La seguridad en 'web services': entre la incertidumbre y la sobreinformación

Resulta difícil ojear revista alguna relacionada con el mundo de los sistemas de información sin encontrar un artículo que analice el concepto de Web *service*. Mucho más difícil es evitarlo en las estrategias de las grandes compañías. Partiendo de una tendencia con un par de años de antigüedad, y cumpliendo el viejo axioma de que "la seguridad siempre la podemos añadir *a posteriori*", no ha sido hasta hace pocos meses que, tanto los cuerpos y organismos de estandarización, como las grandes compañías, nos han abierto los ojos ante la necesidad de incorporar mecanismos de seguridad dentro de la arquitectura/modelo/paradigma que representa el concepto de Web s*ervice*. En el presente artículo intentaremos repasar dichas propuestas, contrastando los objetivos con el estado actual de implantación e intentando echar un vistazo a la bola de cristal, por muy turbia que ésta se encuentre. Pasen y vean.

Roberto López Navarro

¿Qué es un Web service?

Siempre es bueno disponer de una definición de trabajo. Como punto de partida adoptaremos la acepción propuesta por el World Wide Web Consortium [1] / [2]. Salvando la traducción:

1. Un Web service es un sistema software identificado por una URI [3] cuyos interfaces públicos están definidos y descritos mediante XML. Esta definición puede ser accedida por otros sistemas software, los cuales pueden interactuar con el Web service en la forma prescrita en su definición, utilizando mensajes XML y transportados por protocolos Internet.

2. Un conjunto de Puntos Finales (EndPoints).

Tal y como van los estándares, nos encontramos ante una definición estándar: independencia, neutralidad, extensibilidad... y que no ayuda mucho a comprender qué es un *Web service*. Sin embargo proporciona la información necesaria para intuir los aspectos fundamentales:

– sistemas software, *a priori* independientes, que interaccionan (no usuarios, aunque es contemplado en la infraestructura propuesta)

– mediante intercambio de mensajes XML (no ASN.1, OMG IDL, etc.) que se definen también en XML y son accesibles

– sobre protocolos de transporte típicos de Internet (http, smtp, etc...)

¿Por qué Web service?

La comunidad *Web service* está constituida por un conjunto dispar de evangelizadores, arquitectos, desarrolladores y fabricantes motivados por aspectos tales como:

– La idea de "objetos distribuidos" o "integración de aplicaciones", entendiendo la posibilidad de diseñar e implementar aplicaciones distribuidas.

– Los entornos EDI/B2B.

– La *World Wide Web*.

– La necesidad de encontrar una idea que permita aumentar el valor de las acciones.

