Punto de Acceso a Proveedores de Información. PAPIv2

Access Point to Information Providers. PAPIv2

J. López, J. A. Montenegro, D. Ray y F. Moya

Resumen

El trabajo expone las principales características de la versión dos del proyecto 'Punto de Acceso a Proveedores de Información, PAPI [1]'. Esta versión proporciona un nuevo diseño al sistema y su desarrollo en los lenguajes Java y C, mejorando: versatilidad de PAPI a la hora de adaptarse a los distintos sistemas de control de accesos, jerarquía de PoAs, diseño modular, portabilidad, protocolos y formatos de transmisión de datos, y la integración de PAPI con las tecnologías de vanguardia.

Palabras clave: Control de Accesos, Autorización, Autenticación, Certificación Digital, Autoridad de Atributos

Summary

This work presents the main features of Version 2 of 'Access Point to Information Providers, PAPI [1]' project. This version provides a new design of the system and the development in Java and C languages, improving: adaptability of PAPI to other access control systems, modularity, portability, transmission protocols and formats, Point of Access hierarchy, and PAPI integration with the new technologies.

Keywords: Access Control, Authorization, Authentication, Digital Certification, Attribute Authority

En PAPIv2 se ha desarrollado un nuevo mecanismo de plugins que permite al Servidor de Autenticación adaptarse de forma simple a los recursos y políticas de las organizaciones

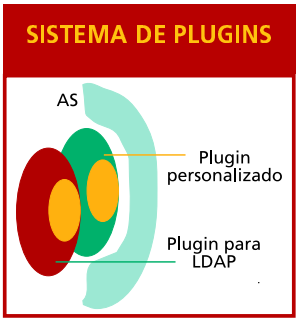
1.- Introducción

Hoy en día, los mecanismos de autenticación y autorización son un campo activo de estudio, tanto en ámbitos académicos, como en el desarrollo de aplicaciones de negocios. Las organizaciones y usuarios requieren tener el control sobre sus accesos a recursos electrónicos externos, y a su vez, los proveedores de estos recursos aplicar sus propias políticas.

El problema no radica únicamente en encontrar mecanismos fiables, sino también en la gran variedad de ellos en el mercado. PAPIv2 ofrece todas las características de sus versiones anteriores, brindando una solución genérica y fiable para el control de accesos en entornos heterogéneos y cambiantes. Acorde, por un lado con las políticas de las organizaciones, y por otro, con las exigencias de los proveedores de los recursos. Las principales mejoras de la versión dos son:

- Nuevo sistema de plugins en el AS, para autenticación y autorización de usuarios
- Refinamiento de la jerarquía de Puntos de Acceso (PoA)
- Nuevos protocolos y formatos de transmisión datos para la comunicación entre los componentes
- Mejora del diseño del sistema, aumentando su versatilidad en tareas de mantenimiento de software, administración y ampliaciones futuras.

2.- Sistema de plugins del Servidor de Autenticación



En esta versión de PAPI se ha desarrollado un nuevo mecanismo de plugins que permite al Servidor de Autenticación adaptarse de forma simple a los recursos y políticas de las organizaciones. Gracias al nuevo diseño del sistema de plugins, cambiar el modo de autenticación y autorización de usuarios, e incluso proveer uno nuevo, es una tarea sencilla y totalmente independiente del núcleo central del AS.

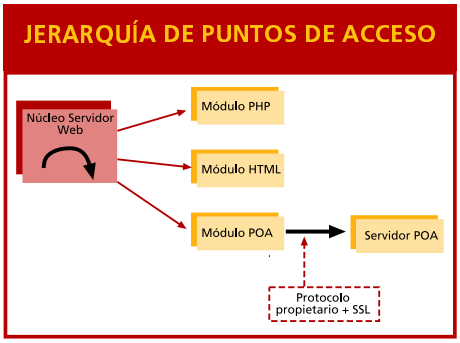
3.- Jerarquía de Puntos de Acceso

Se ha dividido la funcionalidad del Punto de Acceso en los siguientes niveles.

