Aplicación Web de eBanking para realizar consultas, transferencias y pagos de los usuarios de una entidad financiera no bancaria cliente de la empresa SINC

LI. Nadia Guzmán Hinojosa
nguzmanh@gmail.com

M.C.E. Beatriz Alejandra Olivares Zepahua
bolivares@ito-depi.edu.mx

Tecnológico Nacional de México
Instituto Tecnológico de Orizaba

Maestría en Sistemas Computacionales

Objetivo

Desarrollar una aplicación Web de *e-Banking* para facilitar la gestión de sus pagos y manejos de cuenta y seguimiento de saldos de ahorro de los socios en la empresa, utilizando la seguridad de compartición de mensajes que implementa WSI (*Web Services Interoperability*, Interoperabilidad de los Servicios Web).

Introducción

Las entidades financieras no bancarias ofrecen fuentes alternativas de ahorro y crédito dentro del sector financiero de México. En este ámbito, el hecho de contar con instrumentos de pago seguros, al alcance de la población y fáciles de usar es indispensable. Este trabajo presenta el desarrollo de una aplicación Web de *e-Banking* cómo un módulo adicional a un sistema de escritorio ya existente, para efectuar transacciones de: transferencias, consultas y pagos en una entidad financiera no bancaria. Al añadir una solución basada en Web fue necesario garantizar la interoperabilidad entre las dos aplicaciones [1], los servicios Web basados en SOAP, son un mecanismo efectivo [2], así mismo una aplicación Web, por su naturaleza de alcance público, requiere mecanismos de seguridad que contrarresten diversas amenazas. El presente trabajo se remitió, entre otras, a la fuente de información que brinda OWASP (*Open Web Application Security Project*) [3]. que dentro de sus documentos más exitosos incluye OWASP Top 10, que contempla las diez vulnerabilidades más importantes en las aplicaciones Web, y bajo dicha información se implementaron los mecanismos que contrarresten dichas amenazas.

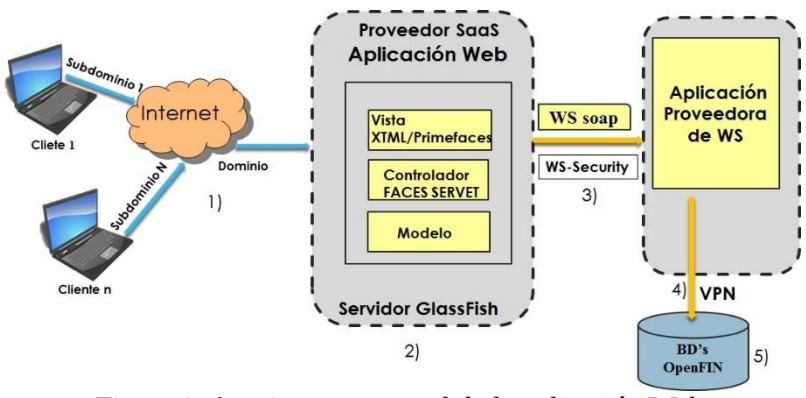

Figura 1. Arquitectura general de la aplicación Web

Mecanismos de seguridad

La amenaza más importante en el entorno de las aplicaciones Web presenta ataques como: "*Injection*" la cual considera la posibilidad de que cualquier persona envíe datos no confiables al sistema para ser ejecutados. La amenaza de "*Broken Authentication and session management*" se refiere a los atacantes externos y anónimos que intentan robar cuentas de otros o que tratan de usar funciones que no tienen permitidas. Por otro lado "*Sensitive Data Exposure"* es otra vulnerabilidad, que considera a quién puede acceder a datos y archivos confidenciales o a cualquier copia de esos datos para hacer mal uso de los mismos.

