adAS, el nuevo sistema de Single Sign-On de la Universidad de Salamanca

adAS, the new USAL's Single Sign On System

Inmaculada Bravo García, Reyes Hernández Rodríguez

Resumen

La Universidad de Salamanca ha financiado el desarrollo de un nuevo software de Single Sign-On, adAS (advanced Authentication Server) [1], que nos ha permitido desplegar una infraestructura de autenticación y autorización completa, dentro de nuestro entorno tecnológico.

Este software ha sido desarrollado en PHP por la empresa española PRiSE[2] y se ofrece a la comunidad como Software Libre bajo licencia Apache 2.0.

adAS (advanced Authentication Server) es un Servidor de Autenticación Avanzado, basado en PAPI[3], que realiza funciones de Proveedor de Identidad, muy flexible, sencillo de administrar y sencillo de integrar aplicaciones .

adAS es un Servidor de Autenticación Avanzado basado en PAPI

adAS es multiprotocolo PAPI, SAML2, SAML1.1/Shibboleth1.3 [4][5]. Ofrece varios métodos de autenticación. La gestión de atributos es muy potente, estos se pueden obtener de diferentes fuentes de datos simultáneamente, se pueden realizar filtrado y tratamiento de los mismos antes de emitirlos a la aplicación. Existen multitud de conectores para integrar diferentes aplicaciones y dispone de una interfaz gráfica amigable de administración, que facilita todas las tareas.

Palabras clave: Single Sign On, Software Libre, PHP, servidor de autenticación, gestión de atributos, autorizacíon, federación, PAPI, SAML, Shibboleth, seguridad

Summary

The University of Salamanca has funded the development of new software for Single Sign-On, named ADAS (Advanced Authentication Server) [1], which allowed us to deploy a complete authentication and authorization system within our technological environment.

This software has been developed in PHP by the Spanish company PRiSE[2] and is offered to the community as Software Libre under Apache license 2.0.

adAS is an Advanced authentication server based on PAPI [3], which performs Identity Provider features. It is very flexible, simple to manage and it is easy to integrate applications with it.

La gestión de atributos es muy potente

adAS is multiprotocol PAPI, SAML2, SAML1.1/Shibboleth1.3 [4][5] and it offers several authentication methods. Attribute management is very powerful, these can be obtained from different data sources simultaneously, can perform filtering and processing them before issuing them to the application. There is a multitude of connectors to integrate different applications, as well a friendly graphical management interface, which facilitates all the tasks.

Keywords: Single Sign On, Software Libre, PHP, authentication server, attribute management, authorization, federation, PAPI, SAML, Shibboleth, security

1. Introducción

Se describe el escenario inicial de partida: la Universidad de Salamanca ofrece una amplia gama de servicios a sus usuarios, entorno a 60.000 usuarios activos actualmente, para acceder a dichos servicios el usuario se debe autenticar en cada uno de ellos con su par usuario/contraseña. En general, las credenciales de estos usuarios se almacenan de modo centralizado en el directorio institucional en OpenLDAP.

La contraseña del usuario es la misma para entrar en todas las aplicaciones, por lo cual se incrementa la necesidad de evitar que sus credenciales se vean comprometidas.

El número de aplicaciones que precisan acceso al directorio comienza a ser elevado, provocando dificultad en la gestión del mismo y en garantizar la seguridad.

Anteriormente al inicio de este proyecto se ha desarrollado un importante esfuerzo en la securización del directorio [7].

Cada una de las aplicaciones cuenta con un formulario de petición de usuario y contraseña que posteriormente validan contra el directorio.

En este momento las posibles **deficiencias de seguridad** que trataremos de solucionar con adAS son las siguientes:

• A las aplicaciones se les provee de una cuenta administrativa con privilegios especiales para conectar con el directorio. Si los servidores que albergan dichas aplicaciones se ven comprometidos, se compromete también la seguridad del directorio.

Las aplicaciones pueden tener vulnerabilidades de LDAP injection o blind LDAP injection

• Los formularios de entrada recogen las credenciales del usuario, pudiendo quedar estas credenciales grabadas en bases de datos o en logs, que además de quedar a disposición de los administradores de la aplicación, por lo general quedarán almacenadas en servidores expuestos a internet, posiblemente con vulnerabilidades que ante un ataque dejará las contraseñas de nuestros usuarios a disposición del atacante.

