“Seguridad Informática en Web Services”

Farías Terrens, María Gabriela. Infante Salgado, Andrea
Director: Ing. Juan Pablo Garzón

RESUMEN:

Los Web services marcan un nuevo norte sobre el desarrollo de aplicaciones Web, son una nueva tecnología que todavía no ha conocido todo su potencial. Los desarrolladores sin embargo reconocen que las empresas se atemorizan al implementarlos porque pueden representar un gran riesgo de seguridad; es así, que empresas líderes como Microsoft e IBM proponen un framework llamado WSE (Web Services Enhancements) que garantiza cierto nivel de seguridad para aplicaciones de Web services. El trabajo de investigación realizado, propone y comprueba un Modelo de Seguridad orientado a Web services que busca complementar el trabajo ya existente.

ABSTRACT:

A web service represents a new trace for the roadmap of Web applications; they are a new technology that has not reach its full potential yet. Instead, developers recognize that organizations are afraid to adopt this new technology because it may represents a wide security risk, for this reason organizations like Microsoft and IBM proposed a framework (WSE – Web Servicces Enhancements) which guarantee some level of security in Web services applications. These research project propose and prove a security model for Web applications oriented to Web services which complements the already Microsoft and IBM proposal.

Introducción:

Diferentes razones o necesidades pueden hacer de un Web service la mejor solución informática, pero todas las soluciones pueden llegar a ser tan complejas, robustas y seguras como la profesionalidad e imaginación de los responsables pueda potencialmente ser.

A continuación se describen las dos aplicaciones más reconocidas. En el primero, se crea una capa de Web Services sobre una aplicación corporativa existente con el fin de permitir que sistemas externos puedan invocar las funciones de la aplicación a través de la Internet o Intranet corporativa sin tener que modificar la aplicación misma. El proveedor actúa como front end de aplicaciones que residen internamente en el mainframe, mientras que el cliente debe invocar y procesar la información que el proveedor responde. Mientras que en el segundo, una red de Web Services puede ser tan compleja al interior como se requiera; un Web Service que es contactado por una aplicación externa a la organización que lo publica, puede comunicarse internamente con uno o varios homólogos para lograr la funcionalidad; aproximándose a principios y patrones de software.

Ahora bien, estas aplicaciones por su naturaleza no incorporan un aspecto, la seguridad informática vista desde los siete principios de seguridad [1]: Integridad, Confidencialidad, No Repudio, Auditoría, Autenticación,

Autorización y Disponibilidad. De manera justificada, por el proyecto de investigación realizado, estas aplicaciones tienen un nivel alto de vulnerabilidad y que por lo tanto representan un riesgo para las organizaciones que los publican y los usuarios que representan, utilizan y confían en los servicios prestados por las mismas.

1. Investigación:

El proyecto de investigación propone un Modelo de Seguridad para aplicaciones en canales públicos orientados a Web services.

La investigación sustenta todo trabajo elaborado y garantiza la calidad del mismo. Por esto que el primer escalón que subimos en busca de alcanzar el objetivo principal es la investigación.
El primer tema estudiado fueron los Web services; ¿cuáles son los componentes?, ¿cómo se describen, publican, localizan y prestan sus servicios?, ¿qué riesgos de seguridad presentan?, entre otras.
A continuación abordamos un tema relacionada con la arquitectura de éstos, llamada Arquitectura Orientada a Servicios. Lo que profundizó, contextualizó y coordinó las partes que componen un sistema Web service simple.

En la medida que aprendíamos, empezamos a descubrir una a una las vulnerabilidades de esta tecnología. Un Web Service no es una aplicación segura porque no tiene un mecanismo de autenticación para usuarios y proveedores (Autenticación), además y teniendo en cuenta el medio de publicación, la Internet, cualquier persona con acceso a este medio puede acceder al servicio (Control de acceso). Sumado a lo anterior, el no seguimiento de la transacción permite el repudio de las acciones allí realizadas y su no registro (No repudio, Auditabilidad). Por último la información intercambiada entre proveedores y usuarios se encuentra en lenguaje XML no encriptado lo que permite que cualquier persona pueda interceptar, interpretar y modificar el mensaje (Integridad, Confidencialidad).