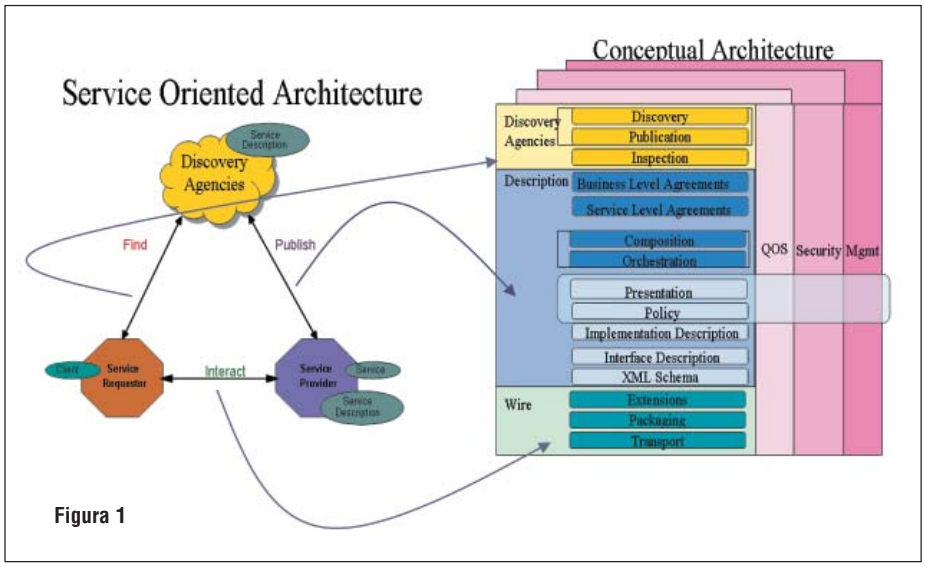

Figura 1

En el corazón de la bestia late la necesidad de cubrir la creciente demanda de integración entre aplicaciones, o mejor dicho, servicios proporcionados por organizaciones independientes, y en consecuencia, sin una plataforma tecnológica común. En realidad, es una revisión de una promesa eterna. Si recordamos iniciativas anteriores (RPC, Java RMI, Microsoft COM/DCOM, OMG CORBA, etc.) es fácil ver cómo cada una de ellas proporcionaban esta funcionalidad, con mayor o menor éxito.

El modelo propuesto en la arquitectura *Web service* [5] es el presentado en la **figura 1.**

El modelo contempla dos componentes básicos: el servicio (*Service*) y la descripción del mismo (*Service Description*); tres roles distintos: proveedor del servicio (*Service Provider*), solicitante de servicio (*Service Requestor*) y las agencias de publicación (*Discovery Agency*); finalmente se definen tres operaciones: Publicación (*Publish*), Búsqueda (*Find*) e Interacción (*Interact*). Una descripción de escenarios puede consultarse en [11].

Los solicitantes de servicio mediante una operación de búsqueda sobre las agencias de publicación obtienen una descripción del servicio proporcionado por un proveedor de servicio que les permitirá interaccionar con el mismo. Sencillo y limpio.

¿Qué aporta este modelo? En dos palabras, interoperabilidad.

XML, SOAP, WDSL... y más siglas

El principal requisito que guía la definición de la anterior infraestructura es la interoperabilidad [4]. El reto fundamental al que nos enfrentamos es garantizar un lenguaje y reglas de comunicación comunes. En el caso de los Web *services* esto se traduce en:

– la utilización de protocolos de transporte comunes y ampliamente implantados, tanto en Internet (http, smtp) como en entornos corporativos (MQSeries, CORBA, IIOP, etc.),

– que permitan intercambiar mensajes XML de acuerdo a un conjunto de reglas sencillo (*Simple Object Access Protocol* SOAP [6])

– según una sintaxis y un protocolo

descrito en la especificación XML del servicio (*Web Services Description Language* WSDL [7])

De esta forma se pretende proporcionar un mecanismo sencillo que permita la colaboración entre servicios, independientemente de la tecnología que da soporte a los mismos. Hasta aquí todo correcto. Pero siempre se quiere más.

La marca personal de los Web *services* se ha traducido en la creencia de que la tecnología por sí misma garantiza semánticas comunes en dominios no tecnológicos, es decir, los modelos de negocio. O dicho de otra forma: la capacidad de automatizar la generación, transmisión y procesado de un conjunto de mensajes XML de forma más o menos automática no implica inmediatamente que la semántica asociada a dichos mensajes sea la misma [8]. De ahí la explosión exponencial de modelos (es decir, siglas de tres o más letras) que intentan capturar dicha semántica y estandarizarla (ebXML [9] es un magnífico ejemplo).

Complejidad al poder. Aunque esa es otra historia que será contada en otra ocasión.

Pero este era un artículo de seguridad...

Sí, es un artículo sobre seguridad en *Web services*. ¿Por qué esta introducción? Porque, ahora que los profesionales de la seguridad disponemos de unos cuantos documentos sobre los que discutir, mi humilde opinión es que se está produciendo el mismo fenómeno de escalado de complejidad. Y a los documentos me remito.

Partamos de los objetivos de seguridad descritos por *World Wide Web Consortium* [1] en su documento de especificación de requisitos [4]. Salvando la traducción:

*AG004 Seguridad. La Arquitectura Web Service (*Web Service Architecture *o WSA) debe proporcionar un entorno seguro para los procesos en línea (*onlines processes*). Los factores críticos de éxito identificados son:*

— AC006 aborda la seguridad de los Web services *en entornos distribuidos, multi-dominio y multi-plataforma:*

– AC006.1 construcción de un modelo de amenazas para los Web services (Web services Threat Model) *basado en un análisis exhaustivo de las amenazas actuales y futuras a las que estarán sujetos tanto los Puntos Finales* (EndPoints) *como los canales de comunicación:*

