Desarrollo de una Web-App con componentes seguros para un inicio de sesión no vulnerabable.

Eduardo
Instituto Tecnológico y de Estudios Superiores de Monterrey
Ingeniería en Tecnologías Computacionales (ITC)
Naucalpan de Juárez, Estado de México, México
a01749375@tec.mx

Bruno Passarette Santos
Instituto Tecnológico y de Estudios Superiores de Monterrey
Ingeniería en Tecnologías Computacionales (ITC)
Miguel Hidalgo, Ciudad de México, México
a01658904@tec.mx

Resumen— El contenido de este artículo tiene como fin informar acerca del desarrollo de una aplicación Web, su diseño de funcionamiento con ayuda de diagramas y del código que fue utilizado en un inicio. La primera parte de este proyecto fue realizar una página sencilla que fuera fácil de atacar y no contara con muchas características que sumadas hacen a una página menos vulnerable. La página podría haberse visto como una aplicación Web insegura que además tenía una puerta trasera que podía ser vista como una vulnerabilidad que podía ser explotada por atacantes y robar información de los usuarios y tomar control de la página y utilizarla de otra manera. El día de hoy, la página web ya cuenta con ciertas medidas que verifican el paso de los usuarios, almacena las contraseñas en una base de datos y permite que el ingreso a la página se vea asegurado. Habiendo dicho eso, se realizará también una comparación directa con ciertos puntos para verificar la seguridad de las páginas Web y se tratará de exponer lo delicado y sencillo que es dejar una vulnerabilidad en una aplicación y que sea expuesta a distintos atacantes.

I. INTRODUCCIÓN

Hoy en día el uso de las aplicaciones móviles, web ha causado una revolución en la forma en la que visualizamos a nuestro entorno. Por esta misma razón, es también momento de preguntarse, ¿nuestro entorno conoce los peligros que corren al dar un simple click? Nuestro trabajo como programadores o especialistas en ciberseguridad nos dicta que seamos éticos y no facilitemos la creación de puertas traseras, así como también debemos de tener cuidado y vigilar que estas no existan en el producto de un cliente. Como analista o desarrollador es nuestra responsabilidad mantenernos éticos y sin sin ningún tipo de doble intención que pueda perjudicar al usuario o al equipo desarrollador.

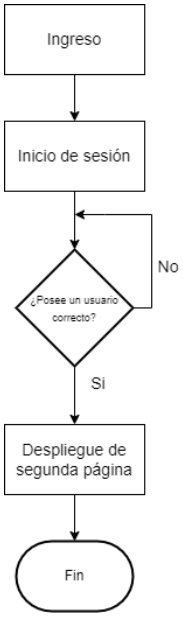
A través del artículo describiremos la primera fase del desarrollo, que fue la más sencilla de implementar, una aplicación insegura con una puerta trasera que pueda ser atacada. Posteriormente describiremos el desarrollo con la solución a una aplicación segura. Seguido, se realizará una comparación con puntos de OWASP y sus medidas de seguridad para aplicaciones.

II. DESARROLLO

El desarrollo de este proyecto fue dividido en dos partes para garantizar el entendimiento de la actividad y que así nosotros pudiéramos garantizar un acceso seguro en la vida real habiendo visto lo sencillo que es dejar por accidente o mero descuido una puerta trasera. Por esta misma razón, es importante mencionar que la aplicación que fue hacer un *login*, se vio modificada a simple vista, muy poco. Sin embargo, los cambios en el *back-end* fueron mejoras que significan un cambio gigantesco en la seguridad del usuario y sus credenciales de inicio. A continuación serán descritas ambos procesos de creación, desarrollo e implementación.

II.-A DESARROLLO APLICACIÓN INSEGURA

Como se mencionó, la realización de la aplicación insegura, fue algo sencillo sin muchas configuraciones que pudieran garantizar un acceso seguro. La finalidad de esta primera actividad era poder hacer algo “al aventón” y que fuera muy sencillo de vulnerar. Por esta misma razón, fue creada únicamente una sencilla entrada en la que un usuario se daba de alta desde el código con sus credenciales.