Es así que un grupo de organizaciones, compuesto por las más prestigiosas como Microsoft e IBM, creó Web Services Enhancements (WSE) [2][3][4][5][6][7]. Este framework de trabajo propone mediante elementos XML cómo dotar de seguridad a un sistema de información de este tipo. WSE hasta el día de hoy, no está terminado; para poder concluir que la mayoría de los siete componentes, todavía se encuentran en estado teórico, a excepción de WS-Security, WS- Policy y WS-SecureConversation.

La seguridad informática fue el siguiente tema de investigación; recurrimos a estudiar y comprender acerca de criptografia, entidades certificadoras, PKI y Kerberos, entre otros tópicos[8][9][10].

Todos los temas resultan ser importantes y complementarios para proponer el Modelo y aplicarlo en el Sistema de Administración de Derechos Electrónicos para el Depósito Central de Valores del Banco de la República.

2. Planteamiento del Modelo de Seguridad:

El siguiente escalón a subir nos dejó con el Modelo de Seguridad planteado después de realizar las labores de resumen, análisis, identificación y descripción. Es así, que se elaboraron tres modelos diferentes pero complementarios; ya que cada uno

ataca diferentes principios de seguridad, estos son: el modelo de interacción, el modelo de tolerancia a fallos y el modelo de seguridad.

a. Modelo de Interacción:

El ***modelo de interacción*** busca suplir de alguna manera la necesidad de ubicar dentro de la arquitectura una entidad de seguridad entre la entidad proveedora del servicio y la entidad consumidora.

Estas son las responsabilidades de la entidad de seguridad:

- **Recibe certificados digitales y/o llaves de la entidad consumidora y se encarga de establecer una conexión con la Autoridad Certificadora correspondiente con el fin de solicitar su validación.**
- **Luego de ser validada la identidad, genera un token de seguridad que asegura la identidad y los requisitos de consumo establecidos por la entidad proveedora.**
- **Valida la emisión de un token de seguridad previo.**
- **Renueva un token de seguridad emitido por ella con anterioridad por solicitud expresa de la entidad consumidora.**

b. Modelo de Tolerancia a Fallos:

El ***modelo de tolerancia a fallos*** busca definir las situaciones en las que pueden ocurrir fallos en el sistema y los efectos que producen en el mismo. Esto con el fin de comprender el impacto de los fallos y especificar mecanismos de control que mitiguen su aparición.

Ver ***Tabla N°1*** *Identificación de Fallos* (siguiente columna).

Así mismo se identificó y describió el impacto que cada uno de estos tendría sobre el Sistema en caso de suceder.. Así que decidimos evaluar cada uno y clasificarlos, comparando uno con otro.

Estos fueron los niveles establecidos de clasificación:

- ***Nivel de impacto Alto*****:** Cuando ocurre el fallo, se disminuye la operación del sistema de un 70% a un 100%
- ***Nivel de impacto Medio:*** Cuando ocurre el fallo, se disminuye la operación del sistema de un 40% a un 70%.
- ***Nivel de impacto Bajo:*** Cuando ocurre el fallo, se disminuye la operación del sistema en menos de un 40%.

Después de la evaluación se clasificaron.

Ver ***Tabla N°2*** *Clasificación de Fallos* (siguiente página).

Id.	Fallo
Fallos en las entidades del sistema de Web services	
f1	Caída del Servicio de registro
f2	Caída del Cliente
f3	Caída del servicio de seguridad
f4	Caída del Web service
Fallos en el canal de comunicación del sistema de Web services	
f5	Falla en el canal Cliente – Servicio de registro
f6	Falla en el canal Web service - Servicio de registro
f7	Falla en el canal Cliente – Servicio de Seguridad
f8	Falla en el canal Web service – Servicio de Seguridad
f9	Falla en el canal Cliente – Web service
f10	Corrupción de los mensajes enviados entre las entidades

Tabla N°1 *Identificación de Fallos*

Id.	Fallo	Nivel de impacto
f2	Caída del Cliente	Bajo
f6	Falla en el canal Web service - Servicio de registro	Bajo
f5	Falla en el canal Cliente – Servicio de registro	Medio
f7	Falla en el canal Cliente – Servicio de Seguridad	Medio
f1	**Caída del Servicio de registro**	**Alto**
f3	**Caída del servicio de seguridad**	**Alto**
f4	**Caída del Web service**	**Alto**
f8	**Falla en el canal Web service – Servicio de Seguridad**	**Alto**
f9	**Falla en el canal Cliente – Web service**	**Alto**
f10	**Corrupción de los mensajes enviados entre las entidades**	**Alto**

