Securing SAS 9.2 Web Applications with IBM Tivoli Access Manager WebSEAL

SAS 9.2 Web applications can run on IBM WebSphere Application Server. You can also integrate WebSphere Application Server with external security manager products, including IBM Tivoli Access Manager. One of the components of Tivoli Access Manager is the WebSEAL reverse Web proxy security server. WebSEAL acts as a front end to Web applications and provides a single point for credential challenges.

Configuration of WebSEAL is complex and requires a thorough understanding of security and the network topology that the security solution will protect. Developing a security solution requires focused attention during installation and configuration. Multiple configuration and topology options are available when you install these tools.

This document focuses on one topology. In this sample topology, you configure WebSphere Application Server to trust authentication that WebSEAL provides. WebSEAL and WebSphere Application Server share the same user registry, an LDAP server. Here is the communication flow:

1. A user at a Web browser sends an HTTP request for a dynamic page to the WebSEAL server. The WebSEAL server protects the HTTP server and WebSphere Application Server. WebSEAL challenges the user for credentials.
2. After authentication, WebSEAL passes the request directly to WebSphere Application Server or to the HTTP server, which then sends the dynamic page request to WebSphere Application Server.
3. Finally, WebSphere Application Server generates a response by executing a dynamic page in one of the SAS Web applications.

This document describes configuring WebSEAL to provide single sign-on to WebSphere Application Server using a Trust Association Interceptor (TAI) or through Lightweight Third Party Authentication (LTPA) token.

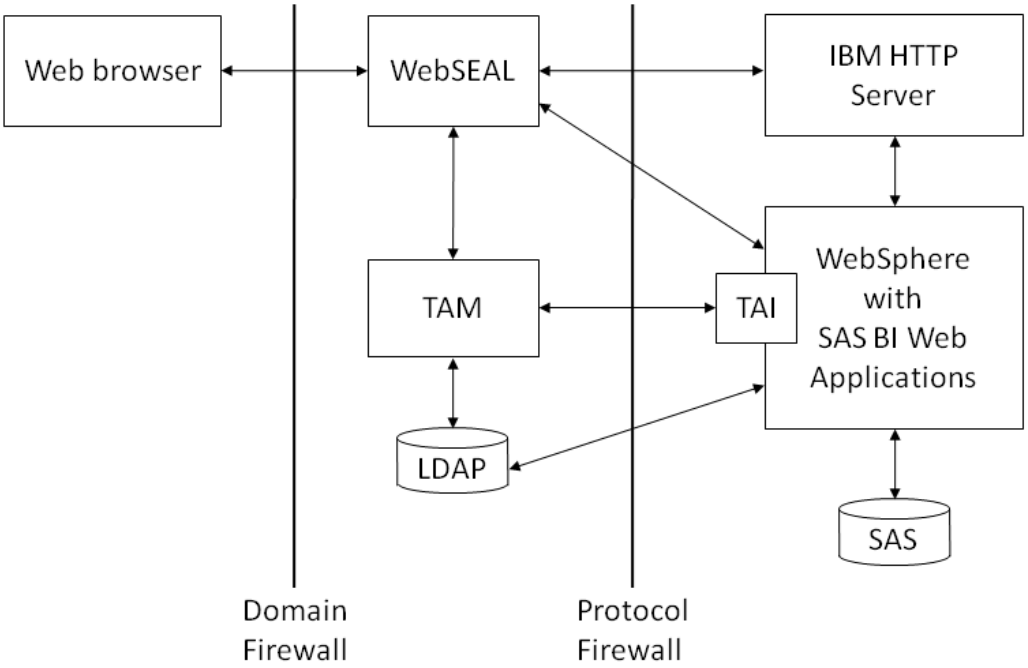
Audience

This document is intended for SAS administrators, Web application server administrators, and WebSEAL administrators. Experience in those areas is necessary for successful deployment.

Sample Topology

The following figure shows the sample topology. WebSEAL, Tivoli Access Manager, and Tivoli Directory Server are deployed on one machine and the middle tier components (HTTP server and WebSphere Application Server), and SAS servers are on another machine. You create a WebSEAL *junction*--a connection mechanism between WebSEAL and other components (explained later in more detail in this document)—to WebSphere Application Server. However, in other topologies you can create the junction to the HTTP server.

