Autenticación centralizada para los sistemas de información de los Institutos Tecnológicos y DGEST

José Jesús Sánchez Farías
Instituto Tecnológico de Celaya
jesus.sanchez@itcelaya.edu.mx

Guillermo Rodríguez Villafaña
Instituto Tecnológico de Celaya
guillermo.rodriguez@itcelaya.edu.mx

Luis Alberto López González
Instituto Tecnológico de Celaya
luislao@itcelaya.edu.mx

Juan Ignacio Cerca Vázquez
Instituto Tecnológico de Celaya
nacho@itcelaya.edu.mx

Resumen

La Dirección General de Educación Superior Tecnológica (DGEST) [1], cuenta con un Sistema Integral de Información, el cual hace uso de una cuenta institucional compuesta por un nombre de usuario y una contraseña para acceder a los módulos de información. Los usuarios finales de dichos módulos son los Institutos Tecnológicos, los cuales también tienen sus propios sistemas de información con sus propias cuentas de usuario para acceder a estos. La problemática para el usuario final surge al momento de tener que gestionar varias cuentas para diferentes sistemas de información, recordarlas y mantenerlas actualizadas todas a la vez. Para solucionar este problema surgen los **sistemas de administración de identidad federada,** los cuales permiten

gestionar una sola cuenta para acceder a los sistemas, concentrando en un solo punto el acceso; existen varias soluciones de código libre tales como Shibboleth, SimpleSAMLphp, entre otros [2][3].

Palabra(s) Clave(s): Administración de identidad federada, federación de identidades, SAML.

1. Introducción

La proliferación del uso de Internet y oferta de servicios se ha expandido a nivel global, no es la excepción en el Sistema Nacional de Institutos Tecnológicos (SNIT) [4], los cuales cada día hacen uso de tecnologías innovadoras para desempeñar de mejor manera sus actividades. Una herramienta tecnológica de uso diario para el personal de los Institutos Tecnológicos y la DGEST son los sistemas de información, específicamente los hospedados en Internet, lo cual facilita su accesibilidad. Desgraciadamente el crecimiento de estos sistemas no ha sido el más adecuado, si bien, han facilitado las actividades, se han mejorado tiempos, se optimizan recursos tanto económicos como humanos; ha faltado un control importante en su crecimiento. Los sistemas que se generan tanto en la DGEST como en los Institutos Tecnológicos, son independientes entre sí, incluso una misma Institución puede generar múltiples sistemas independientes entre sí, no existe algo que los ligue y permita compartir información entre ellos. En el mayor de los casos cada vez que se genera un nuevo sistema, a los usuarios finales se les entrega un nombre de usuario y una contraseña para cada sistema generado; por lo tanto si una Institución tiene “n” sistemas de información, se traduce en “n” cuentas de usuario para acceder a los mismos.

La mayoría de las aplicaciones o sistemas de información requieren que el usuario se autentique; es decir, que éste demuestre que es quien dice ser, habitualmente este proceso se hace a través de un nombre de usuario único y una contraseña asociada. Muchas de estas aplicaciones almacenan datos personales, en muchas ocasiones comunes entre ellos, como pueden ser una dirección de correo electrónico, fecha de nacimiento, domicilio, etc.

En principio, al tratarse de aplicaciones independientes entre sí, cada una de ellas utiliza su propio sistema de autenticación, lo cual implica inconvenientes al usuario, ya que conforme surgen nuevos sistemas, este tiene que recordar diferentes nombres de usuario y contraseñas para diferentes sistemas, así también tiene que autenticarse en cada uno de ellos cada vez que desea usarlos. Adicional a esto, el usuario tiene que mantener actualizada la información personal en cada uno de ellos. Esta situación no solo es incómoda para los usuarios, sino que lleva a que estos realicen prácticas no muy adecuadas y que además pueden comprometer la seguridad de sus sistemas. Entre las prácticas no aceptables está: uso de contraseñas débiles o poco seguras; para recordar las contraseñas, las anotan en papel o un documento en la computadora; reutilizar la misma contraseña para los diferentes sistemas.

El objetivo de las herramientas tecnológicas que se describen en este artículo es dar solución a estos inconvenientes, ofreciendo un esquema de autenticación centralizado y gestión de datos personales.

