Apéndice I

Marco Técnico

Introducción

En esta sección se brindan algunos conceptos necesarios para la comprensión de este documento, concretamente en los temas Seguridad, Web Services y Arquitecturas Orientadas a Servicios.

Seguridad

Conceptos Básicos

Autenticación

Es el proceso de verificar una identidad [1]. Existen tres tipos principales de evidencias que una entidad puede presentar para probar su identidad [2]:

- Algo que conoce (nombre de usuario, contraseña, respuestas a preguntas, etc.)
- Algo que tiene (*token* físico, etc.)
- Algo que es (huella dactilar, escaneo de retina, etc.)

Confidencialidad de datos

Es la propiedad de que la información sensible no es revelada a individuos, entidades o procesos no autorizados. [3]

Los datos intercambiados a través de una red deben ser protegidos para evitar que usuarios no autorizados puedan tener acceso a los mismos. La técnica estándar para asegurar la confidencialidad de los datos intercambiados es la encriptación. [2]

Integridad de datos

Es la propiedad de que los datos no han sido alterados de manera no autorizada. La integridad de datos refiere tanto a los datos que están almacenados, o que están siendo procesados o transmitidos. [3]

No Repudio

Esta característica permite asegurar que el emisor no puede repudiar, o negar, que ha enviado determinada información o mensaje. [2]

Autorización

Una vez que un usuario ha sido autenticado, una aplicación necesita determinar si está autorizado a acceder a la funcionalidad que está solicitando. A la Autorización se le conoce comúnmente también como Control de Acceso. La decisión de otorgar acceso puede depender de múltiples criterios, como la acción solicitada, el recurso sobre el que se aplica y los grupos o roles a los que pertenece el usuario. [2]

Roled Based Access Control (RBAC)

Control de Acceso Basado en Roles (Role Based Access Control, RBAC) [4][5] es un modelo para el control de acceso en el cual los permisos son asociados a roles y los usuarios adquieren permisos al pertenecer a esos roles. La principal motivación de RBAC es facilitar las tareas administrativas.

Claims y Token de Seguridad

Una *claim* es una afirmación acerca de un sujeto hecha por él o por otro sujeto. Toda *claim* tiene un tipo y un valor. Por ejemplo, el tipo de una *claim* puede ser “Nombre” y el valor “Juan”. Las *claims* se empaquetan en *tokens* de seguridad que son distribuidos por su emisor. [1]

Criptografía

La criptografía es la disciplina que engloba los principios, medios y técnicas para la transformación de datos con el fin de ocultar su contenido, impedir su modificación sin ser detectada e impedir su uso no autorizado. [6]

La criptografía permite cifrar (encriptar) y descifrar (desencriptar) mensajes, permitiendo que sólo usuarios autorizados puedan leerlos. La encriptación es el proceso en el cual el mensaje original (texto plano) se transforma en un mensaje codificado (texto cifrado o encriptado) a través de un algoritmo de codificación y una clave de encriptado. [7]

En la criptografía simétrica, o de clave secreta, tanto emisor como receptor utilizan la misma clave. Sin embargo, en la criptografía asimétrica, o de clave pública, se utiliza una pareja de claves (pública y privada). La clave pública se distribuye a cualquier usuario que la requiera, mientras que la clave privada debe conocerla únicamente su dueño.

Firma Digital

Una Firma Digital es una Firma Electrónica que permite validar la identidad del emisor de una transmisión digital y asegurar que la información transmitida no ha sido alterada por personas no autorizadas [9]. Las firmas digitales están basadas en criptografía de clave pública.

Certificados Digitales

Un Certificado Digital es una tecnología similar a una tarjeta de identificación, que certifica la identidad y autenticidad de su dueño y por lo tanto puede ser utilizado para verificar la identidad de una entidad (individuo, servicio, sistema, compañía, etc.) con la cual se está estableciendo una comunicación electrónica. [9]

Los certificados son emitidos y firmados digitalmente por Autoridades Certificadoras (CAs). Además de los datos de identificación y una copia de la clave pública del dueño, el certificado contiene un número serial, la identidad y firma digital de la CA emisora y la fecha de expiración.