Figure 1 WebSEAL and WebSphere Application Server Topology



Required Software Products

To implement the sample topology, you must install and configure several IBM products. This document assumes that you know how to install and configure SAS Web applications and have access to all required IBM products. The sample topology requires these products:

- IBM Tivoli Access Manager for e-business V6.x (Runtime, Policy Server, Policy Proxy Server, Authorization Server, WebSEAL, Web Portal Manager, and Java Runtime Environment)
- IBM WebSphere Application Server V6.1.0.17 or higher
- IBM Tivoli Directory Server V6.x
- IBM DB2 V8.1 (a Tivoli Directory Server dependency)

Install and configure DB2, LDAP, WebSEAL, WebSphere Application Server, and IBM HTTP Server.

Configure Users in LDAP

Tivoli Directory Server is an LDAP server that works with Tivoli Access Manager. You must configure users in the LDAP server so that WebSEAL and WebSphere Application Server can locate them. For information about configuring Tivoli Directory Server or adding users, see the Tivoli Directory Server documentation. The sample in this document assumes that LDAP is created with an LDAP suffix `dc=sas,dc=com`. The example below shows how the LDAP directory in this document is configured. Although a few user accounts are shown, you can add any number of users. Assign the proper password for each user.

```
dn: dc=sas,dc=com
objectclass: top
objectclass: dcObject
objectclass: organization
o: sas
dc: sas

dn: ou=people,dc=sas,dc=com
objectClass: organizationalUnit
ou: people

dn: uid=sasguest,ou=people,dc=sas,dc=com
objectclass: inetorgperson
uid: sasguest
userPassword: xxxxxxxx
cn: SAS Guest
sn: Guest

dn: uid=sasdemo,ou=people,dc=sas,dc=com
objectclass: inetorgperson
uid: sasdemo
userPassword: xxxxxxxx
cn: SAS Demo User
sn: Demo User
```

Configure WebSphere for a User Registry

You must configure WebSphere Application Server to use the LDAP server. For information, see “Configuring the user registry” in *WebSphere Application Server V6.1 Security Handbook* or *Application Server V7.0 Security Guide. WebSphere*.

After reconfiguring WebSphere Application Server, stop, synchronize, and restart the server. It is secured with the user registry after restart. As a test, access a Web application on the server, such as snoop, by opening a Web browser to `http://HOSTNAME:9080/snoop`. If you configured WebSphere Application Server correctly, WebSphere Application Server asks you for credentials that are stored in the LDAP server.

Overview of WebSEAL Junctions and Junction Mapping Tables

This section introduces the concepts for WebSEAL junctions. This information is needed before the next section, which describes how to configure WebSphere Application Server with a WebSEAL junction. A WebSEAL *junction* is a connection mechanism between the WebSEAL server and WebSphere Application Server. It provides protective service on behalf of WebSphere Application Server. WebSEAL authenticates the user and passes encrypted user information to WebSphere

Application Server. WebSEAL junctions can be categorized based on the transport protocol that is used (HTTP or HTTPS) or how user information is passed. A TCP junction uses HTTP (TCP) protocol between WebSEAL and WebSphere Application Server. An SSL junction uses the HTTPS (SSL) protocol. TCP and SSL junctions typically pass authenticated user information through a special HTTP header field, `iv-user`. To extract user information from the special HTTP header that is sent to WebSphere Application Server, WebSphere Application Server supplies a WebSEAL Trust Association Interceptor (TAI) that you must configure to support a TCP or SSL junction in WebSphere Application Server.

As an alternative, WebSEAL and WebSphere Application Server also support a special token called Lightweight Third-Party Authentication (LTPA) token. An LTPA token is used for cross-domain single sign-on. An LTPA junction can be created with the shared LTPA key between WebSEAL and WebSphere Application Server. The LTPA key can be generated from WebSphere Application Server and then exported to WebSEAL for creating the LTPA junction. With an LTPA junction, WebSEAL generates the LTPA token after successful authentication. The token contains user information in a heavily encrypted form. Because WebSphere Application Server has the shared LTPA key, it decrypts the token and uses user information so that SAS Web applications can process Java Authentication and Authorization Service (JAAS). An LTPA junction does not require configuring a TAI because it is based on the shared LTPA key.