- **mPoA.** Se encarga de las tareas ligeras del PoA, es dependiente del servidor que controla el acceso al recurso, y proporciona una barrera entre el mundo exterior y la información a proteger. Su principal objetivo es la comprobación de uno de los tokens de acceso de PAPI, el lcooks. El mPoA se divide a su vez en: mPoA de primer nivel y mPoA de segundo nivel. Esta división permite independizar la funcionalidad propia del mPoA (mPoA nivel 2), de la comunicación con el servidor en concreto (mPoA nivel1). Ambos niveles residen en la misma máquina servidora del recurso y por ejemplo en el caso del servidor Apache, representan uno de sus módulos dinámicos.

- **sPoA.** Se encarga de las tareas pesadas del Punto de Acceso. Su situación natural es en máquinas distintas de las servidoras del recurso, de esa manera se aligera la carga del servidor y aumenta su rendimiento. El sPoA se encarga de: autorizar nuevas sesiones comprobando la autenticidad del AS y los privilegios del usuario, generar los tokens de acceso (lcook y hcook), comprobar sesiones en curso, logouts del sistema, y todo el resto de funcionalidades referentes al Punto de Acceso de PAPIv1.x.

- **gPoA.** En PAPIv2 se ha perfeccionado el concepto de gPoA, y al igual que en sus versiones anteriores, sirve de apoyo jerárquico a la hora de autorizar el acceso a un nuevo usuario. Los gPoAs pueden estar comunicados con Autoridades de Atributos que se encargan de proporcionar mayor información acerca del usuario que requiere acceso al sistema (en caso de que ésta fuese necesaria dependiendo de las políticas de seguridad del proveedor del recurso). No confundir el concepto de autoridad de atributos usado en este trabajo, con el de autoridad certificadora de atributos en el contexto de certificación digital.



La división que se ha presentado proporciona a PAPIv2 una gran capacidad de integración en todo tipo de máquinas servidoras y de control de accesos

La división que se ha presentado proporciona a PAPIv2 una gran capacidad de integración en todo tipo de máquinas servidoras y de control de accesos, bastando con implementar el pequeño módulo mPoA de primer nivel para adaptar el sistema a un nuevo tipo de máquina servidora.

4.- Protocolos de comunicación mPoA-sPoA

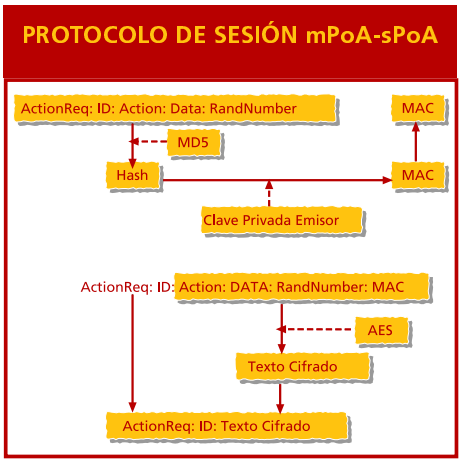
Dado que los mPoAs y los sPoAs se sitúan en máquinas distintas y que entre ellos se transmiten datos de vital importancia, tal como es la información personal de los usuarios, la comunicación entre mPoA y sPoA ha de estar protegida y proporcionar las siguientes propiedades: privacidad de contenidos, integridad de contenidos, y autenticidad de los extremos de la comunicación. La alternativa directa para conseguir estos objetivos es la apertura de un túnel SSL entre mPoA y sPoA, sin embargo, abrir y cerrar comunicaciones SSL es una operación costosa. Este tipo de situación hace que no sea conveniente mantener el canal seguro mediante SSL, ni de forma permanente, ni abriendo uno por cada comunicación mPoA-sPoA. Por esa razón, se divide el proceso de comunicación mPoA-sPoA en dos etapas: una de negociación de claves simétricas e identificadores de sesión, y otra para el resto de las comunicaciones, respectivamente: protocolo de configuración y protocolo de sesión.

