SSO con SAML en Weblogic 9.2

Índice

1 Introducción y esquema general..... 2
2 Paso 1. Crear los dominios para los sitios SAML..... 3
3 Paso 2. Crear usuarios.....4
4 Paso 3. Desplegar aplicaciones.....4
5 Paso 4. Generar y registrar los certificados SSL.....4
6 Paso 5. Configurar domainA como sitio fuente SAML.....5
7 Punto 6: Configurar las propiedades de la relying party.....8
8 Configurar SAML en el source site..... 10
9 Paso 8: Configurar domainB como SAML destination site.....11
10 Configurar las propiedades de la asserting party.....13
11 Configurar el destination site SAML (1.1).....14
12 Testear SSO.....15

1. Introducción y esquema general

La ventaja de utilizar un servidor de aplicaciónes como BEA Weblogic es que implementar un mecanismo SSO es más un tema de configuración que de programación y diseño de asertos. Actualmente es posible establecer SSO entre diferentes dominios. En este ejercicio mostraremos los pasos a seguir para configurarlos adecuadamente desde la consola de administrador.

Concretamente, en este ejercicio mostramos como dos aplicaciones *appA* desplegada en un sitio local y otra *appB* desplegada en un sitio remoto (de otro dominio) implementarán un esquema SSO.

El *sitio fuente* (donde *appA* se autentifica) proporciona el servicio de autentificación y pasa los detalles de autentificación, a través del *Inter-site Transfer Service* (ITS), cuando se lo pide el *destination site*. El sitio fuente incluye un servlet ITS que proporciona funcionalidad SAML como la generación de artefactos y la redirección del usuario al sitio de destino.

El proceso general es el siguiente:

1. El browser del usuario accede a la aplicación *appA* desplegada en el dominio *domainA*.

2. La aplicación *appA* pide y pasa las credenciales al proveedor de autentificación (ASP).

3. Si la autentificación tiene éxito entonces se establece una sesión de autentificación y se muestra la página de bienvenida de *appA*.

4. Desde la página de bienvenida, el usuario clickea en un link para acceder una página segura, la de *appB* desplegada en *domainB*. Esto dispara el servlet ITS.

5. El ITS llama al *SAML Credential Mapper* para pedir un aserto de llamada. Éste le devuelve el aserto. También le devuelve la URL de la página de destino y un path para un formulario POST adecuado (si el sitio fuente está configurado para usar el profile POST).

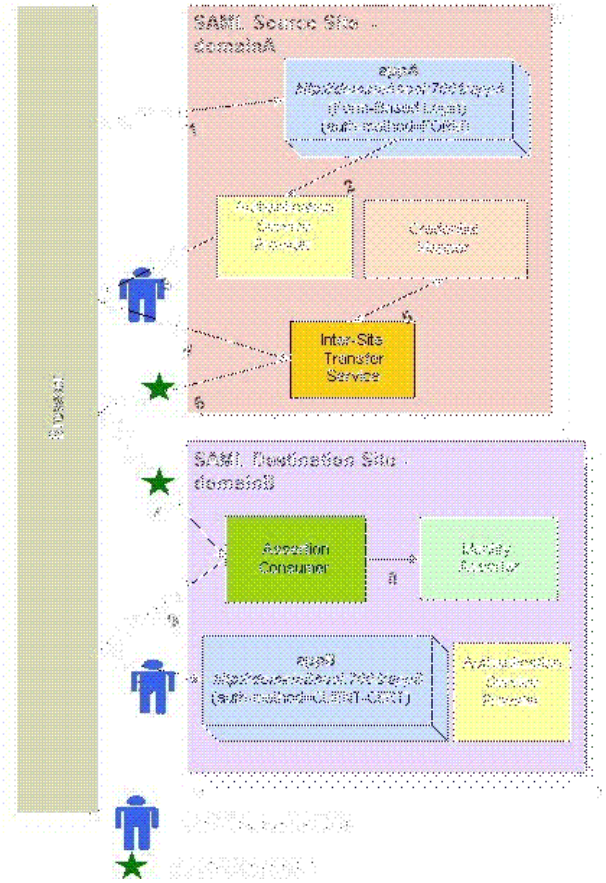
6. El ITS genera una respuesta SAML conteniendo el aserto generado, lo firma, lo codifica en BASE64, lo embebe en el formulario HTML y devuelve el formulario al browser.