Tabla N°2 *Clasificación de Fallos*

Para cada uno de los fallos se describe una posible solución que contrarresta el efecto según el nivel de impacto.

c. Modelo de Seguridad:

El **modelo de seguridad** identifica y evalúa las vulnerabilidades de un Sistema y los riesgos a los que se expone, con el fin de establecer políticas y mecanismos de carácter preventivo, detectivo y correctivo que garanticen la seguridad del mismo.
El trabajo elaborado para la definición de este modelo se divide en tres partes que son:

La primera parte analiza e identifica los riesgos de seguridad a los que se expone un web Service cuando es publicado. En la segunda parte se aplica un anillo de seguridad que consiste en definir para cada riesgo (primera parte) un control preventivo, uno detectivo y otro correctivo. Finalmente, se establecen niveles de seguridad y se analiza la relación costo beneficio de su implementación.

Algunos de los riesgos a los que enfrenta un Web service a causa de sus vulnerabilidades se identifican a continuación:

- ✓ Acceso no autorizado a la información, recursos y servicios.
- ✓ Falta de mecanismos de validación de la identidad de las aplicaciones que acceden al servicio en representación de un cliente gestor.
- ✓ Interferencia en el canal de comunicación establecido entre las entidades participantes de forma activa (alterar el contenido del mensaje) o pasiva (conocer el contenido del mensaje).
- ✓ Suplantación de la identidad de una o varias entidades.
- ✓ Repudiación por parte una entidad de las acciones realizadas.
- ✓ Desconocimiento o no uso de los registros de acciones del Sistema (logs).
- ✓ Infección por virus en cualquiera de las entidades o de máquinas contenedoras.
- ✓ Ausencia de definición de requerimientos, estándares, controles y/o políticas de seguridad del Sistema.
- ✓ Ausencia de controles en la definición de documentos de descripción de servicio por parte de la entidad proveedora.
- ✓ Validación incorrecta de los tokens de seguridad por parte de la entidad de seguridad y proveedora.
- ✓ Incorrecta definición, implementación, configuración y control de la entidad de seguridad.
- ✓ Emisión de tokens de seguridad sin tiempo límite de vigencia.
- ✓ No ejecución de un plan riguroso de pruebas sobre el Sistema de Web service.
- ✓ Interrupción de uno o varios servicios prestados a causa de falta o fallas del fluido eléctrico.
- ✓ Funcionamiento incorrecto de servidores (contenedor del Web service) y dispositivos de redes.
- ✓ Rutinas de código malignas existentes en dispositivos de redes como servidores o equipos que afectan directamente al Sistema de Web service.
- ✓ Ausencia de planes de contingencia.

Una vez identificados los riesgos es necesario establecer un anillo de seguridad compuesto por tres tipos de controles: ***preventivos***, ***detectivos*** y ***correctivos***[11].

También relacionamos cada componente de la familia WS-* a uno o más principios de seguridad; con el fin de establecer qué componente y cómo el modelo daría soporte a cada uno de los siete principios con los avances realizados hasta el momento por otras organizaciones.

El modelo de seguridad concluye identificando tres niveles de seguridad, que son:

- **Nivel de Seguridad Bajo:** Cuando el Modelo da soporte al principio de seguridad de disponibilidad. El Web service con nivel de seguridad bajo se encuentra en un estado desprotegido y el índice de ataques el más alto.

- **Nivel de Seguridad Medio:** Cuando el Modelo da soporte a los principios de seguridad de

disponibilidad, autorización, autenticación, integridad y confidencialidad. El Web service con nivel de seguridad medio se encuentra en un estado medio de protección y el índice de ataques es de nivel medio.

- **Nivel de Seguridad Alto:** Cuando el Modelo da soporte a los principios de seguridad de disponibilidad, autorización, autenticación, integridad, no repudio, auditoría y confidencialidad. El Web service con nivel de seguridad alto se encuentra en un estado alto de protección y el índice de ataques es el más bajo, mas no es nulo.

Estos niveles de seguridad dan mayor soporte a los principios de seguridad según el nivel aumente, teniendo una contraparte, es decir, el costo de proveer seguridad: a mayor beneficio se ofrece o recibe, mayor el costo de su implementación y menor el número de organizaciones que están dispuestas a pagar grandes cantidades de dinero por herramientas tecnológicas de seguridad que soporten cada uno de los principios.

