Facultad de Ciencias Físico Matemáticas

Nombre: Gabriel Omar Sanchez Reyes

Matricula: 1664322

Materia: Diseño orientado a objetos

Tarea 5

Técnicas para comunicar cliente-servidor:

Este reporte va a tratar de las diferentes técnicas de comunicar al cliente y al servidor, que son, como funcionan y algunos aspectos sobre su seguridad.

Cookies

Una cookie es un archivo creado por un sitio web que contiene pequeñas cantidades de datos y que se envían entre un emisor y un receptor. En el caso de Internet el emisor sería el servidor donde está alojada la página web y el receptor es el navegador que usas para visitar cualquier página web.

Su propósito principal es identificar al usuario almacenando su historial de actividad en un sitio web específico.

Su funcionalidad es la siguiente; cada vez que se visita una página web por primera vez, se guarda una cookie en el navegador con un poco de información. Luego, cuando se visita nuevamente la misma página, el servidor pide la misma cookie para arreglar la configuración del sitio y hacer la visita del usuario tan personalizada como sea posible.

El mayor problema de las cookies es la información que contienen. Cuando un usuario se conecta a un sitio web que puede ser personalizado, se le solicita que responda a varias preguntas a fin de crear un perfil y luego almacenar esta información en una cookie. Según el sitio web, la forma en que se almacenan estos datos puede resultar perjudicial para el usuario.

Lo ideal sería que una cookie contenga una cadena aleatoria (identificación de sesión) única, difícil de descifrar y válida sólo por un tiempo determinado. Sólo el servidor debería ser capaz de asociar las preferencias del usuario con el identificador de la sesión. De esta manera, una vez que la cookie expira, el identificador de sesión se vuelve inutilizable y no puede contener ningún dato que se relacione con el usuario.

Sesiones

Una sesión es la duración de una conexión empleando una capa de sesión de un protocolo de red, o la duración de una conexión entre un usuario (el agente) y un servidor, generalmente involucrando el intercambio de múltiples paquetes de datos entre la computadora del usuario y el servidor.

El funcionamiento del sistema de sesiones es relativamente sencillo. Cada vez que un usuario crea una sesión accediendo a una página (que la genere) se crea un objeto a nivel de Servidor con un HashMap vacío que nos permite almacenar la información que necesitamos relativa a este usuario. Realizado este primer paso se envía al navegador del usuario una Cookie que sirve para identificarle y asociarle el HashMap que se acaba de construir para que pueda almacenar información en él. Este HashMap puede ser accedido desde cualquier otra página permitiéndonos compartir información

Los atributos de sesión solo se ven dentro de nuestro JSP o Servlet mientras se está ejecutando. Una vez construida y enviada la respuesta al navegador, desaparecen. Su única utilidad es ayudarnos a construir la página que queremos mostrar en el navegador. Dentro del JSP serían variables normales de JSP.

Al emplear esta técnica hay muchos riesgos, por ejemplo el robo de sesión; HTTP es un protocolo sin estados, lo que significa que las credenciales o información de sesión deberá ir en cada petición; debido a esto dichos datos resultan muy expuestos. Un robo de estos datos podría tener como resultado que alguien se estuviera haciendo pasar por nosotros y realizando acciones con unos privilegios que nos pertenecen. Y tampoco hemos de olvidar que se pude robar la sesión intentando obtener nuestros credenciales de alguna manera (averiguar nuestra contraseña).

Para solucionar este problema de seguridad hay que atenerse a varios aspectos de la seguridad: la autentificación y la sesión; para cada uno de ellos veremos varios aspectos importantes a cubrir para solucionar problemas con el robo de sesión:

- El más importante de todos. Usar SSL sobre HTTP (HTTPS) para transferir los datos y asegurarse de que el cifrado cubre los credenciales y el ID de sesión en todas las comunicaciones. De esta manera los datos de sesión de los que hablábamos siguen estando expuestos pero esta vez se encuentran cifrados.

- Usar un sistema de autentificación simple, centralizado y estandarizado. Es mejor que usemos métodos de autentificación que nos proporcione el propio servidor de aplicaciones en vez de soluciones implementadas por nosotros.

- Posibilitar el bloqueo de autentificación después de un número determinado de intentos fallidos. Esto podría evitar ataques de fuerza bruta intentando averiguar la contraseña del usuario

- Implementar métodos seguros de recuperación de contraseñas: Es común que se intenten usar estos métodos para intentar ganar acceso a una cuenta del usuario, podemos ver una serie de consejos para implementar estos métodos. Pedir al usuario al menos tres datos o más, obligar a que responda preguntas de seguridad.