7. El browser hace un POST del formulario en el ACS (*Assertion Consumer Service*) del sitio de destino.

8. El aserto es validado.

9. Si la validación tiene éxito, el usuario es redirigido al objetivo, esto es a la página web de *appB*.

10. El usuario es logeado en el sitio de destino sin necesidad de re-autentificarse frente a la nueva aplicación.



Proceso general SSO en Weblogic

Todo este proceso se puede realizar sin necesidad de codificación alguna. Casi todo puede hacerse desde la consola de administración.

2. Paso 1. Crear los dominios para los sitios SAML

La configuración de cada dominio es la siguiente:

domainA. Host:localhost, Server: AdminServer, Aplicación: *appA*, Puerto: 7001, Puerto SSL: 7002.

domainB. Host: localhost, Server: AdminServer, Aplicación: *appB*, Puerto: 7003, Puerto

SSL: 7004.

3. Paso 2. Crear usuarios

Los usuarios quedan como sigue:

domainA. Realm: myrealm, Usuario: ssouser, Password: demosaml.

domainB. Realm: myrealm, Usuario: ssouser, Password: demosaml.

4. Paso 3. Desplegar aplicaciones

La *appA* está configurada como una FORM-based authentication. Es una página JSP llamada `auth.jsp` en la carpeta *admin* necesita que el usuario autentificado tenga el rol *admin*. Este rol es mapeado al principal *ssouser* en `weblogic.xml`. La configuración declarativa se muestra en `web.xml`.

Cuando el browser intente acceder a `/admin/auth.jsp` entonces se mostrará la página `login.jsp` pidiendo las credenciales. Entonces se autentificará a *ssouser*. Si la autentificación tiene éxito entonces se mostrará la página `auth.jsp.`

La *appB* está configurada como CLIENT-CERT por lo que usará asertos de identidad para autentificación. Esta aplicación + se desplegará en el *domainB*. Un JSP llamado `services.jsp` y situado en la carpeta `/admin`, requiere que el usario autentificado tenga el rol *admin*. Cuando el SSO esté implementado se podrá acceder a este JSP sin que se le vuelvan a pedir las credenciales.

5. Paso 4. Generar y registrar los certificados SSL

Para conseguir comunicación segura entre los sitios SAML fuente y destino, la comunicación debe estar encriptada. Adicionalmente los certificados deben usarse para verificar la identidad de cada parte en la interacción SAML.

Usaremos en cada dominio los keystores *DemoIdentity.jks* que aparecen al configurar cada dominio.

1. Usar keytool:

keytool -genkey -keypass testkeypass -keystore DemoIdentity.jks -storepass DemoIdentityKeyStorePassPhrase -keyalg rsa -alias testalias.

2. Exportar el certificado `testalias.der`:

keytool -export -keypass testkeypass -keystore DemoIdentity.jks -storepass DemoIdentityKeyStorePassPhrase -alias testalias -file testalias.der

6. Paso 5. Configurar domainA como sitio fuente SAML

Configuraremos el sitio fuente como una instancia de un *SAML Credential Mapper V2*, que actúa como productor de asertos SAML, permitiendo a *domainA* para actuar como sitio fuente para SSO.