Principio de Seguridad	Nivel Básico	Nivel Medio	Nivel Alto
Disponibilidad	X	X	X
Autenticación		X	X
Autorización		X	X
Integridad		X	X
Confidencialidad		X	X
No Repudio			X
Auditabilidad			X

Tabla N°3 *Soporte a los principios de Seguridad por Nivel de Seguridad*

d. Integración de Modelos:

Siguiendo la técnica de diseño “Divide y vencerás” cada uno de los modelos planteados tiene una tarea específica para soportar un principio particular, que se integran en uno para formar un Modelo de Seguridad robusto que soporta los siete principios de seguridad.

El primer modelo sustenta la necesidad de agregar un componente de Seguridad entre el consumidor y el proveedor. El segundo modelo garantiza disponibilidad de la aplicación y el último define tres niveles de seguridad diferentes que soportan los seis principios faltantes.

3. Caso de estudio: Sistema para la administración de derechos electrónicos, Depósito Central de Valores - Banco de la República.

En el tercer escalón se logra desarrollando un caso de estudio. En su orden las actividades fueron: identificar la necesidad del Depósito Central de Valores (entidad administrada por el Banco de La República), definición de requerimientos funcionales y no funcionales, definición y descripción de los casos de uso utilizando UML (Unified Modeling Language).

a. Etapa de Análisis

De manera muy breve los depósitos centralizados de valores son entidades especializadas que reciben títulos valores para administrarlos mediante un sistema computarizado. Su objetivo es eliminar el riesgo que para los tenedores representa el manejo de títulos físicos, agilizando las transacciones en el mercado secundario y facilitando el cobro de rendimientos de capital e intereses. Y cumplen con las siguientes tareas específicas:

a) Eliminar el riesgo que para los tenedores representa el manejo de títulos físicos.
b) Agilizar las transacciones en el mercado secundario.

c) Facilitar el cobro de rendimientos de capital e intereses.

Los usuarios tienen la opción de tomar uno o varios de los siguientes servicios prestados por el Depósito Central de Valores:

- Inversión primaria en títulos inmateriales, que expida o administre el Banco de la República.
- Depósito de títulos físicos. (desmaterialización).
- Transferencia de títulos desmaterializados, por operaciones definitivas de compra y venta, con y sin compensación del valor de la transacción.
- Constitución y liberación de garantías entre depositantes titulares de la cuenta.
- Constitución y liberación de garantías entre depositantes titulares de cuenta y organismos de control y vigilancia.
- Retiro de títulos depositados en DCV (materialización).
- Pago automático de rendimientos y capital sobre títulos depositados en DCV.
- Información permanente del estado de depósito de los títulos desmaterializados.

A lo que se llegó al siguiente listado de requerimientos, cumpliendo así con la *etapa de análisis*:

<u>Requerimientos Funcionales:</u>

- ✓ El sistema debe permitir a los usuarios la consulta de la información relacionada con los derechos de un usuario dado su número de identificación en el sistema; los datos asociados a un derecho son: Número de expedición, descripción, valor, fecha expedición y fecha de vencimiento.

- ✓ El sistema debe permitir a un usuario liquidar uno de sus derechos dado el número de expedición del derecho, el número de identificación del usuario y el número de cuenta donde se desea consignar el valor del derecho una vez ha sido liquidado.

- ✓ El sistema despliega una lista de los movimientos realizados sobre los derechos de un usuario entre una fecha inicial y una fecha final suministradas por el mismo; la tabla de movimientos tiene la siguiente información: identificación del movimiento, fecha en la que fue realizado, desripcion del derecho implicado, identificador y descripción de la operación realizada y valor de la operación.

- ✓ El sistema permite al administrador consultar todos los derechos vigentes en el sistema, con la siguiente información por cada derecho: número de expedición, número de identificación del dueño del derecho, descripción, valor, fecha de expedición y fecha de vencimiento.

- ✓ El sistema permite al administrador transferir un derecho de un usuario a otro, dados el número de expedición del derecho y el número de identificación del usuario al cual será transferido el derecho.

- ✓ El sistema permite el ingreso o inicio de sesión de los usuarios registrados en el mismo mediante la solicitud al usuario de su username y password.