1.1 Autenticación

Antes de desarrollar el tema es necesario comprender el término de **Autenticación**, el cual consiste en comprobar o certificar por parte de un sistema, que un usuario es quien dice ser; lo más habitual para comprobar esto, es utilizar un nombre de usuario en combinación con una contraseña. Una característica importante para que este tipo de sistemas se considere federado, es que dicha autenticación funcione en más de un sistema de información.

1.2 Identidad Federada

Consiste en un mecanismo que permite a una organización aceptar usuarios que ya han sido autenticados en otras organizaciones diferentes, de esta manera se les otorga acceso a ciertos recursos sin necesidad de gestionar nuevamente su identidad.

En la administración de la identidad federada existen tres componentes principales:

- El usuario, quien tiene una identidad digital.
- El Proveedor de Servicio (SP por sus siglas en inglés), el cual provee servicios (por ejemplo acceso a recursos protegidos) para autenticar a los usuarios.

- El Proveedor de Identidad (IdP por sus siglas en inglés), cuyo papel es autenticar al usuario; posteriormente el IdP envía la información de dicha autenticación al Proveedor de Servicio.

La autenticación puede ocurrir de varias maneras: el SP puede iniciar una petición de autenticación al IdP y éste determinar si un usuario es válido, si es así, el usuario se considera autenticado en el SP, otra manera es que el usuario primero se autentique contra el IdP y posteriormente accede al SP. En cual quiera de los casos, la tecnología habilita Single Sign-On (SSO), en el cual el IdP autentica al usuario, y por lo tanto permitiendo el acceso a recursos protegidos en el SP. En un sistema de identidad federada se establece un círculo de confianza que permite a un usuario autenticado en un organismo de la federación acceder a recursos de otro organismo dentro de la misma federación.

Los dos sistemas de administración de identidad federada a describir en este artículo son Shibboleth y SimpleSAMLphp, los cuales son de código libre, lo que indica que pueden ser descargados y utilizados libremente, apegándose a los términos de licencia indicados en cada uno de ellos.

1.3 Single Sign-On (SSO)

Es el proceso de autenticación de usuarios que permite a estos proporcionar un nombre de usuario y contraseña con el fin de acceder a múltiples aplicaciones. Este proceso realiza el proceso de autenticación para todas las aplicaciones o sistemas a las cuales el usuario tiene derecho, esto evita que se tenga que solicitar nuevamente la autenticación del usuario cuando se desea cambiar entre aplicaciones. En resumen el usuario se autentica una sola vez contra el sistema de una organización y su identidad es reconocida por todos los sistema ligados a dicha organización.

1.4 SAML (Security Assertion Markup Language)

Antes de profundizar en sistemas específicos de identidad federada, es conveniente explicar uno de los estándares sobre este tema. SAML desarrollado por OASIS [5], es un estándar para el manejo de la identidad federada, el cual consiste de un lenguaje de

marcado basado en XML que permite el intercambio de datos de autenticación y perfiles de autorización entre distintas organizaciones y dominios, es decir entre distintos IdP's y SP's.

2. Métodos

Las dos implementaciones para la administración de identidad federada a describir son *Shibboleth* y *SimpleSAMLphp*. El primero de ellos desarrollado en el lenguaje de programación Java y que permite la implementación tanto del Proveedor de Identidad (IdP) como del Proveedor de Servicios (SP); el segundo desarrollado en lenguaje de programación PHP y que también permite la implementación tanto del IdP como del SP. Ambos sistemas al estar basados en estándares como SAML se pueden comunicar uno con el otro aunque estén desarrollados en lenguajes de programación diferentes, identificando claramente quién actuará como IdP y quién como SP.

2.1 Shibboleth

Es un proyecto de código abierto desarrollado inicialmente para Internet 2 [6] de la *National Science Foundation Internet2 Middleware Initiative*, que implementa un sistema de identidades federadas para el intercambio de atributos basados en estándares abiertos como *SAML*. Además provee funcionalidad de privacidad que permite al usuario o incluso a nivel de organización controlar los atributos que se intercambian para cada aplicación.