• HTTPS: Si la aplicación no utiliza https las credenciales de los usuarios viajan en claro por la red pudiendo ser interceptadas.

• Las aplicaciones pueden tener vulnerabilidades de LDAP injection o blind LDAP injection[8], a través de las cuales se puede atacar al directorio.

• Las aplicaciones pueden estar mal programadas y pueden intentar hacer búsquedas en el directorio por atributos no indexados, lo que provoca cargas innecesarias en el directorio con su consiguiente pérdida de rendimiento.

• La percepción del usuario de que su contraseña debe introducirla en formularios en diferentes entornos, con diferentes aspectos más o menos institucionales, lo convierte en un usuario más proclive a ser engañado en webs falsas (phising).

El proceso de autorización está basado en el valor de determinados atributos del usuario

Además de los usuarios del directorio institucional, algunas aplicaciones utilizan otros directorios o bases de datos para almacenar usuarios específicos no pertenecientes a la institución, pero a los que se les dará acceso a determinados servicios. Por lo cual necesitamos que adAS permita **autenticar** usuarios contra diferentes fuentes simultáneamente.

Una vez que el usuario se haya autenticado correctamente, la siguiente fase es la autorización. El proceso de **autorización** está basado en el valor de determinados atributos del usuario.

Estos **atributos** no necesariamente están almacenados en el directorio, muchos de los datos de los usuarios se almacenan en las bases de datos de alumnos o de recursos humanos. Por lo tanto otra de las necesidades del sistema es que se puedan recoger datos de diferentes fuentes.

Por el tamaño de la institución y su gran diversidad, no se debe constreñir a los desarrolladores a utilizar una única tecnología, sino que se debe dar **libertad al desarrollador** y ofrecerle un conector apropiado a su aplicación.

Además existen múltiples empresas que ofrecen servicios a nuestros usuarios por lo que necesitamos ofrecerles **tecnologías de federación.**

2. Objetivos

Teniendo una vista general de este escenario inicial de la Universidad, las deficiencias que se deben corregir y las necesidades de los diferentes servicios, se plantean los siguientes objetivos.

Las aplicaciones no recogerán las credenciales de los usuarios

Objetivos enfocados al usuario:

• Disponer de un formulario de entrada único, un único sitio donde los usuarios deban introducir sus credenciales.
• Realizando un único login tendrá acceso a todos los servicios que se le ofrecen, ahorrándole tiempo y mejorando su eficiencia.

Objetivos enfocados a la seguridad:

• Las aplicaciones no recogerán las credenciales de los usuarios, por lo cual quedarán eliminadas todas las posibles vulnerabilidades a las que están expuestas, descritas en la introducción.
• Facilitar las auditorías, monitorizar los accesos de los usuarios y posibilitar la trazabilidad de sus acciones.

Objetivos enfocados a la gestión:

Posibilidad de autenticación de usuarios contra diferentes fuentes simultáneamente

• Único punto de **conexión con el directorio**, evitando dar acceso al directorio a las diferentes aplicaciones.
• Gestión de la **emisión de atributos**: cada aplicación recibirá solo determinados atributos.
• **Entorno gráfico** que facilite las tareas de gestión y ofrezca informes gráficos de funcionamiento del sistema.
• Estadísticas de utilización real de los aplicativos, importante para la planificación y distribución de los recursos.

Objetivos enfocados a los administradores de aplicaciones:

• Ofrecer un marco de trabajo flexible para poder integrar diferentes aplicaciones desarrolladas con diferentes tecnologías.
• Entregarles atributos de diferentes fuentes de datos.
• Posibilidad de autenticación de usuarios contra diferentes fuentes simultáneamente.
• Evitarles utilización de https, al no tener que manejar contraseñas de los usuarios.
• Posibilidad de conocer el método de autenticación, para tomar sus decisiones de autorización.