Requerimientos no funcionales:

- ✓ La consulta del portafolio del DCV deberá ser un servicio que esté disponible 7x24 pues los intermediarios deben tener esta información en el momento que la soliciten.
- ✓ El tiempo de respuesta de la consulta no deberá ser mayor a 5 segundos para períodos de un mes de consulta.
- ✓ Este debe ser un servicio accedido a través de una página Web y/o con la posibilidad de ser invocado desde sistemas de información de los intermediarios financieros.

b. Etapa de Diseño

En la *etapa de diseño* se crearon diagramas de clases (uno diferente para cada Nivel de Seguridad Bajo, Medio y Alto), diagramas Modelo Entidad Relación de las Base de datos y diagrama de componentes que definen el sistema.

Antes de terminar esta etapa se evaluaron diferentes herramientas de desarrollo como lo son Visual Studio.Net 2003 y J2EE. Debido a que Visual Studio.Net presenta un entorno de desarrollo rápido en Web service y es compatible con el framework Web Services Enhancements que garantiza algunas consideraciones de seguridad en Web services, la implementación de este caso de aplicación se realizará haciendo uso de.NET.

c. Etapa de Implementación

La *etapa de implementación*, comenzó haciendo un diagrama explicativo de las entidades que conformarían la solución, estas son:

Web browser aplicación cliente: Es la interfaz gráfica que permite la interacción del usuario con el sistema, en la cual el usuario puede seleccionar el nivel de seguridad con el que desea trabajar y realizar las acciones de administración de los derechos electrónicos tal y como se describe en los casos de uso que se encuentran en la sección 6.1.4. (ver anexos, manual de usuario y de configuración). Esta aplicación cliente consta de páginas aspx en lenguaje Visual Basic y páginas html que conforman la interfaz gráfica del cliente.

Web Service Fachada del sistema: Es el punto de entrada a la lógica del sistema, que administra la comunicación entre el cliente y los diferentes Web services. Cuando así se requiere solicita los tokens de seguridad al Web service de seguridad (para realizar autenticación y federación), y administra las llamadas al Web service del DCV y el bancario según la funcionalidad solicitada por el cliente.

Web Service de Seguridad: Es el Web Service que se encarga de la administración y emisión de tokens de seguridad con el fin de garantizar la autenticación de usuarios en el sistema y crear un contexto federado que establece un conjunto de participantes con relaciones de confianza implícitas entre ellos.

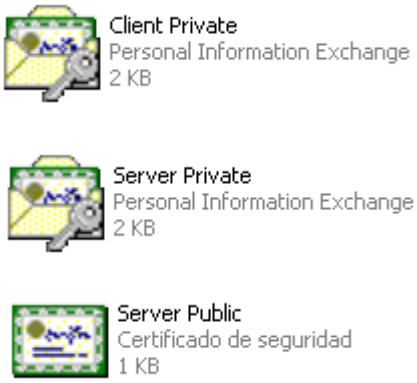
Web Service Depósito Central de Valores (DCV): Es el responsable de administrar los derechos electrónicos que se encuentran almacenados en la base de datos del Depósito Central de Valores. Se encarga de llevar a cabo las operaciones de consulta, liquidación y transferencia de derechos electrónicos

Web Service Banca Central: Este Web Service representa una aplicación específica de la Banca central que administra todas las transferencias monetarias entre diferentes cuentas, registradas en cualquier entidad bancaria, de los usuarios que liquidan y transfieren derechos electrónicos.

Las bases de datos se implementaron usando SQL Server 2000.

Base de Datos DCV: Repositorio que contiene toda la información relacionada con los derechos electrónicos del Depósito Central de Valores.
Base de Datos Banca Central: Repositorio que contiene la información relacionada con las cuentas de los usuarios del sistema.

Luego, se utilizó certificados X509 para la implementación y soporte a los principios de seguridad. Al descargar e instalar Web Services Enhancements trae consigo los siguientes certificados X509 de prueba:



que son utilizados para firmar mensajes SOAP (la llave privada del cliente se usa para ello) y para encriptar y desencriptar mensajes SOAP (se usa la llave pública y privada del servidor).

En la implementación del **Nivel de Seguridad Bajo** solo se soporta un principio de seguridad, ***disponibilidad***. Para garantizar el paso de control entre las máquinas que conforman el sistema replicado es necesario un balanceador de carga, el cual se encarga de direccionar las peticiones a la máquina que se encuentre más disponible y en capacidad de atenderlas, estos balanceadores de carga se pueden conseguir en software, en hardware o pueden ser implementados en una combinación de software y hardware.