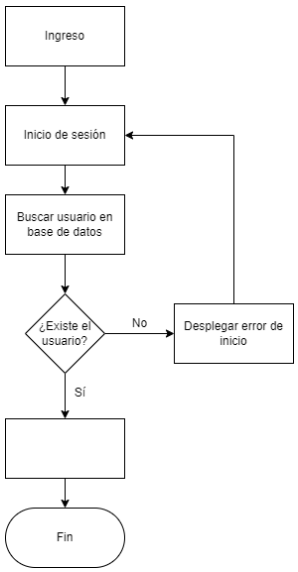


1.1 Diagrama de flujo de aplicación insegura

Una vez creada esta actividad, en el código, el usuario ingresaría, tendría acceso a una nueva página y podría salir una vez más a la página de sesión. Sin forma de verificar más profundo un usuario o con las contraseñas hasheadas, estas podían ser visualizadas en el código de la página y cualquier persona que inspeccionara la pagina podría haber visualizado ahí un usuario con su contraseña.

Por esta misma razón, la página podía verse atacada desde fuera de maneras muy simples, y necesitaba una mejora urgente para contar con métodos de seguridad aprobados por la OWASP y que contara con un ingreso confiable de sesión para el usuario.

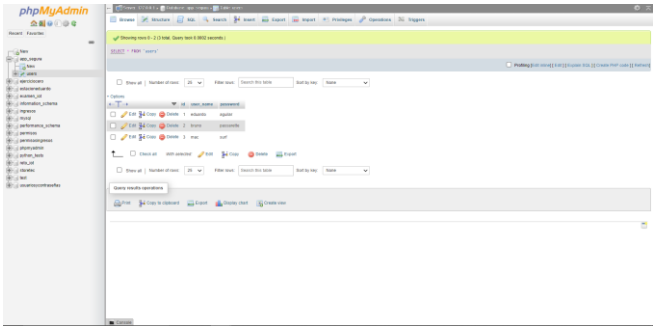
II.-B DESARROLLO APLICACIÓN SEGURA



2.1 Diagrama de flujo de aplicación segura

Las acciones a realizar una aplicación con un *login* seguro fueron implementadas después de una revisión con nuestra profesora, en la que se revisó el código y la funcionalidad de la aplicación, viendo siempre que no solamente fuera más difícil de esquivar la seguridad, sino que también las medidas correspondieran a unas medidas de seguridad para aplicaciones de OWASP. Nuestras acciones para mejorar una aplicación fueron las siguientes:

- Guardar las contraseñas dentro de una base de datos en *phpMyAdmin*.



2.2 Base de datos implementada por el quipo

Como podemos visualizar, en la base de datos fueron creados 3 usuarios, mac, eduardo y bruno y la página funciona correctamente con cualquiera de estos usuarios.

- No permitir que el usuario tuviera una contraseña menor a 8 caracteres ASCII.

```
<input type="text" name="uname" placeholder="Username"><br>
<label>Password</label>
<input type="password" name="password" minlength=8 maxlength="14" placeholder="Password"> <br>
```

2.3 Limitaciones en contraseña

En esta captura, podemos visualizar que en caso de tener una contraseña menor a 8 dígitos, pero superior 14 lanzará una alerta al usuario que desee hacer inicio de sesión y no podrá pasar. Además de que tapará los caracteres de la contraseña al ser ingresada al poseer un tipo de atributo *password*

- Hashear las contraseñas en la base de datos

```
echo password_hash("$password", PASSWORD_DEFAULT);
```