For environments that use a TAI, when WebSEAL receives the response page from WebSphere Application Server, it parses the page and adds the junction name to server-relative URLs whenever possible. However, in such cases as functions that are written in JavaScript, WebSEAL might not be able to add the junction name to server-relative URLs. While those functions execute, the generated URL does not contain the junction and, as a result, the URL is not valid. To accommodate this situation, WebSEAL supports the junction-mapping table. The table contains entries that consist of a junction name and a target Web application resource name. When access to a server-relative URL-based Web application resource fails, WebSEAL looks up the resource name in the table to determine the junction name. If the resource is found in the table, WebSEAL adds the junction name to the server-relative URL and tries again. The request that uses the modified URL then succeeds. More often than not TCP, SSL, and LTPA require use of a junction-mapping table to address the potential situation of dynamically generated URLs from JavaScript functions.

WebSEAL also offers a transparent path junction. This junction has a special attribute that you create with the `-x` option. In this case, the junction name matches the name of the SAS Web application. With this junction, URLs do not include the extra WebSEAL junction name context in the request URL. For this reason, the URLs for the SAS Web application do not require an entry in a junction-mapping table. The only disadvantage in using transparent path junctions is that you must create a separate junction for each SAS Web application.

Configure a Trust Association Interceptor in WebSphere

As mentioned in the previous section, you must configure a TAI for TCP and SSL junctions. An LTPA junction does not require that you configure a TAI because it passes user information through the LTPA token. For TCP and SSL junctions, the TAI in WebSphere Application Server receives the request from WebSEAL and decodes the special header, `iv-user`. The TAI then provides the user credentials—those that WebSEAL authenticated—to the WebSphere Application Server JAAS stack to create Java Principal objects in the Subject object. SAS Web applications use the Subject and Principals.

In this section are sample values for configuring a WebSEAL TAI for a TCP junction. If your site requires an SSL junction, you must configure SSL between WebSphere Application Server and the WebSEAL server. For more information about doing this, see *WebSphere Application Server V7.0 Security Guide*.

Note: This interceptor module works for WebSphere Application Server 6.1 and below. Support of this module has been dropped from WebSphere Application Server V7.0. It has been replaced by `com.ibm.ws.security.web.TAMTrustAssociationInterceptorPlus.`

For WebSphere 6.1, you must enable Trust association in the WebSphere Integrated Solutions Console (known as the *administrative console)*. Configure the `com.ibm.ws.security.web.WebSealTrustAssociationInterceptor` with these new properties:

Name: `com.ibm.websphere.security.trustassociation.types`
Value: `WebSEAL`

Name: `com.ibm.websphere.security.webseal.id`
Value: `iv-user`

Name: `com.ibm.websphere.security.webseal.hostnames`
Value: *`yourWebSEALHostname`*

Name: `com.ibm.websphere.security.webseal.ports`
Value: `80`

For WebSphere 7.0, configure the com.ibm.ws.security.web.TAMTrustAssociationInterceptorPlus with these new properties:

Name: `com.ibm.websphere.security.webseal.loginid`
Value: `<valid-userid>`

Name: `com.ibm.websphere.security.webseal.id`
Value: `iv-user`

Name: `com.ibm.websphere.security.webseal.hostnames`
Value: `<WebSEAL-Hostname>`

Name: `com.ibm.websphere.security.webseal.ports`
Value: `80`

These properties identify the location of your WebSEAL server and the special header to use for user information. The WebSphere Application Server TAI intercepts incoming requests and provides user information to WebSphere Application Server. In case of an SSL junction, the symmetric encryption algorithm that the SSL handshake arranges encrypts communication between the WebSEAL server and WebSphere Application Server.

Configure the WebSEAL TCP Junction

Tivoli Access Manager provides the Policy Director Administration (`pdadmin`) tool. With this tool you can manage Policy Director, the WebSEAL server, users in Tivoli Directory Server, and junctions. After you start the `pdadmin` tool, create a TCP junction that contains the correct user account and password for the trusted association with WebSphere Application Server, as shown in this example:

```
server task webseald-yourwebsealhostname create –t tcp –c iv-user -B
–U “valid-user” -W “password” –h washostname –p 9080 -f /tjc
```

In the example, “valid-user” is the user account that secures the connection between WebSEAL and WebSphere Application Server. Port 9080 is the HTTP port for WebSphere Application Server. The `/tjc` value is the name of the junction that identifies the location of WebSphere Application Server. It is this junction name that you should add to the URL for SAS Web applications. For example, the Web application snoop that is deployed on WebSphere Application Server is accessed through the junction, as shown in this example:

```
http://WebSEAL-Hostname/tjc/snoop
```

WebSEAL issues an authentication challenge for credentials and once authenticated, it creates the special HTTP header, `iv-user`, which contains user information. WebSEAL then passes the HTTP request to WebSphere Application Server. WebSphere Application Server receives user credentials through the TAI and returns output from snoop with no additional challenge.