– AC006.2 el establecimiento de un conjunto de políticas de seguridad para los Web services (Web services Security Policy) *que neutralice o mitigue las amenazas de seguridad identificadas por el modelo anterior*

*– AC006.3 la construcción de un modelo de seguridad (*Web services Security Model*) que integre las políticas de seguridad*

– AC006.4 la implantación del modelo de seguridad a través de un marco de referencia de seguridad para Web services (Web services Security Framework*) que forme parte integral de la arquitectura (WSA)*

— Requisitos:

– AR006.1 el marco de referencia de seguridad considerará la amenaza de ataques contra la accesibilidad/disponibilidad de servicio ([D]DOS, DNS spoofing*, etc.)*

– AR006.2.1 el marco de referencia de seguridad debe proporcionar mecanismos de autenticación a las partes que forman parte de la comunicación

*– AR006.2.2 el marco de referencia de seguridad debe proporcionar mecanismos de autenticación de origen de datos (*authentication of authorship of data*) tanto persistente como transitoria* (transient authentication*)*

– AR006.3 el marco de referencia de seguridad debe proporcionar mecanismos de autorización

– AR006.4 el marco de referencia de seguridad debe proporcionar mecanismos de confidencialidad.

– AR006.5 el marco de referencia de seguridad debe proporcionar mecanismos de integridad de datos

– AR006.6 el marco de referencia de seguridad debe proporcionar mecanismos de no repudio en origen y recepción

– AR006.10.1 el marco de referencia de seguridad debe proporcionar mecanismos para expresar la política de seguridad

– AR006.10.2 el marco de referencia de seguridad debe proporcionar mecanismos que permitan acceder a la política de seguridad definida por un Web service

– AR006.12 el marco de referencia de seguridad debe proporcionar mecanismos de auditoría/trazabilidad (auditing)

– AR006.13 cuando el Web service proporcione mecanismos de seguridad acordes a lo expresado en el punto AR006, proporcionará medios que habiliten la administración de dichos mecanismos

—AC020 aborda aspectos relacionados con la protección de la privacidad para los consumidores de un Web service *en entornos distribuidos (multi-dominio y multi-plataforma).*

—Requisitos:

– AR020.1 la Arquitectura de Web Services *(*Web Service Architecture *o WSA) debe proporcionar mecanismos que permitan expresar políticas de seguridad concernientes a la privacidad de datos*

– AR020.2 la política de privacidad de un Web service *debe ser expresada en P3P [26]*

– AR020.3 la WSA debe permitir a un consumidor acceder a las políticas de privacidad publicadas por un Web service

– AR020.5 la WSA debe permitir la delegación y diseminación de políticas de privacidad

– AR020.6: la operación de un Web service *debe contemplar la posibilidad del anonimato de una o varias partes que participen en la comunicación*

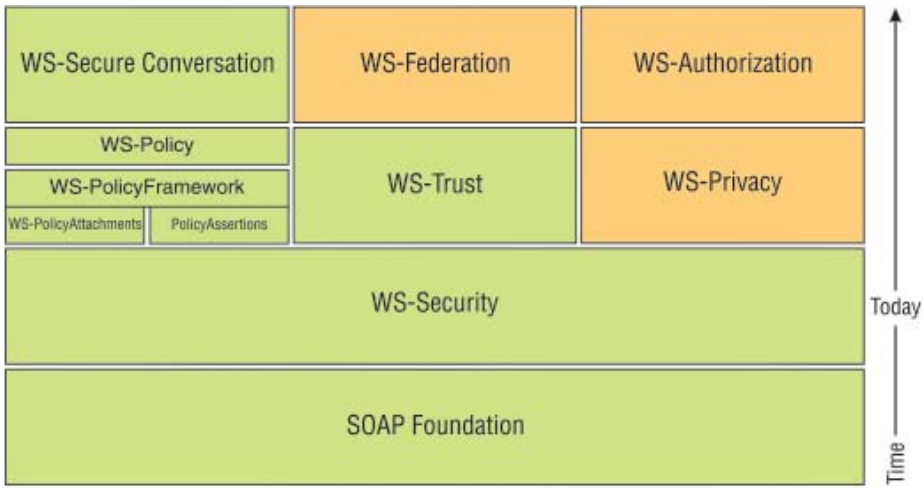

Figura 2

Parte de estos requisitos adolecen de precisión y ponen de manifiesto la falta de consenso en algunos aspectos claves [10]. Lo que queda claro, a la vista de los requisitos, es que la amplitud de miras y la ambición de la propuesta están demostradas.