Objetivos enfocados a la continuidad, escalabilidad y mantenimiento y el contacto con el exterior:

• Difundir y promover su uso en las universidades, con el fin de crear una comunidad de usuarios de adAS para compartir experiencias, manuales, mejoras y nuevas funcionalidades, y sobre todo para

abrir la posibilidad de interoperar con otras instituciones en el futuro. Lo vemos como el inicio de un sistema que debe crecer y evolucionar continuamente.
• Debe ser Software Libre.
• Debe permitir acceder a los servicios federados que ofrece RedIRIS a través del SIR[6],y acceso a WOK de la federación de Fecyt.
• Ofrecer tecnologías de federación las empresas que necesite ofrecer servicios a nuestros usuarios, en lugar de enviarles listados con nuestros datos para que sean dados de alta en sus sistemas (pudiendo incumplir la LOPD).

3. Las principales características técnicas de adAS

Partiendo de las premisas descritas en los apartados anteriores, se desarrolla en php adAS con las siguientes características técnicas:

• **Multiprotocolo:** Proveedor de identidad que implementa los protocolos de identidad digital PAPI v1, SAML 1.1/Shibboleth 1.3 y SAML 2.0.

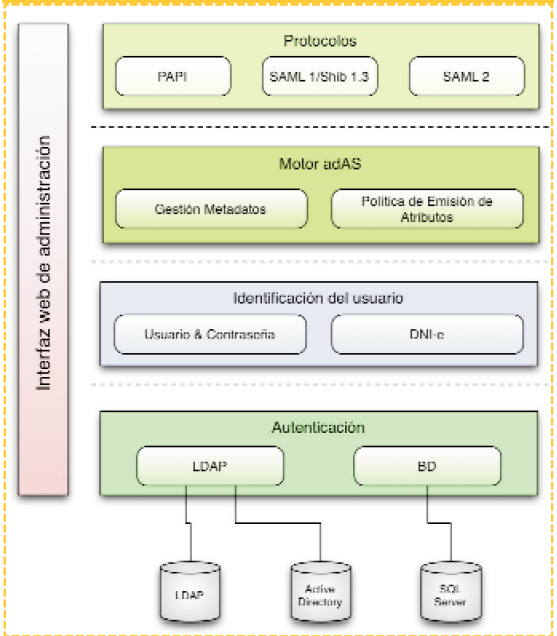
• **Métodos de autenticación:** con el par nombre de usuario/contraseña y con su DNI electrónico.

• **Fuentes de datos:** Inclusión de una o más fuentes de datos a la hora de autenticar u obtener atributos del usuario. De esta forma, ha sido posible ofrecer a los proveedores de servicio integrados con el adAS tanto la información del usuario que hay en el LDAP como la existente en otras bases de datos de Oracle o MySql.

• **Módulos de gestión:** Módulo de gestión de política de atributos que permite definir qué atributo se enviarán a cada uno de los recursos en los que se confía. Esta gestión es independiente del protocolo que utilicen dichos recursos. Además estos atributos pueden ser filtrados y modificados antes de ser emitidos a las aplicaciones. Y el módulo de gestión de metadatos, que permite incluir, modificar o eliminar proveedores de servicio de confianza, para cualquiera de los 3 protocolos implementados.

• **Interfaz gráfica de adAS:** Interfaz gráfica que permite la administración de todo el sistema de Single Sign-On a través de una aplicación web, facilitando y haciendo más amigable la gestión de un sistema que puede llegar a ser bastante complejo. Generación gráfica de informes sobre estadísticas de uso, que permite identificar el uso diario de la infraestructura y los potenciales ataques de phising o mal uso del entorno, de un modo visual.

FIGURA 1



Los atributos pueden ser filtrados y modificados antes de ser emitidos a las aplicaciones

El módulo de gestión de metadatos permite incluir, modificar o eliminar proveedores de servicio de confianza

4. adAS en funcionamiento

En un sistema Single Sign On el usuario debe superar dos fases para acceder a los recursos:

1.- **Autenticación:** el usuario es quien dice ser. Deberá presentar sus credenciales, bien usuario y contraseña, su DNI electrónico, el certificado de la FNMT o cualquier otro método implementado en el sistema.
2.- **Autorización**: dependiendo del valor de ciertos atributos del usuario se le concederá, o no, acceso a la aplicación o a ciertas partes de la misma.