Infraestructura de Clave Pública (Public Key Infraestructure, PKI)

Infraestructura que permite la utilización de criptografía de clave pública en un entorno corporativo o público. Consiste esencialmente en un grupo de servicios que posibilita la generación, almacenamiento y transmisión segura de parejas de claves.

Estos servicios coordinados generalmente incluyen una autoridad certificadora de confianza (CA), una autoridad de registro para aceptar pedidos de certificados digitales, un almacén de certificados en donde los usuarios pueden acceder a las claves públicas de otros usuarios, un certificado digital y un sistema para generar, almacenar y transmitir de forma segura certificados y parejas de claves a las entidades que los solicitan. [9]

Public Key Cryptography Standards (PCKS)

Public Key Cryptography Standards (PCKS) es un conjunto de estándares para criptografía de clave pública. Fueron desarrollados por un consorcio de la industria liderados por RSA Laboratories y especifican cómo un sistema criptográfico de clave pública debería ser implementado y operado. [9]

PKCS #10 (Certification Request Syntax Standard)

Define una sintaxis para solicitar certificados digitales. El pedido es un archivo de texto con un formato definido por el estándar, codificado en base 64.

PKCS #12 (Personal Information Exchange Syntax Standard)

Especifica un formato portable para almacenar y transportar certificados y claves privadas, entre otros datos.

Secure Socket Layer

Secure Sockets Layer (SSL) es un protocolo de seguridad a nivel de capa de transporte utilizado en Internet para asegurar comunicaciones. Provee autenticación, confidencialidad e integridad. [9]

SSL emplea encriptación de clave pública para asegurar la autenticación y de clave simétrica para la encriptación de información transmitida. La integridad del mensaje es garantizada al incorporar un mecanismo de chequeo de integridad llamado Código de Autenticación de Mensaje (Message Autenthication Code, MAC)

HTTPS

Hypertext Transfer Protocol Secure (HTTPS) es una versión segura del protocolo HTTP. Es esencialmente una combinación de HTTP y el protocolo SSL. [9]

HTTPS está basado en el sistema criptográfico de clave pública y permite cifrar información transmitida a través de Internet. Para ejecutar HTTPS en un servidor Web, es necesario instalar un certificado digital en el mismo.

Servicios de Directorio LDAP

Lightweight Directory Access (LDAP) es un protocolo estándar para acceder a información almacenada en un directorio. Un cliente se puede conectar a un servicio de directorio compatible con LDAP (o X.500) para agregar, eliminar, modificar y buscar información, siempre y cuando tenga los suficientes privilegios para hacerlo. [9]

LDAP fue desarrollado por la IETF y la versión LDAPv3 está definida en el RFC 2251. Está diseñado para ejecutar sobre el *stack* de TCP/IP operando como un protocolo cliente/servidor.

Security Assertion Markup Language (SAML)

SAML [8] define un *framework* para intercambiar información de autenticación y autorización entre dominios de seguridad[1] en la forma de *assertions*, en lugar de utilizar identidades o *tokens* de seguridad tradicionales. [2]

Una *assertion* es una sentencia de código que un servicio SAML crea en respuesta al término exitoso de un evento. Los tipos más importantes de *assertions* son:

- *Assertion* de Autenticación: El sujeto en cuestión fue autenticado de alguna forma, en determinado momento
- *Assertion* de Atributo: El sujeto en cuestión posee información adicional de atributos que puede ser útil para un pedido
- *Assertion* de Decisión de Autorización: El sujeto en cuestión debería tener acceso a un recurso especificado

Firewalls

Un *firewall* es una infraestructura de seguridad, ubicada entre redes para separar y proteger de forma lógica la privacidad e integridad de las comunicaciones entre las mismas, y como protección contra usos maliciosos. Los *firewalls* examinan el tráfico de mensajes que entran y salen de una organización, bloqueando el acceso de los mensajes no autorizados de acuerdo a las reglas de acceso definidas en ellos. [7]

En particular, los Firewalls XML operan en la capa de aplicación o sobre ella (en el stack TCP/IP tradicional) e inspeccionan el contenido XML antes que los mensagen pasen a la aplicación o clientes. Están diseñados para abordar ataques comunes que pueden ser transportados vía XML. [10]

[1] Un dominio de seguridad es una colección de sistemas que tienen una política de seguridad única y un sólo conjunto de controles de seguridad.

