Fortalecimiento en la Seguridad de Web Services para Aplicaciones Críticas

Eduardo Casanovas, Fernando Boiero, Carlos Tapia

Instituto Universitario Aeronáutico, Facultad de Ingeniería, Av. Fuerza Aérea 6500, Córdoba, Provincia de Córdoba, Argentina
ecasanovas@iua.edu.ar, fboiero@gmail.com, carlosignaciotapia@gmail.com

Resumen. Reconociendo a las propiedades de la Seguridad de la Información (confidencialidad, integridad y disponibilidad) como elementos esenciales de cualquier transferencia de datos, haciendo foco en el conjunto de protocolos y estándares utilizados para intercambiar datos entre aplicaciones web, es decir, Web Services, para el presente trabajo definimos el objetivo general de asegurar dichas comunicaciones, teniendo en consideración que la información intercambiada es igual o más crítica y confidencial que la transferida entre sistemas informáticos con la interacción humana.
Vamos a definir a las Aplicaciones Críticas, aquellas que se encuentran corriendo dentro de las Infraestructuras Críticas, siendo éstas las definidas por la Oficina Nacional de Tecnología de Información (ONTI), dependiente de la Jefatura de Gabinete de Ministros. En este marco es que los aspectos de seguridad cobran una relevancia superior y por tanto los esfuerzos para segurizar los sistemas deben ser extremos. Bajo este objetivo general, se definieron lineamientos particulares para efectuar lo que denominamos "hardening de seguridad de Web Services", lo que implica la conformación de una guía mediante la cual se establecen medidas de seguridad que ayudan a disminuir los riesgos de violación de las propiedades de seguridad mencionadas.

Palabras clave. web service, Infraestructuras Críticas, Seguridad Informática, cifrado, firma digital, XML, ONTI, security hardening.

1 Introducción

La Seguridad Informática es prioritaria en cualquier ambiente de tecnología donde se procese información computarizada, más aún cuando nos referimos al procesamiento de datos dentro de las Infraestructuras Críticas [**01**] de un país. Por Infraestructuras Críticas se entiende aquellas instalaciones, redes, servicios y equipos físicos y de tecnología de la información cuya interrupción o destrucción pueden tener una repercusión importante en la salud, la seguridad o el bienestar

económico de los ciudadanos o en el eficaz funcionamiento de los gobiernos de los Estados a los que pertenecen.

El concepto primario, el aspecto clave desde el que hay que partir, no es tanto la Infraestructura en sí, sino la función que ésta desempeña o el servicio que presta. En este marco, surge también el concepto de Aplicación Crítica, es decir, aquellos sistemas vitales que funcionan dentro de una Infraestructura Crítica. Por ejemplo, dentro de un aeropuerto (que podría considerarse una Infraestructura Crítica civil), el sistema que permite monitorear por radar a las aeronaves es una Aplicación Crítica.
Continuando con nuestro ejemplo del aeropuerto, cuando la Aplicación Crítica "Sistema de Radar" detecta una potencial colisión o proximidad riesgosa entre 2 aeronaves, se debe comunicar con otro sistema que gatilla alertas tempranas e informa a los servicios de emergencia. La comunicación entre ambos sistemas se establece mediante Web Services (WS), y de allí que sea vital proteger la información transmitida, teniendo en cuenta que errores u omisiones, intencionales o accidentales, pueden derivar en una catástrofe.
Mediante este razonamiento es que se propone en el presente trabajo una suerte de guía para defender a los Web Services que utilicen las Aplicaciones Críticas de cualquier Infraestructura Crítica.

2 Desarrollo

Los Web Services son servicios de intercambio de datos entre aplicaciones que utilizan una serie de protocolos estándares controlados por las organizaciones W3C, OASIS o el WS-I, siendo estos protocolos, por ejemplo, el Simple Object Access Protocol (SOAP), el Web Services Definition Language (WSDL) y el Universal Description Discovery and Integration (UDDI), los cuales en esencia permiten las transferencias de datos independizándose de los lenguajes de programación en que fueron programadas las aplicaciones intervinientes.

En relación a los protocolos mencionados, el WSDL es el encargado de poner a disposición una API (Application Programming Interface) en un formato comprensible para cualquier sistema. Para ello, se vale del lenguaje XML y determina los requisitos funcionales necesarios para establecer la comunicación con el Web Service.