En un sistema Single Sign On el usuario debe superar dos fases para acceder a los recursos: autenticación y autorización

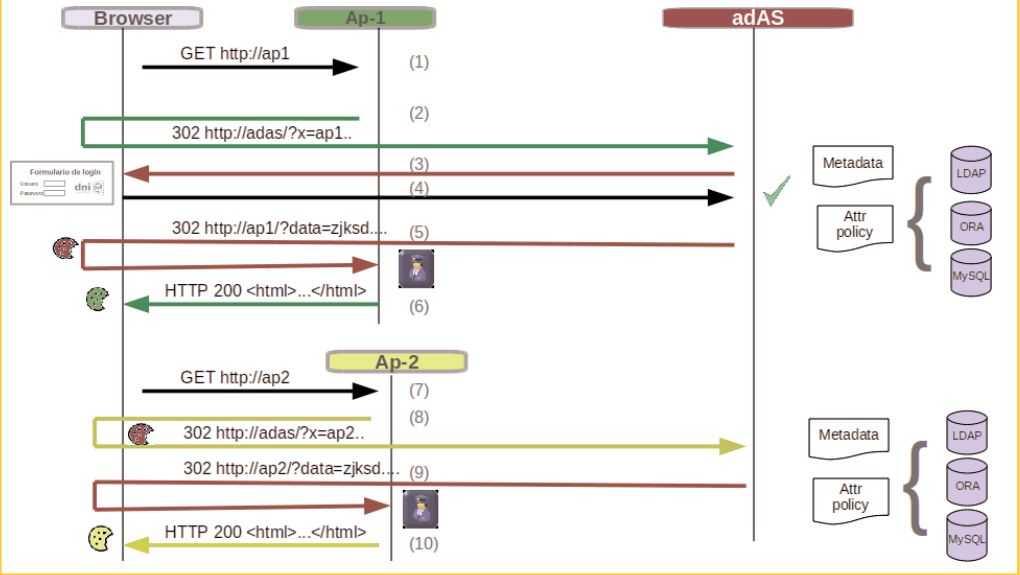
En los servidores que alojan las aplicaciones se instalará el conector adecuado, dependiendo de la tecnología. Por ejemplo, para aplicaciones desarrolladas en php utilizamos phpPoA, papiFilter; para aplicaciones en jsp, papiPoA; para proteger directorios desde Apache o simpleSAMLphp para aplicaciones estándar con plugins de autenticación, SAML. Existen más conectores y se están desarrollando nuevos.

Todas las comunicaciones entre adAS y las aplicaciones, se realizan a través del navegador del usuario. Los mensajes del protocolo de PAPI se transmiten utilizando métodos GET o POST sobre HTTP, mientras que, las respuestas serán redirecciones 302 de HTTP, incluyendo además cookies en el caso de que el usuario deba almacenar algún tipo de información útil para el protocolo. Todos estos flujos de información irán cifrados e implementarán todos los mecanismos de seguridad definidos en el protocolo PAPI.

Todas las comunicaciones entre adAS y las aplicaciones, se realizan a través del navegador del usuario

En el siguiente gráfico, se muestra de modo resumido el funcionamiento del sistema. Como escenario inicial tenemos un usuario que desea conectarse a la aplicación Ap-1 y a continuación durante la misma sesión, se quiere conectar a otra aplicación Ap-2, ambas integradas en adAS:

FIGURA 2



(1) El usuario realiza en su navegador la petición de la web de Ap-1.
(2) Ap-1, a través de una redirección 302, envía una petición de atributos del usuario a adAS.
(3) adAS comprueba que tiene los metadatos de Ap-1, y envía al usuario el formulario de login.
(4) El usuario entrega sus credenciales, adAS comprueba si son correctas.
(5) adAS recoge los atributos del usuarios de las diferentes fuentes de datos definidas, y a través de una redirección302 envía una cookie al navegador del usuario y entrega a Ap-1 los atributos que tiene definidos en la política.
(6) Ap-1 comprueba que el usuario cumple con las políticas de autorización, le envía una cookie y le permite entrar en la aplicación.
(7) El usuario intenta entrar en otra aplicación Ap-2 integrada en adAS.
(8) Ap-2, a través de una redirección 302, envía una petición de atributos del usuario a adAS.
(9) adAS comprueba que tiene los metadatos de Ap-2, también le ha llegado una cookie del usuario, comprueba que la validación continúa estando activa y cumple todas las políticas de seguridad, y a través de una redirección302 entrega a Ap-2 los atributos que tiene definidos en la política.
(10) Ap-2 comprueba que el usuario cumple con las políticas de autorización, le envía una cookie y le permite entrar en la aplicación.