2.2 SimpleSAMLphp

Este también es un proyecto de código abierto desarrollado completamente en PHP nativo, de igual manera implementa un sistema de identidades federadas para trabajar con la autenticación de los sistemas. Este proyecto es desarrollado por UNINETT [7] y permite hacer la función tanto de Proveedor de Identidad como Proveedor de Servicio.

2.3 Implementación

El panorama general de implementación del sistema de administración de identidades federadas consiste en instalar inicialmente un Proveedor de Identidad en servidores de

la DGEST, este realiza la tarea de autenticación a los usuarios de los Institutos Tecnológicos; la base de datos de usuarios a autenticar está hospedada bajo un directorio LDAP [8], el cual está internamente enlazado con dicho Proveedor de Identidad. Este proveedor se implementó con el sistema Shibboleth. El Proveedor de Servicio se instala en cada una de las Instituciones que desean enlazarse con el Proveedor de Identidad ubicado en un sitio central de la DGEST, dicho proveedor se implementó con el sistema SimpleSAMLphp, se eligió este proveedor debido a que la inmensa mayoría de las Instituciones tienen sus sistemas basados en el lenguaje de programación PHP, de esta manera es fácil implementar el Proveedor de Servicio para sus sistemas y aplicaciones locales. Cabe mencionar que de igual manera se puede utilizar el Proveedor de Servicios de Shibboleth.

Los procedimientos generales llevados a cabo para implementar el sistema de administración de identidades federadas en el SNIT es el siguiente:

- Implementación del Proveedor de Identidad en la DGEST.
- Implementación de los Proveedores de Servicio en los Institutos Tecnológicos.

2.4 Implementación del Proveedor de Identidad en la DGEST

El siguiente procedimiento explica la implementación del IdP con las características básicas, se utilizó CentOS 5.6 como sistema operativo.

- **Java.** Descargar Java Development Kit (JDK) de la página de descargas de Oracle (*http://www.oracle.com/technetwork/java/javase/downloads/index.html*), el nombre del archivo a descargar debe ser equivalente a jdk-uXX-linux-i586.bin. Este se coloca en el directorio /usr/local/src, una vez descargado el paquete, asignar permisos de ejecución y ejecutar el instalador:

  # chmod +x jdk-uXX-linux-i586.bin

  # ./jdk-uXX-linux-i586.bin

- **Apache Tomcat.** La versión 2 del Proveedor de Identidad de Shibboleth, es una aplicación web Java estándar basada en la especificación Servlet 2.4, por lo tanto para instalar el IdP es necesario un contenedor de Servlets para realizar el desplegado de la aplicación del IdP. Existen muchos contenedores de Servlets, uno de los más populares es Apache Tomcat, ya que es fácil de utilizar e instalar.

a. Descargar el paquete de la página de Tomcat (*http://tomcat.apache.org/download-60.cgi*) y guardar en la carpeta */usr/local/src*.

b. Descomprimir e Instalar.

 # tar –zxvf apache-tomcat-6.0.37.tar.gz; mv apache-tomcat-6.0.37 tomcat6

Se definen las siguientes variables de ambiente:

 # export JAVA_HOME=/usr/local/src/jdk/jdk1.6.0_25; export PATH=$JAVA_HOME/bin:$PATH

 # export CATALINA_HOME=/usr/local/src/tomcat6; export CATALINA_BASE=/usr/local/src/tomcat6

Para mantener activas estas variables después de reiniciar la maquina, se agregaron al archivo *.bash_profile*.

c. Iniciar Tomcat

 # TOMCAT_HOME/bin/catalina.sh start

- **Shibboleth IdP**

Para la instalación del Proveedor de Identidad, es necesario seguir los siguientes pasos:

a. Descargar el paquete del Proveedor de Identidad de Shibboleth desde la página oficial (*http://shibboleth.net/downloads/identity-provider/latest/*).

b. Extraer el archivo descargado *shibboleth-identityprovider-2.3.0-bin.zip* y hacer ejecutable el script instalador *install.sh*

 # unzip shibboleth-identityprovider-2.3.0-bin.zip; cd shibboleth-identityprovider-2.3.0

 # chmod u+x install.sh

c. Copiar las librerías Endorse Xerces y Xalan ubicados en el paquete fuente de Shibboleth IdP a la carpeta *TOMCAT_HOME/endorsed*. En el directorio *endorsed* están ubicados archivos *jar*, los cuales serán copiados al directorio *endorsed* de Tomcat.