Luego, el protocolo SOAP define la manera en la que se efectuará el intercambio de información mediante XML. Los clientes de conexión utilizan SOAP para efectuar las llamadas a las funciones disponibles mediante la API WSDL.

Por su parte, UDDI se utiliza para publicar la información asociada al Web Service y permite comprobar su disponibilidad y estado.

En forma gráfica, el funcionamiento de un Web Service genérico puede esquematizarse de la siguiente manera:

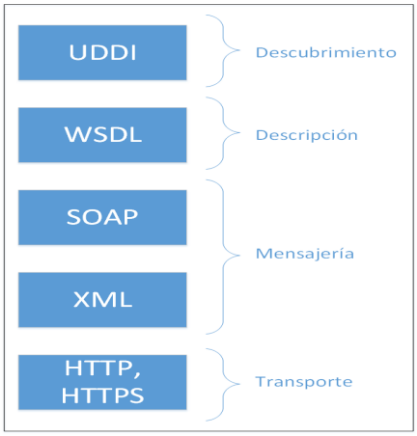

Imagen 1 - Esquema genérico de un Web Service

Los principales riesgos de seguridad de Web Services podrían resumirse en los siguientes ítems:

- No fueron gestados con conceptos de seguridad embebidos.
- Los datos transferidos son confidenciales y deben ser protegidos de igual modo que la información contenida en las bases de datos.
- Existe la posibilidad de captura de la información en texto plano en tránsito.
- Falta de mecanismos de validación de identidad o autenticación para consumir la información brindada por el Web Service.
- Interferencia en el canal de comunicación.
- Interrupción del servicio y falta de plan de contingencia.

Estos riesgos pueden materializarse en ataques como los siguientes:

- A nivel de aplicación:
  - Las aplicaciones transfieren datos relevantes de los procesos de negocio de la organización.
  - Los bugs de diseño, errores de reglas de negocio, se mantienen con o sin WS.
  - Cualquier ataque de la guía OWASP [**02**] se mantiene.
- A nivel de SOAP (Simple Object Access Protocol):
  - Interfaces escritas con WS Description Language.
  - En general no se implementan controles de acceso a WSDL.

- A nivel de XML:
  - Es un estándar para representación de sets de datos.

- Es posible efectuar ataques para que el parser falle.
- Sujeto a ataques de inyección de código.

Luego, a los efectos de tener un marco de referencia específico para comprender los ataques y sus posibles controles mitigantes, el proyecto se vale de la reconocida Guía de Seguridad OWASP para tomar herramientas y métodos, puntualmente tomando los temas tratados en la "Web Service Security Cheat Sheet" [**02**], cubriendo los siguientes aspectos:

1) Transport Confidentiality
2) Server Authentication
3) User Authentication
4) Transport Encoding
5) Message Integrity
6) Message Confidentiality
7) Authorization
8) Schema Validation
9) Content Validation
10) Output Encoding
11) Virus Protection
12) Message Size
13) Availability
14) Endpoint Security Profile

Asimismo, a modo de complemento, resultó interesante incluir las recomendaciones vertidas por el NIST (National Institute of Standards and Technology) de Estados Unidos, el cual publica el documento "Guide to Secure Web Services" (Special Publication 800-95 de Agosto del 2007 [**03**]), el cual contiene una extensa guía con muchos detalles sobre cada medida de control, brinda ejemplos de código fuente para materializar las medidas de seguridad y analiza distintas posibilidades de implementación.

Dimension	Requirement	Specifications
Messaging	Confidentiality and Integrity	WS-Security
		SSL/TLS
	Authentication	WS-Security Tokens
		SSL/TLS X.509 Certificates
Resource	Authorization	XACML
		XrML
		RBAC, ABAC
	Privacy	EPAL
		XACML
	Accountability	None
Negotiation	Registries	UDDI
		ebXML
	Semantic Discovery	SWSA
		OWL-S
	Business Contracts	ebXML
Trust	Establishment	WS-Trust
		XKMS
		X.509
	Trust Proxying	SAML
		WS-Trust
	Federation	WS-Federation
		Liberty IDFF
		Shibboleth
Security Properties	Policy	WS-Policy
	Security Policy	WS-SecurityPolicy
	Availability	WS-ReliableMessaging
		WS-Reliability

Imagen 2 - Medidas de seguridad de WS propuestas por NIST

Se describe a continuación las medidas de seguridad implicadas en cada dimensión de la Guía 800-95 de NIST:

- ***Dimensión "Messaging***": Dentro de esta dimensión se destacan aquellas tecnologías que fortalecen SOAP (que no fuera pensado inicialmente con conceptos de seguridad), como por ejemplo HTTPS, cifrado XML y la firma digital de XML, con los consiguientes beneficios a la hora de dificultar o impedir capturas o manipulación de mensajes en texto plano transmitidos por los WS. Esta dimensión también incluye WS-Security que define los mecanismos para utilizar tanto el cifrado como la firma digital de mensajes SOAP.

