UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

DISEÑO ORIENTADO A OBJETOS

TAREA 5

SEGURIDAD EN APLICACIONES WEB

Maestro: Miguel Salazar

Salvador Rafael González García

Matrícula: 1690539

Grupo: 006

SEGURIDAD EN APLICAICONES WEB

Introducción

Dentro de la arquitectura de una aplicación web, se encuentran unas técnicas o aplicaciones que podemos utilizar para tener una comunicación entre un usuario o cliente y un servidor, entre las cuales se encuentran el uso de cookies, sesiones, hidden inputs, parámetros en la url, entre otras.

Por ejemplo, cuando se utiliza un servicio de Internet como el de consultar una base de datos, transferir un archivo, iniciar sesión de una cuenta, se establece un proceso de cliente/servidor. Ya que, mientras por un lado un usuario trata de ponerse en contacto con una PC para solicitar un servicio, el servidor responde lo solicitado mediante un programa que ejecuta.

Cookies

Las cookies son archivos que crea una aplicación web la cual tiene como contenido datos de comunicación entre un usuario y el servidor.

La finalidad de las cookies es el de identificar a un usuario en base a la actividad empleada en una página web, de modo que las configuraciones de este sitio sean un tanto personalizadas en cuanto a su uso.

Las funciones de las cookies varían dependiendo de su estructura, puesto que mientras unas sólo guardan datos como el de la última fecha en que ingresó el usuario a una página; asimismo hay páginas que cuentan con ventas en línea, por lo que las cookies actuarán para guardar los datos del carrito de compras.

Las cookies más comunes son las de sesión, estas tienen un corto lapso de vida ya que estas se borran cuando se cierra el navegador.

Las cookies persistentes son las que guardan información sobre el comportamiento de un usuario en un sitio web, estas se crean por un determinado tiempo. Las cookies persistentes pueden ser borradas limpiando los datos del navegador. Por ejemplo, cuando un usuario permanece en el estado de registrado después de un tiempo no razonable, tenemos un problema de registros persistentes.

Las secure cookies son usadas sólo en conexiones HTTPS. Estas almacenan información cifrada para evitar que los datos guardados sean vulnerables a algún tipo de ataque.

Las zombie cookies se recrean a sí mismas aun siendo borradas, esto sucede ya que el archivo se guarda en el dispositivo y no en el navegador ya que su objetivo es que se pueda acceder a ellas

sin importar el navegador que se esté usando. Lo anterior implica una amenaza para la seguridad y seguridad del usuario.

El usuario puede deshabilitar el uso de cookies cuando considere oportuno a través de las opciones de configuración/ajuste de su navegador, de tal forma que puede bloquear, restringir, deshabilitar o borrar la aceptación de cookies.

Otra modalidad de cookies son las de terceros, estas se generan cuando la página web que visitamos necesita pedir un elemento que utiliza otra página web, habitualmente son usadas por publicistas o hacer estudio de mercado. De igual manera estas configuraciones se pueden bloquear en los ajustes del navegador.

Sesiones

Java session o HttpSession sirve para guardar información entre diferentes peticiones HTTP. El problema de las sesiones radica que se comparten los datos del usuario entre un conjunto amplio de páginas web.

Esto sucede ya que la clase HttpSession contiene una estructura de HashMap (Diccionario), la cual le permite almacenar cualquier tipo de objeto con la finalidad de que estos puedan ser compartidos por las diferentes páginas por las cuales navegamos.

En cuanto su funcionamiento ocurre cuando un usuario crea una sesión accediendo a una página y está a la vez crea un objeto a nivel de Servidor con un HashMap vacío, el cual permite almacenar información relativa del usuario. Una vez creada la sesión se guarda una cookie que sirve para identificar al usuario y asociarle el HashMap creado para guardar la información obtenida. A este HashMap se puede acceder desde cualquier otra página permitiendo compartir información.

La predicción es un tipo de ataque de sesión el cual genera un identificador válido. Para ello, el atacante aprovecha los patrones de generación de identificadores de sesión que pueda utilizar el servidor y, una vez reducido el espacio de búsqueda, prueba todas las posibilidades posibles mediante fuerza bruta.

Como medidas de seguridad para una buena gestión de sesiones se encuentran las siguientes:

- Establecer un tiempo límite de vida para la sesión
- Regenerar el identificador de sesión cada cierto tiempo
- Detectar intentos de ataque de fuerza bruta con identificadores de sesión
- Requerir una nueva autentificación del usuario cuando vaya a realizar una operación importante
- Proteger los identificadores de sesión durante su transmisión
- Destruir la cookie al finalizar la sesión para evitar el acceso de otro usuario en un entorno público

Hidden inputs

Los hidden inputs o campos ocultos, son un elemento o una cadena de texto la cual está pensada para que esta no sea vista o editada por el usuario. Gracias a este atributo estamos enviando al programa de gestión de datos, aparte de los datos enviados por el propio usuario, datos predefinidos por nosotros mismos invisibles para el usuario

Los controles ocultos son especialmente útiles para enviar datos al servidor definidos por el autor, basados o no en la interacción con el usuario.