En cuanto a los dispositivos Hardware, actualmente se encuentran en los mercados balanceadores de carga de marcas Foundry Networks, Nortel y Cisco entre otros, los cuales permiten distribuir la carga de trabajo en caso de falla de una máquina del sistema de forma automática, sin embargo también se pueden agregar estas funciones de balanceo de carga a firewalls, switches y routers de la red.

Luego de una evaluación definiendo un conjunto de criterios, obtiene mayor puntaje optar por una solución de software; para el caso de esta aplicación, siendo un sistema de prueba y académico, el precio y los requerimientos de hardware son puntos importantes a considerar, por lo tanto se prefiere un balanceo de carga de tipo software, preferiblemente de libre distribución y que opere sobre sistema operativo Windows.

Debido a la ausencia de la infraestructura necesaria para implementar un esquema de disponibilidad, el cual debe contar con una máquina servidor en la que reside el balanceador de carga y diversos nodos con las aplicaciones replicadas, en esta aplicación no se logró comprobar este principio; sin embargo

a partir de la investigación realizada se establecieron pautas y criterios importantes que garantizan la disponibilidad de aplicaciones Web.

Teniendo en cuenta los criterios de evaluación presentados se recomienda que:

- Si la infraestructura de la organización es pequeña, el presupuesto corto y la aplicación no requiere un alto porcentaje de disponibilidad, se puede implementar un balanceador de carga de software (ver Trabajo de Investigación para ampliar la información).

- Si la infraestructura de la organización es considerable, el presupuesto amplio y la aplicación requiere un alto porcentaje de disponibilidad, es aconsejable utilizar un balanceador de carga de hardware.

Para el **Nivel de Seguridad Medio** en el cual se garantizan los principios de seguridad de ***disponibilidad, autenticación, autorización e integridad*** se especificará a continuación cómo deben implementarse. Para el contexto de esta aplicación, teniendo en cuenta que tan solo se tienen dos roles de usuario con características bien definidas, la implementación del principio de autorización se realizó a nivel de presentación (interfaz gráfica).

Cuando el usuario selecciona la opción “Iniciar Sesión” y digita su username y su password, el sistema valida los datos ingresados por el usuario y si están bien, retorna información del usuario entre la que se encuentra el rol que este tiene en el sistema, el cual puede ser de dos tipos, Administrador o Usuario; de esta forma, dependiendo del rol que tenga, se prepara el portafolio de usuario o administrador con sus respectivas opciones. El rol de la persona que ingresa al sistema se guarda como una variable de sesión para manejar la interfaz gráfica que debe mostrarse al usuario.
Cabe aclarar que no se puede tener acceso directo a las páginas de los portafolios tanto de USUARIO como ADMINISTRADOR sin haber iniciado sesión previamente y que sólo desde estas páginas se puede tener aceso a las opciones de consulta de movimientos, liquidación y transferencia de derechos.

En este proyecto de investigación se intentó implementar una federación, utilizando un token de seguridad emitido por la Entidad de Seguridad. La Entidad se encargaría de crear y emitir el mismo token de seguridad a todos aquellos participantes que se ella incluye en una lista de suscritos. Luego de varias pruebas, debido a que se cuenta con certificados X.509 no originales, el proceso de autenticación fallaba de manera constante y por ende, la implementación de Federación también. Por esto solo se planteó este soporte de manera teórica.

Este principio se puede comprobar implementando una congregación de participantes los cuales deben compartir un mismo token de seguridad. Una entidad reguladora de la seguridad administra la emisión, validación, renovación y cancelación de la suscripción de un participante a la Federación; la suscripción consta de un token de seguridad especial ya que identifica de manera única al conjunto y sólo lo conocen aquellos participantes suscritos.
Cada vez que un participante solicita por primera vez al Web service de Fachada el acceso a un servicio, la Fachada solicita al Web service de Seguridad una suscripción a lo que se responde con el token de federación. Las próximas solicitudes, el cliente no deberá autenticarse con una firma ya

que se entabla de manera implícita una relación de confianza.
Si el cliente solicita la renovación del token de seguridad de la federación, la Fachada emite un mensaje a la entidad responsable y esta se encargará de eliminar la suscripción y crear una nueva, a lo que responde con un token con similares características excepto por su vigencia.
Cuando el cliente desea retirarse de la Federación, la Fachada nuevamente procesa la solicitud y la reenvía a la entidad de seguridad.