- ***Dimensión "Resource***": Implica la utilización de mecanismos de control de acceso para que los WS puedan ser consumidos sólo por aquellos sistemas autorizados. Un ejemplo de alternativa de autenticación para WS es XACML (eXtensible Access Control Markup Language [**04**]).

- ***Dimensión "Negotiation":*** Esta dimensión engloba a los mecanismos mediante los cuales se automatiza la negociación y el

descubrimiento de nuevas funcionalidades ofrecidas por los WS. Los subcomponentes incluidos en la dimensión "Negotiation" son:

- Protocolo de negociación.
- Servicio de negociación.
- Servicio de mediación.
- Servicio de auditoría.
- Servicio de informe de status de negociación.

Ejemplo de protocolo de negociación es ebXML (Electronic Business using eXtensible Markup Language [**05**]).

- ***Dimensión "Trust":*** Para cumplir con esta dimensión esencialmente se selecciona uno de los modelos de confianza más difundidos para WS:

  - Modelos "Pairwise": Es el modelo más simple, pero también el menos escalable.
  - Modelo "Broken trust": se utiliza una tercera parte independiente para determinar la relación de confianza.
  - Modelo "Federated trust": se utilizan mecanismos de federación de relaciones de confianza, combinando el modelo "Pairwise" y el "Broken trust".
  - Modelo "Perimeter defense strategy": se utilizan dispositivos llamados "XML gateways" que funcionan como proxy entre los proveedores y los solicitantes.

- ***Dimensión "Security Properties":*** Dentro de esta dimensión se proponen medidas de seguridad adicionales como lo son WS-Policy [**06**], WS-SecurityPolicy [**07**], WS-ReliableMessaging [**08**] y WS-Reliability [**09**].

El foco del presente trabajo de investigación es la confección de una guía de hardening de seguridad específicamente aplicable a Web Services, que, si bien es de aplicación universal para cualquier Web Service de cualquier organización, se pone especial énfasis en aquellos que intercambian datos dentro de las Infraestructuras Críticas, es decir, los Web Services de las Aplicaciones Críticas.

A los efectos del presente trabajo de investigación, se tomó como ejemplo de aplicación a un Web Service que transfiere información del Sistema de Gestión de AIT (Asociación de Investigaciones Tecnológicas) y el Sistema de Gestión del IUA (Instituto Universitario Aeronáutico).

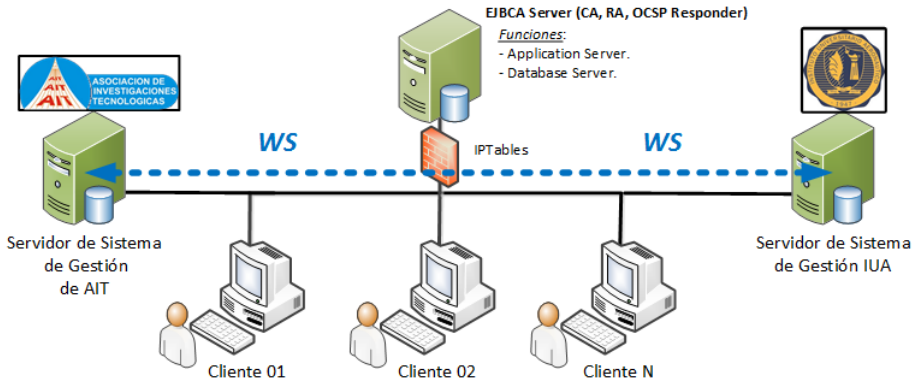

Imagen 3 - WS utilizado como ejemplo de aplicación

3 Hardening

Para acotar nuestro trabajo vamos a describir las actividades generales de un modelo de hardening sobre WS, pero haciendo foco sobre las tecnologías utilizadas en la aplicación del ejemplo. Haciendo una analogía sobre estas actividades se podría utilizar este modelo sobre otras tecnologías, siempre teniendo en cuenta las particularidades de cada una y los nuevos actores que podrían surgir que hacen mover el enfoque de la matriz de riesgos asociada.
Del relevamiento, los resultados obtenidos fueron los siguientes:

Servidores intervinientes:

- Servidor de Aplicaciones AIT
  - Sistema Operativo: GNU/Linux Centos 6.6 x64
  - Servidor de Aplicaciones: Apache Tomcat 7.0.47

- Servidor de Aplicaciones IUA
  - Sistema Operativo: GNU/Linux Debian 7 Whezzy x64
  - Servidor de Aplicaciones: Apache Tomcat 8.0.11

Los WS intervinientes son del tipo SOAP implementados en Java utilizando el API provisto por Apache CFX en su versión 3.0.5 para JAX-WS [**10**]. El Servidor del WS está desplegado el Servidor de Aplicaciones del AIT en el puerto 8080 y desde un cliente desplegado en el servidor de Aplicaciones de IUA se lo consume.

La autenticación entre el cliente y el servidor se hace por un mecanismos desarrollado por los programadores por derivaciones de funciones de hash aplicadas al usuario y contraseña pre-compartidos.

Después del análisis de los requerimientos tanto de la guía del OWASP ("Cheat Sheet de WS") como de la Guía del NIST (Special Publication 800-95), incorporamos a la infraestructura de seguridad de nuestros sistemas una arquitectura PKI compuesta por una Autoridad de Certificación que emitirá los certificados necesarios para los servidores y aplicaciones según los diferentes requerimientos. En la Autoridad Certificación será necesario crear los siguientes perfiles:

- Perfiles de certificados de Sitios Seguros utilizados por los Servidores Apache Tomcat para dar soporte SSL/TLS.
- Perfil de Certificado de Aplicaciones, los que son utilizados por los Web Services para la autenticación y cifrado de los XML.

El otro elemento incorporado fue un servicio de emisión de token de seguridad "SecurityTokenService" con el cual se validarán los accesos a los WS.
Además, será necesaria una reingeniería de las aplicaciones en su diseño de seguridad que se adecue a los siguientes estándares en cada una de las siguientes dimensiones:

- **Messaging**
  - Confidentiality and Integrity
    - Modificar la configuración de Apache CFX para que implemente los servicios de WS-Security.
    - Cambiar la Configuración de Apache Tomcat instalando los certificados emitidos SSL/TLS.
  - Authentication
    - WS-Security Tokens Se agregó en un servicio para generar estos accesos.
    - SSL/TLS X.509 Certificates. Se incorporaron al proceso de Autenticación.
- **Trust**
  - Establishment
    - WS Trust
    - X.509
- **Security Properties**
  - Policy
    - WS Policy

En resumen, se observa que con una simple reingeniería de la aplicación en los servicios de autenticación, reutilizando un componente preexistente (PKI) y agregando un nuevo servicio (SecurityTokenService) se mitigaron los posibles riesgos de la mayoría de los requerimientos de seguridad planteados por los estándares internacionales mencionados.

Como trabajo futuro se prevé implementar las restantes medidas de hardening para Web Services, planteando a partir de la implementación completa de la guía de fortalecimiento, una cierta periodicidad para evaluar los controles, garantizando razonablemente la protección de la información transferida vía Web Services.

Adicionalmente, se apuntará a terminar de relevar todos los WS del IUA y AIT para efectuarles el correspondiente hardening según la guía elaborada.

4 Conclusión

Con las medidas aplicadas demostramos que se incrementaron los niveles de seguridad de los datos transferidos vía WS entre los sistemas tomados como ejemplo.
Pudimos comenzar con la utilización de la PKI ya montada para cubrir parte del hardening de WS.
Sentamos las bases para auditar y efectuar hardening de cualquier WS de cualquier organización, incluidas las Infraestructuras Críticas.

5 Referencias bibliográficas:

[01] https://www.incibe.es/CERT/Infraestructuras_Criticas/

[02] https://www.owasp.org/index.php/Web_Service_Security_Cheat_Sheet

[03] http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf

[04] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[05] http://www.ebxml.org/geninfo.htm

[06] http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html

[07] http://docs.oasis-open.org/ws-rx/wsrm/200702

[08] http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf

[09] http://www.w3.org/TR/ws-policy/

[10] http://cxf.apache.org/docs/index.html