 # cp -r endorsed/ /usr/local/src/tomcat6

 # export JAVA_ENDORSED_DIRS=/usr/local/src/tomcat6/endorsed

d. Ejecutar el script de instalación del IdP de Shibboleth

 # ./install.sh

Este script hará una serie de preguntas, entre ellas es indicar el directorio de instalación, el default es */opt/shibboleth-idp*, así como el nombre del host del servidor IdP (*idp.dgest.gob.mx*). Es importante que este nombre de host no se cambie posteriormente a la instalación, ya que este nombre

es utilizado en los archivos de configuración del IdP y SP. No menos importante es recordar el password de instalación ya que se utilizará posteriormente durante la configuración. El directorio de instalación indicado será referenciado por la variable de entorno *IDP_HOME*.

e. Se definen algunos enlaces simbólicos por conveniencia. Se enlazan el directorio */etc/shibboleth* al directorio de configuración de shibboleth-idp, así como el directorio */var/log/shibboleth* al directorio de bitácoras de shibboleth-idp.

 # ln –s /opt/shibboleth-idp/conf /etc/shibboleth; ln –s /opt/shibboleth-idp/logs /var/log/shibboleth

f. Definir la variable de ambiente *IDP_HOME*

 # export IDP_HOME=/opt/shibboleth-idp

Se agrega la siguiente entrada al archive *.bash_profile* para que se exporten las variables al reiniciar el sistema.

 # IDP_HOME=/opt/shibboleth-idp; export IDP_HOME

g. Desplegado del archivo WAR del IdP, archivo está ubicado en *IDP_HOME/war*. Para esto crear y abrir el archivo *TOMCAT_HOME/conf/Catalina/localhost/idp.xml* y agregar la siguiente contenido:

<Context docBase="/opt/shibboleth-idp/war/idp.war" privileged="true" antiResourceLocking="false" antiJARLocking="false" unpackWAR="false" swallowOutput="true" />

h. Hasta este punto la instalación y configuración básica del IdP está completada, para probar el funcionamiento del mismo, se reinicia Tomcat para realizar el desplegado de la aplicación:

 # /usr/local/src/tomcat6/bin/catalina.sh stop; /usr/local/src/tomcat6/bin/catalina.sh start

Agregar la dirección IP local y el nombre del host al archivo */etc/hosts*:

 # vi /etc/hosts y agregar: 127.0.0.1 idp.dgest.gob.mx idp

Una vez reiniciado el servicio de Tomcat, mediante un navegador acceder a la siguiente dirección *http://idp.dgest.gob.mx:8080/idp/profile/Status*. Si se recibe una página con el mensaje *"Ok"*, significa que la configuración realizada hasta el momento está correcta.

i. **Definición del mecanismo de autenticación.** El IdP de Shibboleth permite varios mecanismos de autenticación, el Proveedor de Servicios requiere precisamente que uno de estos mecanismos sea implementado.

   a. **Autenticación por Usuario/Contraseña.** Se utiliza este mecanismo ya que el personal de los Institutos Tecnológicos ya tienen precisamente una cuenta Institucional asociada. Este manejador de autenticación por lo tanto requiere de un nombre de usuario y contraseña para autenticar al usuario como tal, este procedimiento se realiza mediante *Java Authentication and Authorization Service (JAAS)* [9]. Este manejador presenta al usuario una página de autenticación, donde se proporciona el nombre de usuario y contraseña validándolo contra un directorio LDAP. Para completar este paso se abre el archivo *handler.xml* ubicando en *IDP_HOME/conf*, se habilita el manejador *UsernamePassword* eliminando los comentarios alrededor del elemento *UsernamePassword* <*LoginHandler*>, nos aseguramos que el elemento *RemoteUser* <*LoginHandler*> permanezca comentado.