Como se puede observar, la mayoría de estos procesos son transparentes para el usuario.

adAS se ha conectado con el Servicio de Identidad de RedIRIS (SIR) y la federación Fecyt

5. Resultados

Al soportar varios protocolos de identidad digital, permite la rápida integración de aplicaciones web ya desplegadas en la Universidad, puesto que en muchas de ellas existe un conector o plugin para integrarse con alguno de los protocolos anteriormente comentados.

Actualmente ya se han integrado aplicaciones como Wordpress, aplicaciones desarrolladas en la universidad con php y jsp, se ha conectado con el Servicio de Identidad de RedIRIS (SIR) y la federación Fecyt. Se han realizado pruebas de integración con Moodle, Drupal, Docnet, Mantis y aplicaciones desplegadas en un servidor de aplicaciones GlassFish.

Trabajos que se están desarrollando en estos momentos:

Se están desarrollando conectores para integrar servidores FTP

• Conector para integrar en adAS OracleSSO, para permitir conectar las aplicaciones de OCU y varias aplicaciones desplegadas en el Oracle Aplication Server.
• Nuevos métodos de autenticación: mediante el certificado FNMT, con otros certificados de smartcard como los carnes de estudiantes y a través de STORK.
• Conector de módulos PAM para la integración de los Webmailes.
• Sistemas de Single Log Out (SLO). El sistema de LOGOUT en la actualidad consiste en cerrar todas las ventanas del navegador.
Trabajos que se desarrollarán próximamente:
• Conectores para integración de los escritorios de las aulas de informática, desde equipos con Windows, Linux y Macintosh.
• Conectores para integrar servidores FTP.
Inconvenientes detectados:
• Habría que concienciar la usuario de que no debe abandonar el terminal con una sesión abierta ya que todos los servicios a los que tiene acceso podrán quedar comprometidos.
• Integrar aplicaciones que ya están en explotación supone, como cualquier cambio, un esfuerzo por parte de los administradores, por lo cual la implantación debe estar fuertemente respaldada por la dirección.

6. Conclusiones

Tras varios meses de trabajo conjunto entre la Universidad de Salamanca y la empresa PRiSE, podemos concluir que hemos obtenido un software de gran calidad, muy flexible y potente, fácil de desplegar y de gestionar.

Que cumple la mayoría de los objetivos propuestos inicialmente, exceptuando aquellos que dependen de la integración completa de todas las aplicaciones de la Universidad, proceso en el que estamos inmersos.

Además, adAS está teniendo una buena acogida en la comunidad científica y universitaria nacional, conectada a través de RedIRIS.

Hemos obtenido un software de gran calidad, muy flexible y potente, fácil de desplegar y de gestionar

La Universidad de Girona ha financiado el desarrollo del conector para aplicaciones .NET y en estos momentos ya existe un grupo de universidades e instituciones interesadas en desplegar adAS y dispuestas a financiar nuevos componentes y funcionalidades.

Referencias

[1] adAS, Advanced Authentication Server Single Sign On http://www.adas-sso.com/

[2] PRiSE http://www.prise.es/

[3] PAPI Protocolo y herramientas de autenticación y autorización http://www.papisoftware.net/

[4] simpleSAMLphp http://simplesamlphp.org/

[5] Shibboleth http://shibboleth.internet2.edu/

[6] SIR Servicio de Identidad de RedIRIS http://www.rediris.es/sir/

[7] Bravo, Inmaculada, et al. OpenLDAP Baseline Security Analyzer. Boletín RedIRIS 88-89 http://openldap-bsa.forja.rediris.es/

[8] Alonso, Jose María, et al. LDAP Injection Techniques. , 2008. ISBN 978-1-4244-2423-8. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4737330

Inmaculada Bravo García
(inma@usal.es)
Reyes Hernández Rodriguez
(reyes@usal.es)
Universidad de Salamanca