Notes about WebSEAL LTPA junction and Single Sign-On (SSO)

As previously described, a WebSEAL LTPA junction is a special junction that creates the LTPA token to pass authenticated user information to WebSphere Application Server. The LTPA token that the LTPA junction creates is heavily encrypted and uses the shared LTPA key that is generated in WebSphere Application Server and then exported to WebSEAL. The LTPA token is used for cross-domain single sign-on in IBM environments that include WebSEAL and WebSphere Application Server. Because the WebSphere Application Server JAAS system login has the module that supports an LTPA token, you need not configure a Trust Association Interceptor (TAI) in WebSphere Application Server. For more information about configuring LTPA, see “Configuration for the LTPA approach” in *Application Server V7.0 Security Guide. WebSphere.* Transfer the LTPA key file to the WebSEAL machine and create the LTPA junction with it (`-A` option):

```
server task webseald-yourwebsealhostname create –t tcp –c iv-user -B
-U “valid-user” -W “password” -h washostname -p 9080  -f /ljc -A
 -F “LTPA_key_file_location” -Z “keyfile_password”
```

Web applications, such as snoop, on WebSphere Application Server can then be accessed through the LTPA junction, as shown here:

```
http://WebSEAL-Hostname/ljc/snoop
```

In this configuration, WebSEAL issues an authentication challenge for credentials. Once the user is authenticated, WebSEAL builds the LTPA token and passes it to WebSphere Application Server. WebSphere Application Server decodes the LTPA token, reads user credentials from the token, and returns output from snoop with no additional challenge.

Use of a Junction-Mapping Table

SAS 9.2 Web applications can call other SAS Web applications, or JavaScript functions can contain server-relative URLs. As mentioned previously, generated URLs are unlikely to include the junction name and, as a result, they are not likely to be valid. WebSEAL provides a mechanism to detect URL failures and modify a URL with a junction name from a junction-mapping table.

The location of a junction-mapping table for WebSEAL is defined in the WebSEAL configuration file. Look for the location that is identified by `jmt-map=` in the `[junction]` stanza. The typical location is *`TAM_HOME`*`\PDWeb\`*`WebSEAL_server_name`*`\lib\jmt.conf`. For SAS 9.2, all Web applications that customers see must be defined in the table. The example below shows table entries for the previously mentioned WebSEAL `/tjc` junction and SAS Web applications that are deployed in WebSphere Application Server:

```
/tjc  */SASBIDashboard/*
/tjc  */SASWebDoc/*
/tjc  */SASPortal/*
/tjc  */SASLogon/*
/tjc  */SASStoredProcess/*
/tjc  */SASWebOLAPViewer/*
/tjc  */SASWebReportStudio/*
/tjc  */sasweb/*     (used by SAS/ GRAPH applets)
/tjc  */SASTheme_default/*
/tjc  */SASPackageViewer/*
/tjc  */SASBIWS/*
/tjc  */SASPreferences/*
/tjc  */SASAdmin/*
/tjc  */SASSharedApps/*
/tjc  */SASFlexThemes/*
```

Notes about Using Junction Cookies

When a WebSEAL junction is created with the junction cookie option (`-j`), WebSEAL modifies the Web page that is returned to a client by inserting code in a JavaScript or AJAX code block that contains the junction information. The inserted code sets a junction-identifying cookie in the Web browser. The junction cookie can be used to add a junction name to dynamically generated server-relative URLs for JavaScript and AJAX calls, but it cannot handle the external links that lack a WebSEAL junction name. SAS is aware of known issues that arise when WebSEAL inserts the junction information in an AJAX code block. Some SAS Web applications use AJAX and the code that WebSEAL inserts causes a syntax error during HTTP requests. SAS has also noticed that output display behavior is not consistent among Web browsers that SAS 9.2 supports when a junction cookie is used. For the initial release of SAS 9.2, SAS Web applications do not support using junction cookies. Support of junction cookies will be considered for a future SAS 9.2 maintenance release.