Arquitecturas Orientadas a Servicios

La Computación Orientada a Servicios (Service Oriented Computing, SOC) es un paradigma de computación que utiliza servicios como elementos fundamentales para dar soporte al desarrollo rápido, y de bajo costo, de aplicaciones distribuidas en ambientes heterogéneos. [11]

Los Servicios son entidades de software autónomas, auto-contenidas e independientes de la plataforma. Los Servicios proveen funcionalidades de negocio, tienen una interfaz pública y pueden ser descubiertos, invocados y combinados de forma dinámica.

La puesta en práctica del paradigma SOC requiere la implementación de Arquitecturas Orientadas a Servicios (Service Oriented Architectures, SOAs). Una SOA es una forma lógica de diseñar un Sistema de Software para proveer servicios, a usuarios finales, aplicaciones u otros servicios, a través de interfaces públicas que pueden ser descubiertas. Para esto es necesario contar con tecnologías de middleware que permitan el descubrimiento, utilización y combinación de servicios interoperables, para dar soporte a los procesos de negocio de las empresas. [11][12]

Los tres roles principales que pueden encontrarse en una SOA son: Proveedor de Servicios, Registro de Servicios y Consumidor de Servicios.

El Consumidor de Servicios (también denominado cliente) es una aplicación, módulo de software, o servicio, que busca servicios en el registro y ejecuta las funcionalidades de negocio que estos proveen. Por otro lado, el Proveedor de Servicios es una entidad, accesible en la red, que acepta y ejecuta pedidos de los consumidores. El proveedor publica sus servicios en el registro para que los consumidores puedan descubrirlos e invocarlos. Por último, el Registro de Servicios brinda mecanismos para que los consumidores interesados puedan buscar y descubrir servicios. [13]

Las SOAs facilitan muchas de las tareas del desarrollo de aplicaciones empresariales distribuidas, como su integración, la implementación de procesos de negocios y el aprovechamiento de sistemas legados. Además, las SOAs proveen la flexibilidad y agilidad que requieren los usuarios de negocio, permitiéndoles definir servicios de alta granularidad que pueden ser combinados y reutilizados para abordar necesidades de negocio actuales y futuras.

Web Services

Un Web Service es una aplicación de *software* identificada por una URI, cuyas interfaces y formas de acceso pueden ser definidas, descriptas y descubiertas como artefactos XML, y soporta la interacción directa con otros componentes de software utilizando mensajes basados en XML que son intercambiados a través de protocolos basados en internet. [14]

Actualmente, los Web Services son el principal mecanismo de integración de aplicaciones tecnológicamente heterogéneas, y constituye la tecnología preferida para la implementación de SOAs.

Los Web Services se apoyan en un conjunto de estándares, varios de los cuales se describen en esta sección.

Estándares Básicos: SOAP, WSDL y UDDI

La tecnología de Web Services estuvo inicialmente basada en tres estándares principales: Simple Object Access Protocol (SOAP) [15], Web Services Description Language (WSDL) [16] y Universal Description Discovery and Integration (UDDI) [17].

Simple Object Access Protocol (SOAP)

SOAP es una especificación de la W3C que define un protocolo de comunicación basado en XML para intercambiar mensajes entre computadores, sin importar su sistema operativo o entorno de programación. SOAP es actualmente el estándar de facto utilizado por los Web Services. [7]

SOAP especifica un formato de mensaje, en el cual los mensajes son definidos como SOAP *Envelopes* y están compuestos por un conjunto de cabezales (*header*) y un cuerpo (*body*). Los cabezales tienen como propósito alojar información referente a infraestructura y direccionamiento, mientras que el cuerpo del mensaje aloja información específica del negocio.