La autenticación se logra utilizando el certificado que el mismo cliente tiene y con él se crea un token de seguridad para firmar el mensaje que emite.
Para comprobar el soporte de WSE a este principio y dar así soporte a la aplicación desarrollada, se intentó utilizar certificados X.509 de prueba que WSE trae consigo. Estos certificados de prueba cuentan con las mismas características que uno original, pero ninguna Autoridad Certificadora los valida y aprueba en Internet, lo que limitó la prueba del código fuente que se encargaba de cumplir con el soporte a este principio de seguridad para aplicaciones Web.

Para los casos, en que no se pudo comprobar el soporte a estos principios por cuestiones de certificados, se incluye el código fuente necesario para el cliente como el servidor, que se encarga de procesar el mensaje SOAP y validar el contenido respectivo para cada caso.
La confidencialidad, también conocida como privacidad, es el proceso de asegurarse que los datos sensibles o privilegiados permanezcan privados y confidenciales y por consiguiente no puedan ser entendidos por usuarios sin autorización o por personas que monitorean el flujo de tráfico a través de la red (uso de sniffers).

La encriptación es la técnica usada para garantizar la confidencialidad en esta aplicación.
El framework WSE permite a clientes y Web Services creados usando ASP.NET encriptar y desencriptar mensajes SOAP, los cuales por defecto viajan por la red en texto plano y por lo tanto pueden ser vistos y entendidos por cualquier persona, las técnicas que este framework ofrece para la encriptación son la simétrica y la asimétrica; en la encriptación asimétrica un cliente puede encriptar el mensaje utilizando la llave pública del certificado X509 del usuario final (receptor del mensaje) para garantiza que solo el dueño de la llave privada de dicho certificado X509 pueda desencriptar el mensaje, por otra parte, la encriptación simétrica requiere que tanto el emisor como el receptor del mensaje compartan una llave secreta para con ella encriptar y desencriptar el mensaje SOAP.
Debido a que la técnica actualmente más usada de encriptación es la asimétrica, en esta aplicación se utilizan certificados digitales X509 de prueba que trae consigo el framework WSE para encriptar y desencriptar mensajes entre el cliente y los diferentes Web Services del sistema, por otro lado, teniendo en cuenta que estos Web services pueden estar ubicados físicamente en lugares remotos pero trabajan de manera conjunta para obtener un resultado, el cliente Web no necesita conocer la llave pública de otro nodo final, sino solamente la llave pública del nodo intermedio que se encargará de desencriptar la información que recibe del cliente, procesarla y nuevamente encriptarla con la llave pública del próximo nodo. Y así sucesivamente, hasta que el nodo final reciba el mensaje encriptado con su llave pública y lo desencripte con su llave privada.

La integridad es el principio que garantiza que los datos estén protegidos contra modificaciones accidentales o deliberadas. La integridad, se garantiza mediante la implementación de firmas digitales con XML; Mediante el uso de XML-Signature y certificados X.509. Los certificados por ser de prueba limitaron las pruebas de este principio ya que ninguna autoridad certificadora se presta para validar y autenticar su legitimidad.

El último nivel, ***Nivel de Seguridad Alto***, se deben soportar los siete principios de seguridad, a continuación se especificará cómo deben implementarse los principios de no repudio, confidencialidad y auditoria.

Para implantar un sistema de control interno informático se agrupan cinco componentes que cumplen con una secuencia y son: dirección, exigencias internas y externas, políticas y directrices, normas y procedimientos, implantar procedimientos de control, y comprobación y seguimiento de controles.
El principio de No Repudio se debe implementar en un sistema de Web Services teniendo en cuenta las siguientes consideraciones:

- Se debe verificar la identidad del usuario a través de un certificado digital u otro medio que la garantice.
- Se deben registrar las operaciones críticas realizadas en el sistema de tal forma que reflejen información clave para el no repudo como identificación del usuario, fecha y hora de la operación, entre otros.

Por estas dos razones se hace uso de las firmas digitales como medio para garantizar la autenticiad del usuario y el registro del log de auditoría para establecer exactamente el ámbito en el que se ha realizado cierta operación.