   b. **Servidor LDAP.** Se reutilizó el servidor LDAP manejado por la DGEST el cual está implementado por *Apache Directory* [10]. La dirección utilizada para acceder a este servicio es: *ldap://localhost:10389*; el nombre distinguido (DN) utilizado es: *ou=usuarios,dc=dgest.gob.mx,o=dgest*

   c. **Configuración de JAAS.** La política de autenticación JAAS a ser utilizada con Shibboleth es especificada por el atributo *jaasConfigurationLocation* dentro del elemento *AuthenticationHandler*. El desaplegado del IdP de Shibboleth requiere el uso de *AuthenticationHandler* como entrada de configuración para la política JAAS. Para esto se abre el archivo *login.config* ubicado en el directodio *IDP_HOME/conf*, al final del archivo está ubicado el elemento *ShibUserPassAuth*, este debe configurarse de la siguiente manera:

ShibUserPassAuth { edu.vt.middleware.ldap.jaas.LdapLoginModule required
host="ldap://localhost:10389"
base = "ou=usuarios,dc=dgest.gob.mx,o=dgest" ssl="false" userField="uid";};

j. **Configuración de metadatos.** La configuración de metadatos se realiza al momento de instalar el SP, ya que deberán intercambiar archivos para establecer la comunicación.

- En este punto queda instalado y configurado el IdP de Shibboleth.

2.5 Implementación de los Proveedores de Servicio en los Institutos Tecnológicos

Como se mencionó anteriormente el Proveedor de Servicio (SP) se implementó con SimpleSAMLphp, el cual estará enlazado con el IdP de Shibboleth, comunicándose durante el proceso de autenticación y así los usuarios puedan acceder a los recursos protegidos por parte del SP. El proceso de implementación para el Proveedor de Servicios para los Institutos Tecnológicos es el siguiente:

- **Instalación de SimpleSAMLphp.** Se sigue el procedimiento estándar publicado en la página oficial del sistema [11].
- **Configuración del Proveedor de Servicio.** Siguiendo el procedimiento de instalación de SimpleSAMLphp el directorio final de instalación es */var/simplesamlphp*. Bajo este directorio hay que realizar algunas configuraciones bajo *config/authsources.php*. Se define las siguientes opciones de configuración:

  'default-sp' => array('saml:SP', 'privatekey' => 'saml.pem', 'certificate' => 'saml.crt',
  'entityID' => NULL, 'idp' => 'http://idp.dgest.gob.mx/idp/shibboleth', 'discoURL' => NULL,),

  El IdP le exige al SP tener un certificado, este permitirá el envío de peticiones y respuestas de manera encriptada entre ambos. Para esto cambiarse al directorio *cert* y generar el certificado con el comando *openssl*:

  # cd cert; openssl req -newkey rsa:2048 -new -x509 -days 3652 -nodes -out saml.crt -keyout saml.pem
- **Definición del IdP.** El Proveedor de Servicio necesita conocer el Proveedor de Identidad con el cual se estará conectando. Esto es configurado en *metadata/saml20-idp-remote.php* con la siguiente información:

  $metadata['http://idp.dgest.gob.mx/idp/shibboleth'] = array(
  'name' => array('en' => 'DGEST IDP', 'no' => 'DGEST IDP',),
  'description' => 'Aqui puedes autenticarte al proveedor de indentidad de la DGEST',
  'SingleSignOnService' =>'http://idp.dgest.gob.mx/idp/profile/SAML2/Redirect/SSO',

'SingleLogoutService' => 'https://idp.dgest.gob.mx/examples/jsp/lgoog.jsp',
'certFingerprint' => 'C80FC64BC9B5AE40934C744FB94D9F231657F6E5');

- **Intercambio de metadatos con el IdP.** Con el fin de completar la conexión entre el IdP y SP, se deben agregar los metadatos del SP al IdP. Estos metadatos se encuentran en la pestaña **Federation**, ahí se muestra una opción para descargar el archivo de metadatos, el cual se guarda como *simplesaml-metadata.xml* y posteriormente se copia al directorio *metadata* del IdP de Shibboleth.