Un posible modelo de seguridad...

La administración y gestión segura de la identidad digital ha sido el primero de los temas que ha saltado del papel al desarrollo. La controversia, basada fundamentalmente en una cruzada (otra más) entre Microsoft (.Net+Passport [12]) y Sun Microsystems (Sun One + Liberty Alliance [13]), queda lejos de estar resuelta, y plantea dudas respecto a qué es más importan-

te: cumplir un requisito de seguridad o ganar mercado. Partiendo de que ambas aproximaciones son igualmente válidas y seguras, parece que una arquitectura basada en la idea de federación de servicios es acorde con la idea de federación de identidad digital. Actualmente, y como suele ser habitual, la solución quedará en un punto intermedio.

Dejando a un lado este aspecto, la única iniciativa que ha removido los bajos fondos de la seguridad ha sido la propuesta presentada, hace ya casi un año, conjuntamente por Microsoft e IBM [14]. Actualmente la iniciativa es coordinada por OASIS [18].

Dicha propuesta describe una estrategia para abordar los objetivos y requisitos de seguridad de la arquitectura *Web service,* definiendo un modelo de seguridad que pretende soportar, integrar y unificar diferentes modelos de seguridad (sic), mecanismos y tecnologías de amplia aplicación (Kerberos, X.509, etc...). La **figura 2** representa el modelo propuesto. Se incluye la última revisión propuesta.

WS-Security [17] describe cómo aplicar técnicas de firma digital y cifrado a los mensajes SOAP mediante la utilización de propuestas XML pioneras: *XMLdsig* [15] y *XMLenc* [16].

WS-Policy [19], establece cómo los solicitantes y proveedores del servicio pueden especificar requisitos que determinen cómo debe llevarse a cabo la comunicación. Estos requisitos en su mayoría hacen referencia a políticas de seguridad aplicables (mecanismos de autenticación, algoritmos criptográficos, longitudes de clave, tratamiento de información personal, etc....) pero se incorpora un mecanismos de extensibilidad para incluir otro tipo de requisitos no directamente relacionados con la seguridad.

WS-Trust [20] especifica mecanismos para establecer diferentes modelos de confianza, incluyendo aspectos tales como la impersonación y la delegación.

WS-Privacy permitirá (durante la redacción del presente artículo no se ha encontrado documentación específica) definir y especificar requisitos relacionados con el tratamiento de la información personal, partiendo del marco descrito por *WS-Policy*.

WS-SecureConversation [22] proporcionará mecanismos para contextos de seguridad para el establecimiento de comunicaciones autenticadas. El contexto de seguridad comprenderá aspectos tales como el establecimiento de claves de sesión, claves derivadas, algoritmos, etc. Cabría esperar que en su definición final influyeran algunas propuestas XML pioneras: XKMS [23].

WS-Federation abundará (durante la redacción del presente artículo no se ha encontrado documentación específica) sobre aspectos relacionados con la definición y creación de escenarios y modelos de confianza basados en el concepto de federación, inicialmente orientándolo hacia la integración de Kerberos y PKI (no podemos olvidar a los autores de la estrategia).

Finalmente *WS-Authorization* abordará (durante la redacción del presente artículo no se ha encontrado documentación específica) cómo deberán especificarse las políticas de acceso a un Web *service*. En este sentido, cabe esperar cómo se incorporan aspectos como SAML o XACML [25]

Opiniones y conclusiones

Tal como se planteaba en la introducción, de pronto nos encontramos ante una verdadera avalancha de recomendaciones, iniciativas y estándares que tienen como misión proporcionar seguridad. O más bien, un entorno seguro y confiable, basado en la federación de servicios, en el cual las relaciones de confianza, las políticas de seguridad, la identidad digital, los procesos de negocio y los beneficios se gestionarán, administrarán y operarán de forma automática. Por supuesto, el cambio de paradigma supondrá el impulso final a la economía digital.