4. Pruebas

El último paso que dio este Proyecto de Investigación fue realizar un conjunto de pruebas para dar por completados y satisfechos los objetivos planteados. Las pruebas debieron limitarse a la medición del rendimiento, confiabilidad y disponibilidad del sistema en términos de **tiempo de respuesta** a las solicitudes del usuario y por otra parte la evaluación de la **confidencialidad e integridad** de la información que viaja entre los Web services mediante el uso de un analizador de protocolos de red. También evaluamos el correcto funcionamiento del balanceador de carga.

Los resultados más importantes son: primero, si las validaciones de seguridad aumentan, el tiempo de respuesta se hace más grande. También usando un sniffer, un analizador de protocolos, logramos observar que la información contenida en un mensaje realmente viaja encriptada. El balanceador de carga tuvo una limitante en las pruebas, debido a que sólo contábamos con dos máquinas y ninguna de ellas era servidor.

5. Conclusiones

Es así como este trabajo de investigación aporta a la seguridad de Web services un Modelo Integrado compuesto de un Modelo de Interacción, un Modelo de Tolerancia a Fallos y un Modelo de Seguridad; siendo este último el que alcanza la mayor profundidad de estudio.

El Modelo de Seguridad propuesto analiza las vulnerabilidades de estas aplicaciones Web e identifica, establece, describe e implementa un anillo de seguridad de tipo preventivo, detectivo y correctivo para soportar los

siete principios de seguridad que son el ideal de la máxima seguridad.

El Modelo de Interacción revalúa los componentes ya existentes de la arquitectura orientada a servicios y justifica un nuevo componente de Seguridad que interactúa entre el cliente y el Web service consumidor.

El Modelo de Tolerancia a Fallos identifica los incidentes que pueden presentarse en el sistema y plantea tres niveles de impacto en el cual son ubicados dichos incidentes. Mediante este análisis se logra dimensionar la importancia de la definición de una estructura que garantice la disponibilidad y confiabilidad del sistema.

Incluimos un Manual de Configuración que responde a todas las inquietudes, que debieron ser contestadas con gran esfuerzo, y se documentan cada uno de los pasos obligatorios para alcanzar el funcionamiento de la aplicación.

Este Proyecto de investigación propone políticas, metodologías, procedimientos, análisis e implementa un sistema de control interno de sistemas de información ya que Web Services Enhancements no tiene un componente destinado a la auditoría; principio muy importante para también soportar el principio de no repudio.

La Seguridad Informática es supremamente extensa y profunda, y también poco aplicada alrededor del mundo, no sólo en nuestro país. Es por esto, que así como este proyecto de investigación culmina, quedan abiertas las puertas e invitamos a nuevos estudiantes a vincularse con un tema de fuerte potencial en el mercado global.

6. Referencias bibliográficas

[1] Microsoft Corporation. "Web Application Security fundamentals". http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secmod/html/secmod74.asp Enero de 2004.

[2] Microsoft Corp., IBM y otros. "Web Services Secure Conversation Language (WS-SecureConversation)". http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-secureconversation.asp . Versión 1.1, Mayo de 2004.

[3] Microsoft Corp., IBM y otros. "Web Services Federation Language (WS-Federation)". http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-federation.asp . Versión 1.0, Julio 8 de 2003.

[4] Microsoft Corp., IBM y otros. "Web Services Trust Language (WS-Trust)". http://www6.software.ibm.com/software/developer/library/ws-trust.pdf Febrero 2005

[5] Microsoft Corp., IBM y otros. "Web Services Security Language (WS-Security)". http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpcondiscoveringwebservices.asp

[6] Microsoft Corp., IBM y otros. "Web Services Policy Language (WS-Policy)". http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-policy.asp

[7] Microsoft Corp., IBM y otros. "Web Services Trust Language (WS-Trust)". http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-trust.asp

[8] VeriSign Inc. "Understanding PKI". http://verisign.netscape.com/security/pki/understanding.html . 2000.

[9] Shashi Kiran, Patricia Lareau y Steve Lloyd. "PKI Basis, A technical perspective". http://www.pkiforum.org/pdfs/PKI_Basics-A_technical_perspective.pdf . Noviembre de 2002.

[10] IBM, Murdoch Mactaggart. "Introduction to cryptography". Marzo de 2001 http://www-128.ibm.com/developerworks/security/library/s-crypt07.html

[11] PIATTINI, Mario G, Del PESSO, Emilio. Auditoría Informática. Alfaomega, México DF. 2001.