Web Services Description Language (WSDL)

WSDL es una especificación de la W3C que define un lenguaje basado en XML para describir de forma estándar a un Web Service. Esta descripción puede incluir las operaciones que ofrece el Web Service, los protocolos de mensajería XML soportados, información de tipos de datos para los mensajes, información acerca del transporte específico a utilizar e información de direccionamiento para localizar al Web Service. [7]

Los documentos WSDL se componen de dos grandes partes: una descripción abstracta y una descripción concreta. La descripción abstracta representa la interfaz del Web Service y no hace referencia a tecnología o protocolo. Por otro lado, la descripción concreta consiste en la definición de protocolos de comunicación y transporte que permiten la ejecución de las operaciones definidas en la interfaz del servicio.

La principal ventaja de esta separación, abstracta y concreta, es la posibilidad de definir la interfaz de un servicio con sus operaciones y mensajes de forma independiente de los protocolos de transporte, ubicaciones y tecnologías, los cuales podrían cambiar más frecuentemente a lo largo del tiempo.

Universal Description, Discovery and Integration (UDDI)

UDDI (Universal Description, Discovery and Integration) es una especificación de la OASIS que tiene como propósito principal, facilitar la categorización, descubrimiento y recuperación de Web Services. Los registros UDDI permiten categorizar los servicios de acuerdo al tipo de negocio, al tipo de servicio o a las relaciones que tienen con otros servicios. A su vez, define una API basada en Web Services que permite la búsqueda, recuperación y publicación de servicios.

WS-Addressing

WS-Addressing [18] es un conjunto de especificaciones recomendadas por la W3C, que definen un mecanismo independiente del medio de transporte para referenciar Web Services y "direccionar" mensajes. Dentro de este conjunto de especificaciones se encuentran: WS-Addressing - Core, WS-Addressing - SOAP Binding y WS-Addressing - Metadata.

WS-Addressing Core es la principal de las especificaciones ya que define qué es un *Endpoint Reference* (EPR), qué son los *Messages Headers* y cómo pueden ser usados para referenciar Web Services o ser útiles para el direccionamiento de mensajes. Una de las principales características de esta especificación es que sus definiciones son totalmente abstractas e independientes del medio de transporte, por lo cual es posible utilizarlas en conjunto con HTTP, SMTP o cualquier otro protocolo.

Un EPR está compuesto por una dirección, parámetros secundarios y *metadata*. Por otro lado, los *Message Headers* o *Message Addressing Properties* son un conjunto de propiedades que permiten el

direccionamiento de mensajes dando soporte a varios patrones de comunicación para Web Services. Algunas de las propiedades definidas por WS-Addressing Core son:

- To: Es una IRI absoluta que representa la dirección del receptor del mensaje.
- From: Es un EPR que indica quién envió el mensaje.
- ReplyTo: Es un EPR que indica a dónde se deben enviar las respuestas del mensaje.
- FaultTo: Es un EPR que indica a dónde se deben enviar los errores relacionados al mensaje enviado.
- Action: Es una IRI absoluta que identifica de forma unívoca la semántica del mensaje.
- MessageID: Es una IRI absoluta que identifica de forma unívoca el mensaje.

Por otro lado, la especificación WS-Addresssing – SOAP Binding define cómo utilizar las propiedades, definidas de forma abstracta en el Core, en mensajes SOAP.

Por último, WS-Addressing – Metadata, indica cómo describir las propiedades utilizando un WSDL, cómo incluir *metadata* del WSDL en un *Endpoint Reference* y cómo utilizar WS-Policy para indicar que un Web Service soporta WS-Addressing.

WS-Security

WS-Security [23] es un estándar de la OASIS que propone un conjunto estándar de extensiones a SOAP que pueden ser utilizadas para enviar *tokens* de seguridad como partes del mensaje SOAP y proveer integridad y confidencialidad del contenido de los mismos [7]. WS-Security describe principalmente cómo incluir *tokens* de seguridad en mensajes SOAP y cómo utilizar XML Encryption y XML Signature para cifrar y firmar esos *tokens*, así como otras partes del mensaje.