Seamos serios. Si el presente artículo incluye 25 referencias (y cada referencia a su vez incluye unas cuantas docenas de siglas) es para demostrar que el escenario en el que nos encontramos se caracteriza por dos factores: incertidumbre y sobreinformación. O dicho de otra forma, corren tiempos arriesgados (eso ya lo sabían, pero no hablaba de cifras económicas).

La gran oportunidad estará en aquellas compañías y/o profesionales que sean capaces de distinguir cuál es la iniciativa que saldrá victoriosa en los *lobbies* de estandarización.

Por otro lado, es excitante ver como viejas tecnologías (recuerden que hablamos de Kerberos, X.509, etc.) encuentran sitio en las últimas tendencias. Mala hierba nunca muere, y en seguridad, aún menos. ■

Roberto López Navarro
Jefe de la División de Identidad Digital
SGI SOLUCIONES GLOBALES INTERNET
rolopez@sgi.es

REFERENCIAS

[1] <http://www.w3.org>

[2] Web Services Glossary W3C Working Draft 14 November 2002 (<http://www.w3.org/TR/2002/WD-ws-gloss-20021114/>)

[3] Uniform Resource Identifiers (URI): Generic Syntax, IETF RFC 2396, T. Berners-Lee, R. Fielding, L. Masinter, August 1998 (See <http://www.ietf.org/rfc/rfc2396.txt>.)

[4] Web Services Architecture Requirements W3C Working Draft 14 November 2002 (<http://www.w3.org/TR/2002/WD-wsa-reqs-20021114>)

[5] Web Services Architecture W3C Working Draft 14 November 2002 (<http://www.w3.org/TR/2002/WD-ws-arch-20021114/>)

[6] SOAP Version 1.2 Part 1: Messaging Framework W3C Candidate Recommendation 19 December 2002 (<http://www.w3.org/TR/2002/CR-soap12-part1-20021219>)

[7] Web Services Description Language (WSDL) Version 1.2 W3C Working Draft 3 March 2003 (<http://www.w3.org/TR/2003/WD-wsdl12-20030303>)

[8] Web Services: It's So Crazy, It Just Might Not Work, Clay Shirky October 03, 2001 (<http://webservices.xml.com/pub/a/ws/2001/10/03/webservices.html>)

[9] <http://www.ebxml.org/>

[10] Web Services Architecture WG Issues Document Last update: 03-Feb-2003 (<http://www.w3.org/2002/ws/arch/2/issues/wsa-issues.html>)

[11] Web Services Architecture Usage Scenarios W3C Working Draft 30 July 2002 (<http://www.w3.org/TR/2002/WD-ws-arch-scenarios-20020730/>)

[12] <http://www.passport.net>

[13] <http://www.projectliberty.org/>

[14] Security in a Web Services World: A Proposed Architecture and Roadmap, April 7, 2002 Version 1.0 (<http://msdn.microsoft.com>, <http://www-106.ibm.com>)

[15] XML-Signature Syntax and Processing W3C Recommendation 12 February 2002 (<http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>) (http://www.ietf.org/rfc/rfc3275.txt)

[16] XML Encryption Syntax and Processing W3C Recommendation 10 December 2002 (http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/)

[17] Web Services Security (WS-Security) Version 1.0 05 April 2002 (<http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>)

[18] <http://www.oasis-open.org/>

[19] Web Services Policy Framework (WSPolicy) Version 1,0 December 18, 2002 (<http://www-106.ibm.com/developerworks/library/ws-polfram/>)

[20] Web Services Trust Language (WS-Trust) Version 1.0 December 18, 2002 (<http://www-106.ibm.com/developerworks/library/ws-trust/>)

[21]

[22] Web Services Secure Conversation Language (WS-SecureConversation) Version 1.0 December 18, 2002 (http://www-106.ibm.com/developerworks/library/ws-secon/)

[23] XML Key Management Specification (XKMS) W3C Note 30 March 2001 (<http://www.w3.org/TR/2001/NOTE-xkms-20010330/>)

[24] <http://www.oasis-open.org/committees/security/>

[25] <http://www.oasis-open.org/committees/xacml/index.shtml>

[26] <http://www.w3.org/P3P/>