Notes about WebSEAL Transparent Path Junctions

A WebSEAL transparent path junction is a special junction because the name of the junction matches the name of the SAS Web application. The advantage of using transparent path junctions instead of a standard junction is that they avoid the need to include the junction name in the URL. Therefore, a junction-mapping table is not needed to resolve server-relative URLs. The disadvantage is that you must create a junction for each SAS Web application. You create the transparent path junctions with the -x option, as shown in these examples:

```
 server task webseald-yourwebsealhostname create -t tcp -c iv-user  -B
-U "user" -W "password" -h washostname -p 9080  -f /SASPortal -x

server task webseald-yourwebsealhostname create -t tcp -c iv-user  -B
-U "user" -W "password" -h washostname -p 9080  -f /SASLogon -x

server task webseald-yourwebsealhostname create -t tcp -c iv-user  -B
-U "user" -W "password" -h washostname -p 9080  -f /SASTheme_default -x
```

The transparent path junction for `SASTheme_default` is a special case and is required when SAS Web applications are accessed through a transparent path junction. With a transparent path junction, SAS Information Delivery Portal is accessed through a URL, as in this example:

```
http://WebSEAL-Hostname/SASPortal/
```

Configure WebSphere Application Server for Web Authentication

By default, SAS Deployment Wizard configures SAS software so that SAS Web applications authenticate to the SAS Metadata Server. The default configuration for the metadata server is to authenticate to the operating system. Because WebSEAL is performing authentication, you must reconfigure WebSphere Application Server and SAS Web applications to trust the authentication that WebSEAL performs. For more information about reconfiguring Web authentication, see "Configuring IBM WebSphere Application Server 6.1 for Web Authentication with SAS 9.2 Web Applications" or "Configuring IBM WebSphere Application Server V7.0 for Web Authentication with SAS 9.2 Web Applications".

Configure Metadata for SAS Web Applications

Once Web authentication is configured, you can access SAS Web applications through WebSEAL and its junction to WebSphere Application Server. Internally, SAS Web applications maintain connection information (also called *service name*) in the SAS Metadata Server for resource access and inter-application communication. Initially, connections are based on the application server host and port number. Because WebSEAL is configured to protect WebSphere Application Server and SAS Web applications are accessed through WebSEAL at this point, you must change connection information for the external SAS Web applications (those that customers see). Change connection information for each application that is identified in the junction-mapping table from section "Use of a Junction-Mapping Table" except for SASTheme_default. You do not need to change the URL information for SASTheme_default because SAS Web applications that query metadata for connection information can access the SAS Themes without needing to go through WebSEAL. For other applications, change connection properties to a URL that includes the WebSEAL host and also a junction, if a standard junction was used.

To change the connection access point from WebSphere Application Server to the WebSEAL server, follow these steps in SAS Management Console:

1. Select **Application Management** > **Configuration Manager**.

2. Right-click on the SAS Web application that you want to reconfigure, and select **Properties**.
3. Click **Connection**, modify the connection parameters, and click **OK**.

For SAS Logon Manager 9.2, you might prefer to reconfigure the default logon target from `/SASLogon` (which have no user interface) to an application such as SAS Information Delivery Portal.

SAS Web Report Studio 4.2 or 4.3 Specific Update

By default, SAS Web Report Studio 4.2/4.3 uses a special redirection filter. When you use it with WebSEAL, you must disable this filter. To do this, follow these steps in SAS Management Console:

1. Select **Application Management** > **Configuration Manager**.
2. Right-click **Web Report Studio 4.2/4.3**, and select **Properties**.
3. Click **Advanced** and then click **Add**.
4. Enter a Property Name of `App.RedirectionFilterDisabled` and a Value of `true`.
5. Restart SAS Web Report Studio 4.2 from the administrative console.

Recommended Reading

The following URLs are current as of May 2010.

- IBM Corporation, 2008. *IBM Tivoli Access Manager for e-business V6.0*. ibm/Redbooks. Available at http://www.redbooks.ibm.com/abstracts/sg247207.html?Open.
- IBM Corporation, 2009. WebSphere Application Server V7.0 Security Guide. ibm.com/Redbooks. Available at http://www.redbooks.ibm.com/redbooks/pdfs/sg247660.pdf