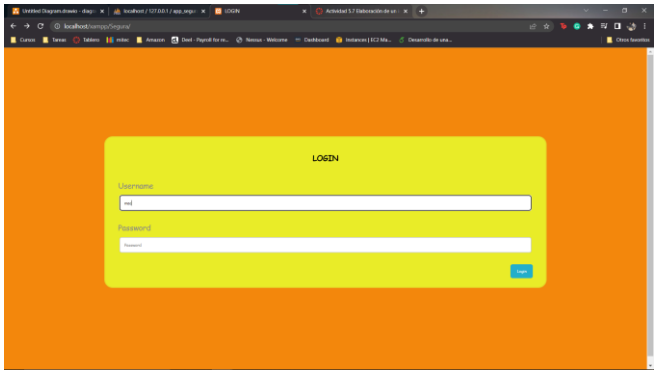
2.4 *Hasheo* a contraseñas

Una vez implementadas estas medidas nuestra aplicación se vería más segura y un ataque sería mitigado, nuestra puerta trasera sería mitigada y no sería más complicado de acceder a los usuarios y sus credenciales, así que el objetivo de esta parte habría sido alcanzado y la vulnerabilidad se habría visto mitigada.

Link de github: https://github.com/EduAguich/Aplicaciones

III. PRUEBAS

Para esta parte hemos adjuntado el GitHub de un integrante del equipo en la que se puede visualizar todos los códigos implementados para la creación de esta práctica.



3.1 Login en la página

Por supuesto, además se grabaron videos en los que se puede ver el funcionamiento de esta página y como es posible crear usuarios desde la base de datos. En el GitHub se podrán encontrar ambas carpetas, los videos y además el archivo en *pdf* una vez más cargado pero esta vez directamente desde el *Github*. Lamentablemente, no es posible cargar la base de datos, debido a que fue implementada siempre de manera local, sin embargo, en el código se puede visualizar la estructura de la misma además de la captura 2.1.
La página tiene un inicio, una página home en la que inicia sesión y despliega una bienvenida para el usuario en turno, y un *logout* que acaba la sesión.

IV. CONCLUSIONES

La creación de la página nos permitió visualizar lo sencillo que es dejar pequeños huecos en los que una vulnerabilidad puede verse incluida por descuidos, flojera o incluso negligencia. Es importante nosotros como programadores siempre estar atentos a las recomendaciones de agencias de seguridad y mantenernos siempre lo más éticos y parciales a la hora de desarrollar alguna aplicación en el mundo laboral.
Haciendo este informe, buscamos similitudes a los controles sugeridos por la OWASP para aplicaciones móviles, y estamos seguros que el control MSTG-STORAGE-7 lo cumplimos, debido a que no mostramos en la interfaz ningún tipo de dato sensible, es por eso que estamos seguros que cumplimos con este control.
También manejamos una política de contraseñas, tal como lo es nuestro mínimo y máximo, así que el control MSTG-AUTH-5 es llevado a cabo correctamente, de otra manera sale una alarma que inválida el ingreso. Además de que el usuario si comete el error de autenticación, no se le informa en dónde se ha equivocado si llegase a cometer el error de autenticación, no se le informa en dónde fue que existe el error.

Para concluir este trabajo, queremos decir que el aprendizaje que nos llevamos Eduardo y Bruno, es que las aplicaciones móviles deben de pasar por muchas pruebas y análisis antes de ser liberadas al público y esto no lo sabíamos. Por esto mismo estamos impresionados, sin embargo, también nos dimos cuenta de lo sensible que es cometer un diminuto error de tipografía. Un dedazo puede costar daños irreparables para los desarrolladores y su credibilidad y fama como programadores se ve dañada sin vuelta atrás. Por eso mismo, si decidiésemos desarrollar en un futuro, verificaremos los controles y recomendaciones por especialistas en ciberseguridad, para minimizar las amenazas. Además de someter la aplicación a infinidad de pruebas para realmente comprobar que no haya algún cabo suelto que pueda perjudicarnos.