SSO Administration tool (Programs → Enterprise Single Sign-On → SSO Administration)
b. Click on the *Affiliate Applications* node.
c. Right-click and choose *Create Application.*
d. To map individual front-end users to individual back-end users, select *Individual* for *Application type*. Or, to map all front-end users to one shared back-end user, select *Group* for *Application type*.
e. Enter the *Application name* that will be used to configure access in DBXL.
f. Add one or more *Application administrators* who will have permissions to change mappings.
g. Under *Application* users, add one or more individual front-end users or domain groups that will be granted mappings to the back-end service.
h. Click *Next.*
i. Leave the configuration options set to the defaults and click *Next.*
j. Leave the field mappings as default. If possible, check *Credentials are Windows credentials* for increased security. In the scenario of accessing SharePoint servers via DBXL, this is desirable.
k. Click *Create* and then *Finish.*
l. Set the credentials by right-clicking on the newly created mapping and selecting *Set Credentials.* (For *Individual* application types, individual users can use the *SSO Client Utility* to set their own credentials, so the administrator does not need to have access to each user's credentials.)

2. Specifying the user mappings
    a. To create a mapping for either *Individual* or *Group* application types:
        i. Select the Application to configure under the *Affiliate Applications* node.
        ii. Right-click and select *New Mapping.*
        iii. Enter the front-end user name for *Windows user* and the back-end user to map to for *External user.*
        iv. Click *Ok.*
        v. Multiple mappings can be entered for the *Individual* case.

USING THE MAPPING IN DBXL 2.3 TO ACCESS A SHAREPOINT FORM LIBRARY

I will assume that you are familiar with DBXL and have configured a document type with an InfoPath form template. To connect the document type to a SharePoint form library, so that documents submitted to the DBXL document type will also be published to the form library, follow these steps. Note: if the SharePoint form library is on the same server as DBXL, you typically do not need to use SSO. It is required in cases where the SharePoint form library is on a different server than the one DBXL is running on.

1. Open DBXL Administration Tool (DAT).
2. Edit the document type (configuration) that you wish to add the SharePoint form library mapping to.
3. Click on the *SharePoint* tab.
4. Either locate the existing SharePoint mapping you wish to configure SSO for, or create a new one, as follows:
    a. Click *Insert mapping.*
    b. Enter the URL of the SharePoint form library you wish to create. (e.g. http://<yourservername>/<formlibraryname>/Forms/AllItems.aspx. Note that the form library name you will specify in the URL should be a non-existing library.
    c. Choose *Web Service* from the *Method* dropdown.
    d. Select the SSO provider you wish to use (SharePoint Portal or Enterprise SSO).
    e. Enter the SSO Application Name that you configured earlier.
    f. Check the Publish Library and XSN checkbox.
    g. Save the configuration.
5. Verify that the mapping is working by switching to the *Documents* tab, creating a new document, submitting it, and seeing if it is published to the new SharePoint form library by the correct user. If you need more information on how to create SharePoint Mappings using Web Service, see the document How to Set up a SharePoint Mapping using Web Service.

ALTERNATIVES

There may be times that SSO is not the best solution to managing cross-server identity problems. In this case, there are a couple of other options available.

Kerberos

If it is possible to enable Kerberos constrained delegation, it can be used to achieve the same effect as SSO. In particular, constrained delegation allows a specific service on the front-end machine to authenticate with the user's credentials (delegation) to a specific list (constrained) of services on other servers. This means no user mapping needs to be configured; the front-end user will be authenticated onto the second back-end service.

The difficulty with Kerberos constrained delegation is that some data centers are not able to use it because of the complexity of configuration and the increased scope for security breaches. The security risk is that if the front-end service is compromised (such as by the existence of a data-execution bug), the malicious code has access to all of the targets that are permitted by the constrained delegation white list. By default under NTFS, a compromised service can only damage the local machine.

Configuring the Qdabra IIS App-Pool Service Account Identity

Another option is to configure the IIS Application Pool Service Account identity. By default it is NT AUTHORITY\NETWORK SERVICE, which is a local account that does not have access to domain resources. It is possible to use a domain account as the service account identity, allowing the service to authenticate as itself into other servers. In this case, the service account can be granted access to back-end resources. This would be similar to using an SSO group mapping application.

Like in the case of constrained delegation, there is an increase in security risk. The service's domain account has access to network resources, so if the service is compromised, the damage it can do extends beyond the local machine to those network resources.