WS-Security define un conjunto de *security tokens* y el mecanismo para adjuntarlos, identificarlos y referenciarlos dentro de un mensaje. El *UserNameToken*, por ejemplo, permite especificar un nombre de usuario y opcionalmente una contraseña. A su vez, el *BinarySecurityToken* permite incluir certificados X.509 o tickets Kerberos. Finalmente, a través de los *XMLTokens*, es posible adjuntar *security tokens* basados en XML utilizando distintos formatos como Security Assertion Markup Language (SAML) y eXtensible Rights Markup Language (XrML).

WS-Security provee un modelo de seguridad extensible que brinda soporte a múltiples *tokens* de seguridad, dominios de confianza, firmas digitales y algoritmos de encriptación, que junto con un modelo de seguridad específico (PKI, Kerberos, SAML, etc), permite lograr una solución completa y segura de punta a punta.

WS-Security versus SSL

Dado que HTTP es el principal medio utilizado para el transporte de mensajes SOAP, el uso de HTTPS (HTTP sobre SSL) es una clara alternativa para proveer propiedades de seguridad al intercambio de dichos mensajes. Sin embargo, SSL por sí sólo no permite proveer la protección necesaria para algunos escenarios de uso de Web Services.

Concretamente, la tecnología de Web Services prevé la existencia de intermediarios, que se sitúan entre el cliente y el Web Service destino. Los intermediarios pueden realizar tareas de ruteo, auditoría, transformación y validación de mensajes. SSL protege el mensaje únicamente cuando está siendo transmitido, pero no cuando llega a la capa de aplicación. De esta forma, un servicio o aplicación intermediaria podría examinar o modificar el mensaje antes de transmitirlo al destinatario final. Por otro lado, SSL no permite cifrar sólo partes del mensaje SOAP, como lo permite WS-Security.

WS-Trust

WS-Trust [24] es un estándar de la OASIS que extiende WS-Security y define un modelo de seguridad para Web Services basado en relaciones de confianza y mecanismos de emisión, renovación y validación de *tokens* de seguridad.

El modelo de seguridad definido por WS-Trust se basa en un proceso por el cual los servicios requieren que todo mensaje recibido contenga un conjunto de *claims* para ser aceptado. Todo mensaje que no las tenga, es rechazado. Los servicios generalmente utilizan WS-Policy para especificar qué *claims* requieren.

En casos en los cuales los clientes no tengan posesión de las *claims* para consumir un servicio, pueden contactar a autoridades apropiadas para así obtenerlas. Dentro de la especificación WS-Trust, se conoce a estas autoridades como los *Security Token Services* (STS). Los clientes pueden contactar a los STS para que éstos les emitan los *claims* y así poder consumir el servicio. En determinados casos puede ocurrir que los STS tengan definidas políticas de seguridad, siendo necesario presentar un conjunto de *claims* ante ellos.

Enterprise Service Bus

Un Enterprise Service Bus (ESB) es una plataforma de integración basada en estándares que combina mensajería, Web Services, transformaciones de datos y ruteo inteligente para conectar y coordinar, de forma confiable, la interacción de diversas aplicaciones con integridad transaccional. **¡Error! No se encuentra el origen de la referencia.**

Si bien la tecnología de Web Services constituye una base sólida para la implementación de SOAs, existen algunos aspectos, como su naturaleza punto a punto, que pueden afectar la flexibilidad y escalabilidad de soluciones que usan exclusivamente esta tecnología.

En este contexto, el ESB brinda una capa de integración intermedia que provee lógica de integración y comunicación reutilizable, para posibilitar la interacción entre clientes y servicios en una SOA. El ESB acepta pedidos en la forma de mensajes, sobre los cuales se pueden realizar distintas operaciones de mediación [20][21] para solucionar heterogeneidades entre clientes y servicios. A modo de ejemplo, el ESB puede interactuar con servicios a través de protocolos de comunicación no soportados por los clientes, o puede transformar los mensajes enviados por el cliente para que se ajusten al formato que espera el servicio.

La utilización de un ESB promueve el bajo acoplamiento entre clientes y servicios, dividiendo la lógica de comunicación e integración en pequeñas piezas de software administrables. Permite además tener una clara separación entre la lógica de integración y comunicación, y la lógica de negocio implementada por los servicios.

