Trabajo Vulnerabilidades Web

Sistemas de gestión de seguridad de sistemas de información

Jon Ander Fontán / Egoitz Puerta
5-2-2013

Índice

Introducción

El trabajo consiste en crear una aplicación web que contenga varios errores, citados en el informe de la OWASP sobre las 10 vulnerabilidades web más importantes, y luego subsanar esos errores.

A continuación se citan los cinco errores que nuestra aplicación contiene, donde están localizados y como los hemos subsanado.

Errores

A1 – Inyección

¿Qué es?

> *Las fallas de inyección, tales como SQL, OS, y LDAP, ocurren cuando datos no confiables son enviados a un intérprete como parte de un comando o consulta. Los datos hostiles del atacante pueden engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados.*

¿Dónde está el error?

El error de inyección SQL se puede cometer al no tratar lo introducido en los campos de texto por usuarios, como en el formulario de identificación inicial. Esto es, que el atacante en lugar de introducir la contraseña del usuario, introduzca una sentencia SQL que le dé acceso a la aplicación. Por ejemplo la sentencia "*' OR 1='1* " en el campo contraseña le garantizaría el acceso.

¿Cómo lo solucionamos?

Para solucionar el error tenemos dos opciones: No permitir que el usuario utilice caracteres especiales en el usuario y/o contraseña mediante funciones PHP o javascript que eliminen esos caracteres del campo; o bien usar una función que le quite el valor especial a estos caracteres cuando introduzca los campos en la base de datos, al introducir contrabarras en los caracteres que pueden producir estos errores.

A2 – Secuencia de comandos en sitios cruzados (XSS)

¿Qué es?

> *Las fallas XSS ocurren cada vez que una aplicación toma datos no confiables y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.*

¿DÓNDE ESTÁ EL ERROR?

El error de XXS se comete al dejar introducir código malicioso en formularios, en nuestro caso al introducir una nueva noticia, se puede incluir código en JavaScript, por ejemplo, para redireccionar a sitios web maliciosos o simplemente para quebrar la seguridad de nuestra aplicación.

¿CÓMO LO SOLUCIONAMOS?

Para subsanar este error, usamos la clase "*class.inputfilter.php*" que elimina las etiquetas del campo para evitar modificar el código y por lo tanto no se puede introducir código malicioso, tanto en el registro como en el área de texto para introducir noticias.

A3 – Pérdida de Autenticación y Gestión de Sesiones

¿QUÉ ES?

> *Las funciones de la aplicación relacionadas a autenticación y gestión de sesiones son frecuentemente implementadas incorrectamente, permitiendo a los atacantes comprometer contraseñas, llaves, token de sesiones, o explotar otras fallas de implementación para asumir la identidad de otros usuarios.*

¿DÓNDE ESTÁ EL ERROR?

El error se encuentra tras insertar una noticia. Se pierde la sesión tras haber insertado la noticia y al volver a la página de noticias. Esto conlleva que no se pueda acceder de manera correcta a la página y nos redirija a la página principal.

¿CÓMO LO SOLUCIONAMOS?

Hemos añadido el tratamiento de sesiones, mediante la cual se pueda mantener identificado un usuario a pesar del cambio entre páginas. Con esto se consigue que cuando un usuario inserte una noticia, no pierda la sesión y tenga que identificarse de nuevo.

A7 –Almacenamiento Criptográfico Inseguro

¿QUÉ ES?

> *Muchas aplicaciones web no protegen adecuadamente los datos sensibles, tales como tarjetas de crédito, NSSs, y credenciales de autenticación con mecanismos de cifrado o hashing. Atacantes pueden modificar o robar tales datos protegidos inadecuadamente para conducir robos de identidad, fraudes de tarjeta de crédito u otros crímenes.*

¿DÓNDE ESTÁ EL ERROR?

Las contraseñas no están encriptadas y se pueden obtener mediante inyecciones de SQL (por ejemplo) de manera que queden expuestas y puedan ser usadas con mala intención.

¿CÓMO LO SOLUCIONAMOS?

Para solucionar este error se pueden utilizar funciones HASH para convertir la contraseña en un resumen criptográfico y almacenar este resumen de manera que sea imposible obtener la contraseña a partir de él. Luego, para saber si el usuario ha introducido correctamente su contraseña, se usa la misma función HASH y se comparan ambos resúmenes, el almacenado y el introducido por el usuario.

A8- Falla de Restricción de Acceso a URL

¿QUÉ ES?

> *Muchas aplicaciones web verifican los privilegios de acceso a URLs antes de generar enlaces o botones protegidos. Sin embargo, las aplicaciones necesitan realizar controles similares cada vez que estas páginas son accedidas, o los atacantes podrán falsificar URLs para acceder a estas páginas igualmente.*

¿DÓNDE ESTÁ EL ERROR?

El error consistía en que, mediante el método GET, se podía indicar a la página un usuario mediante la URL con lo que podías tener acceso sin tener permisos para acceder.

¿CÓMO LO SOLUCIONAMOS?

Esto se arregló con el uso de sesiones aplicado en el punto 2, evitando tener que detallar el usuario en la URL y evitando accesos indeseados.

Conclusión

Como se ha podido apreciar, estos errores son comunes y fáciles de cometer si no se tienen en cuenta, pero también son fáciles de subsanar, puesto que PHP contiene mucha variedad de funciones y clases que nos ahorran mucho trabajo y tiempo.

Personalmente nos ha parecido más difícil programar estos errores sabiendo que teníamos que cometerlos que programar la aplicación sin cometer los errores directamente.

Bibliografía

- Informe de la OWASP sobre las 10 vulnerabilidades web más importantes.
- Php.net