Este tipo de datos ocultos no se muestran en la página, aunque sí pueden ser detectados solicitando el código fuente. El atributo hidden no se llega a usar en páginas escritas en html, sólo en las que empleen también otro tipo de lenguajes.

De las ventajas que se encuentran en el uso del hidden input es que al ocultar el tipo de dato que se está transportando no es tan obvio como enviar el ID en una URL, por lo tanto se requiere de un conocimiento más elevado para efectuar ataques contra aplicaciones web que utilicen este tipo de estructura.

Además, permite a los usuarios almacenar el URL sin proveer información acerca del ID de sesión.

Por otra parte, aun si el navegador del usuario cuenta con una seguridad alta y del mismo modo tiene desactivado las cookies, estos elementos siguen permaneciendo ocultos.

Pero, no hay que dejar de lado que cualquier tipo de estructura o arquitectura de una aplicación puede contar con una o varias vulnerabilidades al momento de su creación, por lo tanto este tipo de campos ocultos pueden ser detectados por herramientas que estén disponibles en la red.

En cuanto a su programación puede resultar más compleja de lo habitual, ya que almacena una cantidad mayor de información acerca de la página, lo cual hace que sea más grande en cuanto a tamaño, por lo que al momento de la interacción del cliente y del servidor se tenga que enviar una cantidad mayor de información al navegador del cliente, dando una sensación de latencia.

El uso de los campos ocultos a mi parecer no se trata de una medida de seguridad, ya que cualquier "hacker" puede acceder y leer el código HTML de una cierta página web. Lo que sí se recomienda al hacer uso de estos es que se tenga una validación para acceder a la información que pueda ser enviada al servidor.

Parámetros en la URL

Una de las actividades que son usadas para acceder a diferentes rutas o sitios en donde se encuentra información contenida en una aplicación web, son las URL. La manipulación de estas puede permitir el accedso a zonas restringidas, cargar archivos malignos en el servidor, redirecciones a sitios terceros, entre otros.

Una forma de lograr esto es la inclusión de archivos locales. Es decir, cuando una aplicación web acepta cargar archivos recogiendo la ruta del archivo por la URL y retornando y mostrando el archivo en la página web.

La página web crea automáticamente los datos contenidos en la URL y, al navegar normalmente, el usuario simplemente hace clic en el vínculo. Esto da pie a que el usuario pueda modificar el parámetro manualmente probando diferentes valores en una URL.

Si el diseñador no ha previsto esta posibilidad, es posible que el hacker pueda tener acceso a un área que, en general, está protegida.

Para identificar alguna vulnerabilidad en el código por la manipulación de URL es la de aplicar pruebas manualmente, por lo que el programador debe buscar en todos los enlaces del sitio que los datos que son enviados cumplan con las restricciones requeridas, ya sea de cantidad de caracteres, espacio, tipo de archivos, entre otros.

Así que por lo que por cuestiones de seguridad es necesario o recomendable supervisar las vulnerabilidades con que cuente el código de una aplicación web, así como el de aplicar las actualizaciones proveídas por el editor del servidor web.

Además, tomar en cuenta la siguiente configuración:

- Nunca enviar por URL datos de acciones como: registros, ediciones, y borrado.
- Validar en el servidor que el usuario que intente acceder a una zona realmente cuente con los permisos adecuados para acceder a la misma.
- Impedir la navegación por páginas que estén bajo la raíz del página web.
- Deshabilitar la visualización de los archivos de un directorio que no contiene un archivo índice.
- Eliminar directorios y archivos inservibles.
- Asegurarse de que el servidor proteja el acceso a directorios que contienen datos importantes.
- Eliminar las opciones de configuración innecesarias.
- Asegurarse de que el servidor interprete las páginas dinámicas con precisión, incluso archivos de copias de seguridad.
- Eliminar los intérpretes de secuencias de comandos innecesarios.
- Impedir la visualización HTTP en páginas HTTPS accesibles.

Conclusión

Es pertinente que a la hora de programar o escribir el código de aplicación web, tener en cuenta las posibles vulnerabilidades que pueda tener este, ya que por exceso de trabajo, desconocimiento

o porque se deja parar por desapercibido, comprometen la arquitectura de un sitio web, dando entrada de que existan ataques por parte de personas malintencionadas. Esto, porque a medida que las líneas de código vayan implementando nuevas formas de automatización para la cuestión usuario, pueden dejarse de hacer cosas que pongan en peligro la interacción de los usuarios, quedando desprotegidos a la hora de que estos naveguen por la red.

De esta manera, los programadores tienen un papel importante en el momento de la creación de sitios web, ya que son el primer filtro para que se puedan evitar algún tipo de ataque hacia personas, empresas o cualquier individuo que tenga acceso a internet. Por lo que su trabajo a la hora de programar es fundamental en la protección tanto de archivo como de información personal y sensible.

Referencias

https://www.seguridad.unam.mx/historico/documento/index.html-id=17

https://blogthinkbig.com/que-son-las-cookies

https://www.arquitecturajava.com/usando-java-session-en-aplicaciones-web/

https://www.incibe.es/extfrontinteco/img/File/intecocert/Formacion/EstudiosInformes/gestion_sesiones_web_seguridad.pdf

http://xue.medellin.unal.edu.co/seguridad/2015/01/manipulacion-de-url-url-manipulation/