Las principales funcionalidades que generalmente proveen los productos de tipo ESB son: la transformación de protocolos de comunicación, la transformación de mensajes, el ruteo de mensajes, el enriquecimiento de mensajes y mecanismos de mensajería confiable. Asimismo, en general también cuentan con funcionalidades que permiten garantizar propiedades de calidad de servicio, como performance, disponibilidad y seguridad.[19][22]**¡Error! No se encuentra el origen de la referencia.**

Referencias

[1] D. Baier, V. Bertocci, K. Brown, M. Woloski, and E. Pace, A Guide to Claims-Based Identity and Access Control, Microsoft Press, 2010.

[2] Ramarao Kanneganti, Prasad Chodavarapu. SOA Security. Manning. 2008.

[3] NIST. Glossary of Key Information Security Terms. http://csrc.nist.gov/publications/nistir/NISTIR-

7298_Glossary_Key_Infor_Security_Terms.pdf [Accedida en Junio de 2010]

[4] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-Based Access Control Models," IEEE COMPUTER, vol. 29, 1996, pp. 38--47.

[5] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST Model for Role-Based Access Control: Towards A Unified Standard," IN PROCEEDINGS OF THE FIFTH ACM WORKSHOP ON ROLE-BASED ACCESS CONTROL, 2000, pp. 47--63.

[6] W. Fumy and J. Sauerbrey, Enterprise security: IT security solutions : concepts, practical experiences, technologies, Wiley-VCH, 2006.

[7] M. Papazoglou, Web Services: Principles and Technology, Prentice Hall, 2007.

[8] SAML Specifications. http://saml.xml.org/saml-specifications [Accedida en Junio de 2010]

[9] I. Tulloch, Microsoft Encyclopedia of Networking, Microsoft Press, 2002.

[10] A. Belapurkar, A. Chakrabarti, H. Ponnapalli, N. Varadarajan, and S. Padmanabhuni, Distributed systems security: issues, processes, and solutions, John Wiley and Sons, 2009.

[11] Dimitrios Georgakopoulos, M. P. Papazoglou. Service-Oriented Computing. The MIT Press. 2009

[12] M. P. Papazoglou, P. Traverso, S. Dustdar, F. Leymann. "Service-oriented computing: State of the art and research challenges". 2007.

[13] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pal Krogdahl, Min Luo, Tony Newling. Patterns: Service-Oriented Architecture and Web Services. IBM Redbooks. 2004.

[14] W3C. Web Services Description Requirements. http://www.w3.org/TR/ws-desc-reqs/#definitions [Accedida en Junio de 2010]

[15] Simple Object Access Protocol. http://www.w3.org/TR/soap/ [Accedida en Junio de 2010]

[16] Web Services Description Language. http://www.w3.org/TR/wsdl [Accedida en Junio de 2010]

[17] Universal Description Discovery and Integration. http://www.oasis-open.org/committees/wsrp/specifications/version1/wsrp-pfb-uddi-tn-1.0.pdf [Accedida en Junio de 2010]

[18] Web Services Addressing Working Group. http://www.w3.org/2002/ws/addr/ [Accedida en Junio de 2010]

[19] T. Rademakers and J. Dirksen, Open-Source ESBs in Action. Manning Publications, October 2008.

[20] Colombe Hérault, Gael Thomas, and Philippe Lalanda. Mediation and Enterprise Service Bus: A position paper. 2005.

[21] M. T. Schmidt, B. Hutchison, P. Lambros, R. Phippen. "The enterprise service bus: making service-oriented architecture real" IBM Syst. J., vol. 44, no. 4, pp. 781-797, 2005.

[22] W. Roshen, SOA-Based Enterprise Integration: A Step-by-Step Guide to Services-based Application, 1st ed. McGraw-Hill Osborne Media. 2009.

[23] WS-Security 1.1. http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf [Accedida en Mayo de 2010]

[24] WS-Trust 1.3. http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html [Accedida en Junio de 2010]

[25] Dave Chappell. Enterprise Service Bus. O'Reilly. 2004.