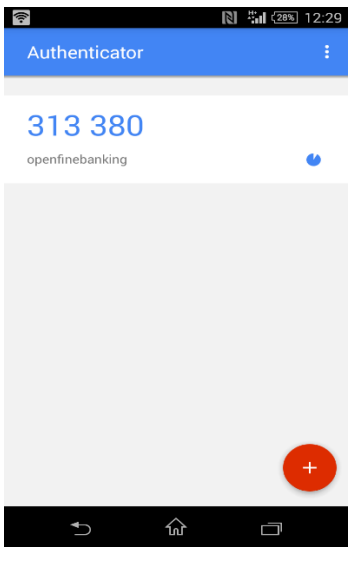

Figura 3. *Token* de autorización generado en el dispositivo móvil

Arquitectura

En la figura 1 se presenta la arquitectura general de la aplicación, así como las tecnologías necesarias que conforman la misma. La arquitectura de software bajo la cual se desarrolló la solución Web de *e-Banking* consiste en 2) la aplicación Web que ve el usuario de la entidad financiera y 4) la aplicación proveedora de WS que funciona como intermediario entre el repositorio de la aplicación de escritorio actual (OpenFIN) y la aplicación Web, ésta se conecta mediante funciones almacenadas a la base de datos del sistema principal.

```
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <S:Body>
      <ns2:wslogin xmlns:ns2="http://wsprovider/">
         <usr>nguzmanh@gmail.com</usr>
         <pwd>usuario1</pwd>
         <session>afs332ash</session>
         <firma>
         YmbXxkN3UIbanb7RYwHZopRA6xpTfz/GUBKlmlQqhXyo2P4tVAkvrB8ZSxXn636uI7/szmgM3BlKzK
         pdSovan0uKo4HhS5x7x6RfV/1/Ybzh9NT1yh/4S7YxAQoajJWE98cMK6lx4DVHbvyH4HKxJ56nrl+1X/b6Qxs
         qNJlmL99m7SXmYbzreQTmiSxvqwqKSZ+qdHMHc84370vC7/q3gAY9BYtQ0VzNMHjq5kbZHn7cpYNPcCVE73RA
         zx3aXbU5CIWtIRQQvQ55AHXXezJOER4R3ulwaUiD2rTS+KZt3QiQresPwrd2SM6lgpmM0z6doJA+YMDZpJA+p
         MgSv+Zm7w==
         </firma>
      </ns2:wslogin>
   </S:Body>
</S:Envelope>
```

Figura 2. Firma del mensaje

Resultados:

Se obtuvo una aplicación donde es posible realizar las operaciones comunes con las cuentas de los usuarios de la entidad financiera no bancaria. Además en la aplicación Web se aplicaron las capacidades de validación de JSF y PrimeFaces para verificar el formato de los datos de entrada para tratarlos como texto y evitar que se interpreten como comandos ejecutables. El control de autenticación mediante clave y contraseña cifrada obliga a que el usuario se autentifique y al mismo tiempo dichos elementos sean ilegibles ante los atacantes. El uso de *tokens* para autorizar transacciones obliga a que solo el propietario de la cuenta autorice una transacción. El filtro de Java Servlet, no permite visualizar páginas si no se ha iniciado sesión y evita acceder a funciones prohibidas. La firma digital en los WS no ejecuta una petición si el mensaje no mantiene su integridad. Por su parte el uso de un canal HTTPS asegura la información cifrada al transitar por Internet.

.

Conclusiones

En este trabajo se destaca la importancia que tiene cubrir las áreas más vulnerables de la seguridad de una aplicación Web de *e-Banking*, debido a que la información almacenada es muy delicada. Por lo tanto, la implementación de los mecanismos vistos permite el ofrecer la confianza a los usuarios finales de que sus datos serán gestionados y almacenados con el debido cuidado.

Referencias

[1] D. k. Barry, Web Services, Service-Oriented Architectures and Cloud Computing. Waltham Estados Unidos, 2013.
[2] "About OWASP." [Online]. Available: https://www.owasp.org/index.php/About_OWASP. [Accessed: 02-Feb-2017].
[3] "Service Oriented Architecture." [Online]. Available: http://www.omg.org/news/meetings/workshops/MDA-SOA-WS_Manual/00-T1_Newcomer/CH5-SOAandWS_V9-Standard.pdf