4.1.- Protocolo de configuración mPoA-sPoA

El protocolo de configuración se ejecuta una vez por mPoA. En él se negocian las claves simétricas y el identificador usado en el protocolo de sesión. La negociación se lleva a cabo bajo SSL, autenticando ambos extremos, de esta forma se garantiza: la privacidad, integridad y autenticidad de la información transmitida. En el protocolo de configuración no se transmite ninguna información referente a usuarios, y se realiza durante la puesta en marcha del servidor.

4.2.- Protocolo de sesión mPoA-sPoA

El protocolo de sesión crea un canal seguro y eficiente para la transmisión de datos entre el mPoA y sPoA. En esta fase, el mPoA transmite información de vital importancia referente a los usuarios que intentan acceder al sistema. La protección de la información transmitida se consigue mediante un cifrado simétrico del contenido, AES, y el registro de contenido mediante MAC. AES proporciona a la comunicación la propiedad de confidencialidad, y MAC la de autenticidad e integridad.



Las características buscadas para el formato de transmisión de datos de PAPIv2 son: independencia del contenido, y adaptabilidad al canal de transmisión

5.- Formato de transmisión de datos

Las características buscadas para el formato de transmisión de datos de PAPIv2 son: independencia del contenido, y adaptabilidad al canal de transmisión.

Muchos formatos de transmisión de datos separan los diferentes valores transmitidos mediante delimitadores, que en la mayoría de los casos pueden llegar a estar dentro del conjunto de caracteres de la información. Esto puede crear situaciones ambiguas a la hora de la recuperación de los campos, como ejemplo: "Jose Gutierrez$Directivo$Caja Madrid$10000•:es". Aquí el delimitador del formato es el carácter Dólar '$', sin embargo, en una situación en la que el salario mensual estuviese en 'Dolares', se confundiría el dato transmitido '$', con el separador '$'. El siguiente problema radica en el soporte de datos del canal de transmisión, como ejemplo SMTP, que tan sólo soporta caracteres US-ASCII. Para aumentar la versatilidad de SMTP se creó MIME.

```
Nombre -> Diego Ray
Ocupación              -> Informático
Clave     -> 857?69+`ç'/   (Array de bytes codificados base64)

3:0.6.8:14.23.11:34.39.11.aplication/octet-stream.base64:nombreDiego
RayOcupacionInformaticoClave857?69+`ç'/
```

PAPIv2 debe de tener en cuenta estos dos problemas, ya que, por un lado transmite información binaria por canales que no soportan este tipo de datos, como es HTTP, y por otro, no es conveniente limitar el conjunto de caracteres transmitidos por el uso de delimitadores como los vistos en el ejemplo anterior.

Para solventar estos problemas, en PAPIv2 se ha creado un formato de transmisión de datos propio e independiente del medio y contenido transmitido. Toda la información de indexación y separación de campos se mantiene en una cabecera inicial, y se proporcionan funcionalidades para la codificación de los datos, por ejemplo, base 64.

6.- Conclusiones

Las organizaciones y los proveedores de recursos de información tienen sus propias políticas de seguridad, y ambos desean mantenerlas. Sin embargo, la necesidad de acceder a estos recursos ajenos al dominio de protección, es una actividad habitual y necesaria, por lo que las TI deben de proporcionar mecanismos para el control de accesos tales como el sistema PAPI, que se adapta a las distintas necesidades de ambos, en un ambiente cooperativo y extensible.

Las organizaciones y los proveedores de recursos de información tienen sus propias políticas de seguridad, y ambos desean mantenerlas

Referencias

[1]López, Diego; Castro, Rodrigo. PAPI: http://papi.rediris.es

Javier López,
(jlm@lcc.uma.es)
José A. Montenegro,
(monte@lcc.uma.es)
Diego Ray
(diegoray@lcc.uma.es)
Dpto. de Lenguajes y CC de la Computación
UMA
Fernando Moya
(fmoya@novasoft.es)
Novasoft S.A.