1. Log-in en la consola de administración de *domainA*.

2. Seleccionar *Security Realms*.

3. Seleccionar un security realm. El realm por defecto es *myrealm*.

4. Seleccionar el tab de *Providers* y seleccionar el tab *Credential Mappings*.

5. Si el *SAML CredentialMapper* no existe entonces crear uno tal como se indica en la figura:

Crear SAMLCredentialMapper

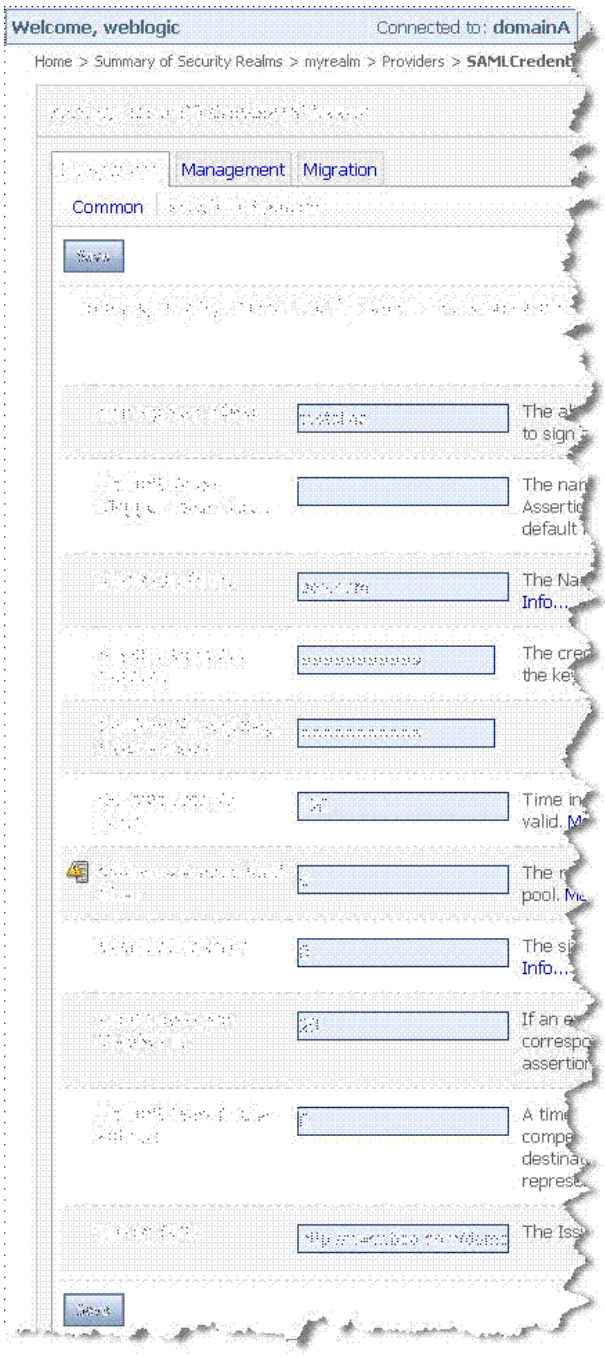
6. Seleccionar *SAMLCredentialMapper* y entonces seleccionar proveedor específico.

7. Seleccionar *Lock and Edit* para editar los settings.

8. Editar la URI del emisor `http://www.bea.com/demoSAML`. Esta URI dice a (*domainB/appB*) el origen del mensaje. La URL se usa para garantizar unicidad.

9. Entrar la *Signing Key Alias* (testalias) y la *Signing Key Alias Phrase* (testkeypass) que se usaron en keytool.

10. Seleccionar el resto de parámetros con los que se muestran en la imagen:



Crear SAMLCredentialMapper

11. Click salvar.

12. Activar Cambios.

En este punto, el proveedor SAML credential mapper está configurado para permitir a *domainA* ser un sitio fuente para asertos SAML. También está configurado para usar el keystore generado en el Paso 4.