  Durante este paso cada uno de los Institutos Tecnológicos tendrá que generar su archivo de metadatos y enviarlo al administrador del IdP en la DGEST, el cual recibirá y posteriormente copiará al directorio *metadata* ubicado en el *IDP_HOME*.
- **Integración de las aplicaciones o sistemas de los Institutos Tecnológicos.** Para las aplicaciones que se quieran proteger, se deben agregar algunas líneas de código PHP.
  a. **Registrar las clases de SimpleSAMLphp con PHP autoloader.** *require_once('../../lib/_autoload.php');*
  b. **Requerir autenticación de usuario para los lugares que se desea proteger.** *$as = new SimpleSAML_Auth_Simple('default-sp'); $as->requireAuth();*
  c. **Acceder a los atributos de usuario.**
     $attributes = $as->getAttributes(); print_r($attributes);
     Entre los atributos que se pueden consultar sobre el usuario es su nombre de usuario, nombre completo, correo electrónico, etc. De esta manera se puede utilizar por ejemplo el nombre del usuario para determinar el rol al que pertenece y determinar la página de opciones a mostrar.

3. Resultados

La Dirección General de Educación Superior Tecnológica está en proceso de implementación de este esquema a nivel nacional en los Institutos Tecnológicos, en el cual se podrán mejorar las comunicaciones entre sistemas de información. Actualmente está implementado el IdP en un servidor central, internamente está en proceso de integración del Sistema Integral de Información de la DGEST a la federación, de esta

manera se homologan en la parte de autenticación los sistemas funcionales e independientes de la Institución.

De igual manera se han comenzado a realizar pruebas de integración del Sistema Integral de Información del Instituto Tecnológico de Celaya, en el cual se instaló el Proveedor de Servicio SimpleSAMLphp, logrando una comunicación e intercambio de información con el Proveedor de Identidad de la DGEST.

4. Conclusión

La principal motivación para la implementación de un sistema de identidad federada es lograr la integración entre los sistemas centrales a nivel nacional proporcionados por la Dirección General de Educación Superior Tecnológica con los sistemas locales de cada uno de los Institutos Tecnológicos, de esta manera se mejora la comunicación e intercambio de información entre las diferentes Instituciones. Así, cada vez que se desarrollen nuevos sistemas tanto a nivel nacional como a nivel local en las instituciones, ya contarán con un respaldo en cuanto a la plataforma de autenticación de usuarios, evitando duplicidad de cuentas, centralizando la identidad de los mismos y aprovechando un trabajo previo para la generación de cuentas institucionales que en la actualidad ya son un estándar y requisito por parte de la DGEST para acceder a los sistemas de información.

El Sistema Nacional de Institutos Tecnológicos es un sistema muy grande en el cual el intercambio de información es complicado al igual que la coordinación de grupos de trabajo para el desarrollo de los sistemas de información. Día a día surgen y se desarrollan nuevos proyectos de software que pueden ser aprovechados por toda la comunidad tecnológica a nivel nacional, pero que muchas veces perecen en el camino por no cumplir con estándares de integración a los sistemas centrales o implicar un esfuerzo muy grande de implementación de los mismos. Mediante esta iniciativa se pretende que el talento y el trabajo de las Instituciones se aprovechen y sea más fácil su incorporación a un Sistema de Información Integral Nacional en el que el flujo de la información sea más flexible y permita la toma de decisiones a los altos funcionarios para la mejora de los Institutos Tecnológicos.

Referencias

[1] Dirección General de Educación Superior Tecnológica (DGEST): http://www.snit.mx/.

[2] Shibboleth: http://shibboleth.net/.

[3] SimpleSAMLphp: http://simplesamlphp.org/

[4] Sistema Nacional de Institutos Tecnológicos (SNIT): http://www.snit.mx/informacion/institutos-tecnologicos-de-mexico.

[5] OASIS: https://www.oasis-open.org/.

[6] Internet 2: http://www.snit.mx/telecomunicaciones/que-es-internet-2.

[7] UNINETT: https://www.uninett.no/.

[8] LDAP: http://ldapman.org/articles/sp_intro.html.

[9] JAAS: http://docs.oracle.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html.

[10] Apache Directory: http://directory.apache.org/apacheds/.

[11] Procedimiento de instalación de SimpleSAMLphp: http://simplesamlphp.org/docs/stable/simplesamlphp-install.