7. Punto 6: Configurar las propiedades de la relying party

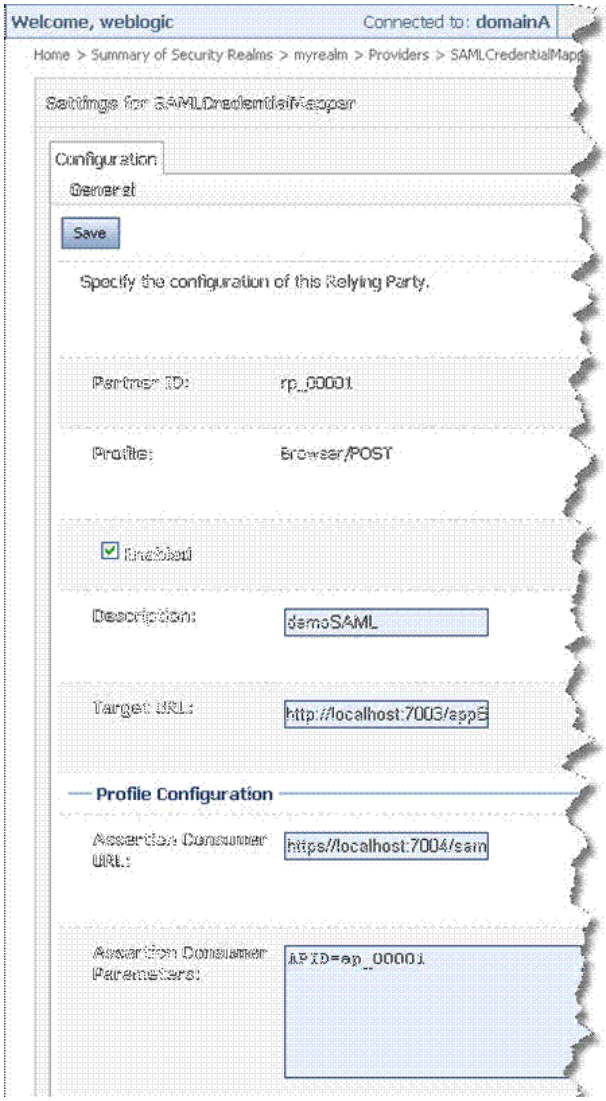
Cuando configuramos el servidor como fuente de asertos SAML se deben registrar las partes que pueden pedir assertions asertos SAML que serán aceptados. Para una *relying party* SAML, puedes especificar: el profile SAML usado, detalles sobre la *relying party* y los atributos esperados en los asertos para la *relying party*.

La *relying party* determina si confía en los asertos que le proporciona la *asserting party*. SAML define un número de mecanismos que permiten a la *relying party* a confiar en los asertos que se le proporcionan.

1. En el tab de *Management* click *Relying Parties*.

2. En la tabla de *Relying Parties*, click *New*.

3. En el menú de *Profile* seleccionar *Browser/POST*. En el camo *Destination* poner el nombre *demoSAML* para identificar la *relying party* como se muestra en la figura:



Settings para el SAMLCredentialMapper

Los parámetros son:

Enabled: true

Target URL: http://localhost:7003/appB/admin/services/services.jsp

Assertion Consumer URL: https://localhost:7004/samlacs/acs

Assertion Consumer Parameters: APID=app_00001

Signature Required: true

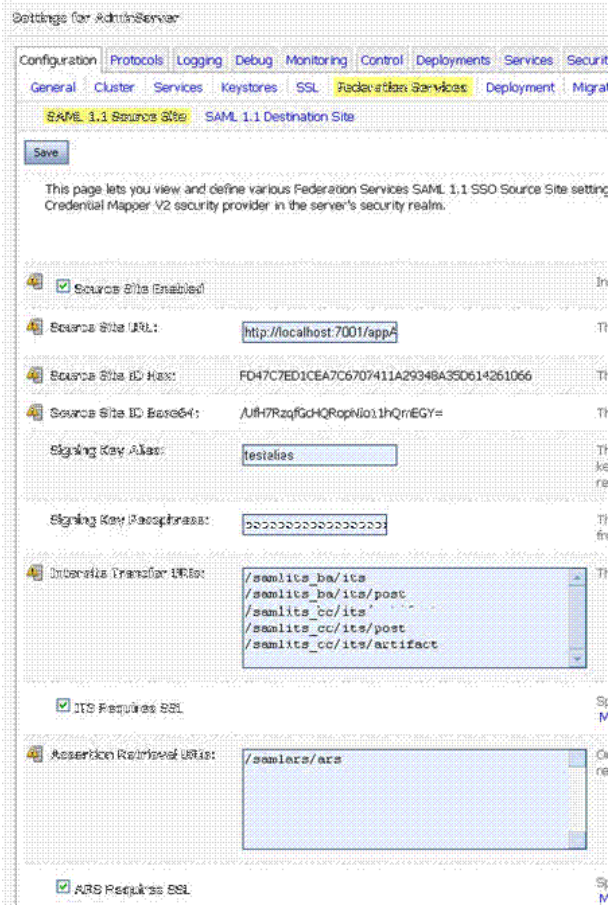
Include Keyinfo: true

8. Configurar SAML en el source site

Procede ahora configurar varios servicios de federación para el servidor que ejecuta *appA*. Estos settings permiten que ese servidor sea un *source site* SAML, definen las URIs del sitio y las URIs de servicio, añadimos certificado para firmar asertos y configurarán SSL para recoger asertos.

1. En la consola de administración, en *Domain Structure* seleccionar *Environment* y *Servers*.

2. Seleccionar *AdminServer* y entonces en los settings para para ese servidor clickear *Federation Serivces* como se indica en la figura:



Settings para AdminServer

9. Paso 8: Configurar domainB como SAML destination site

Hay que crear una instancia *SAML Identity Assertion Provider V2*. Este proveedor valida los asertos SAML checkeando la firma y validando el certificado para confiar en el registro de certificados mantenido por el proveedor. Por eso lo primero que hay que hacer es importar el certificado generado en el Paso 4 en el registro de certificados del proveedor.

Importar el certificado:

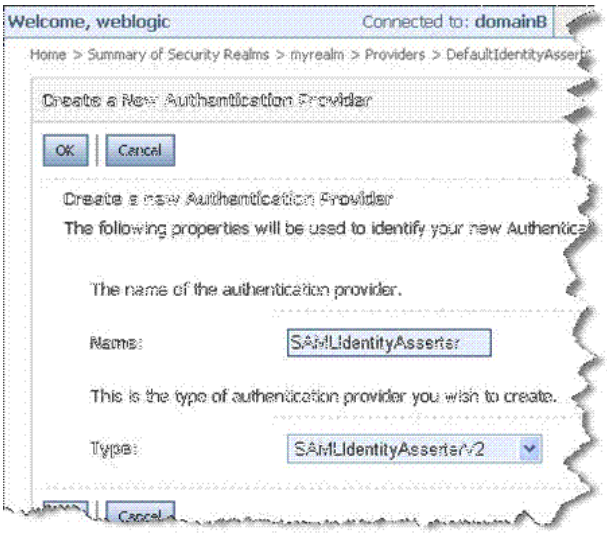
1. Copiar el fichero `testalias.der` en `C:\bea\weblogic92\server\lib`.

2. Logearse en la consola de administración del *domainB*.

3. Seleccionar *myrealm*.

4. Seleccionar el tab de *Providers* y luego el de *Authentication*.

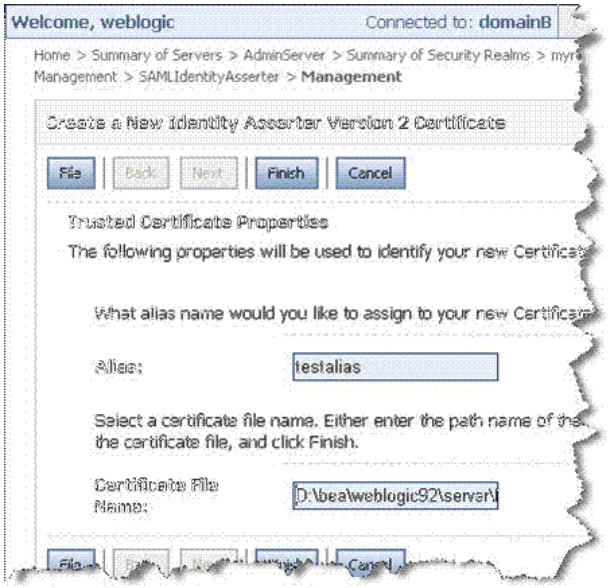
5. Si el *SAMLIdentityAsserter* no existe, entonces crear uno nuevo como se indica en la figura:



Crear un nuevo identity asserter

6. Seleccionar *SAMLIdentityAsserter*, clickear el tab de *Management* y clickear *Certificates*.

7. En el diálogo de *Certificates* clickear *New* como se muestra en la figura:



Crear un nuevo certificado de asserción

8. En el campo *Alias* poner el alias del certificado.

9. Entrar el nombre del certificado.

10. Clickear *Finish*.Si no hay problemas se muestra el mensaje "The certificate has beeen successfully registered".

10. Configurar las propiedades de la asserting party

Cuando se configura un servidor como consumidor de asertos se necesita configurar las partes cuyos asertos serán aceptados. Para una *Asserting Party* puedes especificar el profile SAML usado, detalles sobre la *asserting party* y los atributos esperados en asertos recibidos de la *asserting party*.

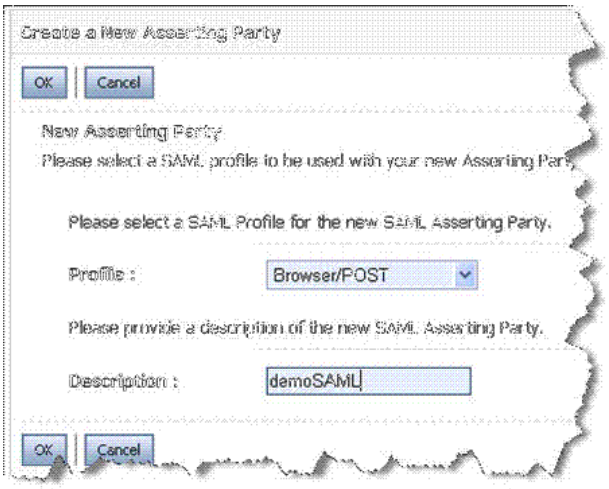
La *asserting party* aserta que un usuario ha sido autentificado y se le han dado ciertos atributos. Por ejemplo, está el usuario `ssouser` que es autentificado en su dominio usando un mecanismo de password. Las *asserting parties* se llaman *SAML authorities*:

1. En el tab *Management* clickear *Asserting Parties*.

2. En la tabla *Asserting Parties*, clickear *New*.

3. En el menú de *Profile* seleccionar *Browser/POST*. En el campo *Description* entrarel

nombre *demoSAML* para identificar la *asserting party* como se indica en la figura:



Crear un nueva asserting party

Los parámetros son:

Enabled: true

Target URL: http://localhost:7001/appA

POST Signing Certificate alias: testalias

Source Site Redirect URIs: /appB/admin/services.jsp

Source Site ITS URL: https://localhost:7002/samlits_ba/its

Source Site ITS Parameters: RPID=rp_00001

Issuer URI: http://www.bea.com/demoSAML

Signature Required: true

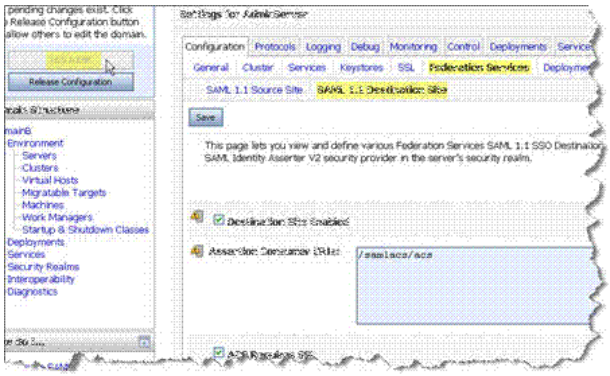
Asserting Signing Certificate Alias: testalias

11. Configurar el destination site SAML (1.1)

Configuraremos parámetros del server ejecutando *appB*. Estos settings permiten al servidor servir como un *destination site* SAML, define URIs de servicio, añadir un certificado para firmar respuestas de profile tipo POST y configurar SSL para*Assertion Consumer Service* (ACS)

1. Em la consola de administración seleccionar *Environment*, después *Servers* en la ventana de *Domains Structure*.

2. Seleccionar *AdminServer*, y luego en los *Settings* para ese servidor clickear *Federation Services* y entonces en el tab *SAML 1.1 Destination Site* como se ve en la figura:



Crear un nueva asserting party

Los parámetros son:

Destination Site Enabled: true

Assertion Consumer URIs: /samlacs/acs

ACS Requires SSL: true

SSL Client Identity Alias: testalias

SSL Client Identity Pass Phrase: testkeypass

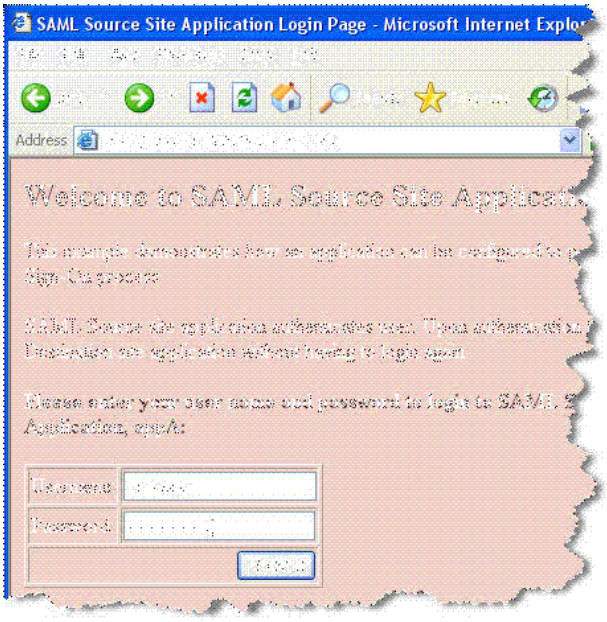
POST Recipient Check Enabled: true

POST one USe Check Enabled: true

Used Assertion Cache Properties: APID:=ap_00001

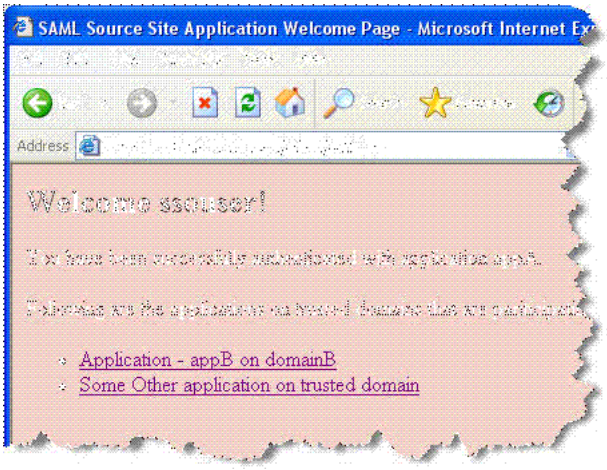
12. Testear SSO

Abrir un browser en la URL: `http://localhost:7001/appA/`. La autentificación tipo FORM configurada para *appA* mostrará `login.jsp`. Entrar usuario y password:



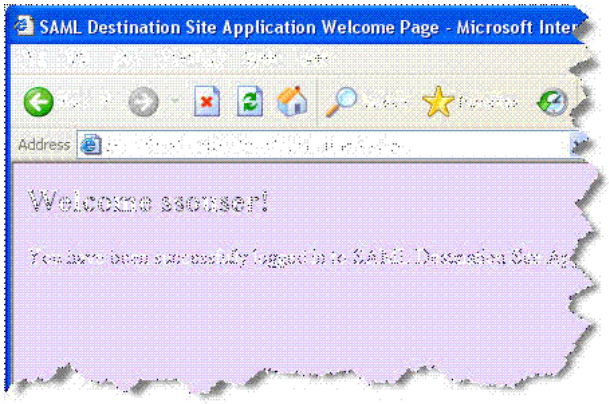
Browser en el appA login

La página `auth.jsp` se muestra. Esta página muestra un link a *appB* (`http://localhost:7003/appB/admin/services.jsp`).



Browser en el appA login

Haciendo click en este link llamaremos al ITS servlet y causaremos la generación del aserto. Entonces el control serás transferido al sitio de destino:



Browser en el appA login