Hidden inputs

El elemento input, teniendo el valor "hidden" en su atributo type, representa cualquier cadena de texto arbitraria que no está pensada para ser vista o editada por 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.

Atributos específicos:

- Autocomplete: Una lista de identificadores de detalles de autocompletado describiendo el significado del valor provisto en el atributo value. Una lista de identificadores de detalles de

autocompletado puede estar compuesta por los siguientes identificadores en el orden especificado.

- Autofocus: valor booleano que instruye al navegador a establecer el enfoque sobre este control cuando el documento termina de cargarse o cuando el cuadro de diálogo (dialog) donde el control se encuentra es mostrado. Si el atributo tiene el valor "autofocus" o la cadena vacía (""), o si simplemente está presente, el control debería obtener el enfoque tan pronto como sea posible, luego de que la página o cuadro de diálogo hayan sido cargados.

- Disable: n valor booleano que indica si el control se encuentra deshabilitado o no. Si el atributo toma el valor "disabled" o la cadena vacía (""), o si está simplemente presente, el control estará deshabilitado.

- Form: El valor del atributo id del formulario con el que este control está asociado. Este atributo es nuevo en HTML y ayuda a definir la pertenencia de los controles en formularios anidados o distantes.

- Name: Un nombre para el control. Este nombre será enviado por el navegador al agente procesador, emparejado con el contenido del atributo value. Ambos atributos juntos conformarán un par nombre-valor que será utilizado para procesar la información del formulario.

- Type: Un valor que indica el tipo de campo que representa este elemento. Existen veintidós valores posibles (insensibles a mayúsculas/minúsculas):

- Value: Un valor para el control. Este valor será enviado por el navegador al agente procesador, emparejado con el contenido del atributo name. Ambos atributos juntos conformarán un par nombre-valor que será utilizado para procesar la información del formulario.

Como todas, esta técnica incluye vulnerabilidades y riesgos en cuanto se seguridad se trata; de esto hablaremos:

Manipulación de campos ocultos (Hidden field manipulation)

Algunos sitios y aplicaciones web incluyen campos ocultos para pasar información entre el servidor web y el navegador. Estos campos ocultos son representados por <input type="hidden">. Debido a malas prácticas de programación, algunas veces estos campos son utilizados para pasar información importante (como el precio de productos, tipos de usuarios, códigos de elementos a editar, entre otros). Información que debería estar almacenada únicamente en el servidor y no en el cliente. Normalmente los usuarios nunca ven estos datos, pero los hackers o usuarios maliciosos, fácilmente pueden descubrir estos datos y explotarlos siguientes los siguientes pasos:

- Ver el código fuente HTML de un sitio web: en Chrome por ejemplo, solo basta con dar click derecho en una página web y luego click en inspeccionar elemento. De esta manera se accede a todo el código HTML.

- Cambiar la información guardada en estos campos: simplemente se debe buscar el elemento oculto a editar (como por ejemplo un input donde se almacene el precio) y cambiarlo de 1000 a 10.

- Enviar la información: una vez se hicieron los cambios, se envía la información y se pueden comprar muchos ítems por un bajo precio.

Hay que las siguientes recomendaciones para prevenir este tipo de ataques:

- Nunca utilices campos ocultos para asignar datos delicados: por ejemplo nunca coloques en un campo oculto el tipo de un

usuario, ya que una manipulación de este campo colocaría a que cualquier usuario se registre como administrador.

- Valida que los campos cumplen con las restricciones necesarias nuevamente en el servidor, nunca confíes de las restricciones impuestas en los inputs o en el cliente.

- Cuando edites la información de un usuario que esta logueado maneja su información en SESSION, nunca coloques el ID en input hidden, siempre manéjala en SESSION.

- Verifica el código fuente utilizando la consola de Chrome tratando de verificar puntos débiles.

Parámetros en la URL

Los parámetros de URL constan de una clave y un valor separados por un signo igual (=) y unidos por el signo et (&). El primer parámetro siempre aparece después de un signo de interrogación en una URL.

Ayudan a controlar qué la URL de un sitio deben ser rastreadas por el robot de Google (en este caso), en función de los parámetros que aparecen en dichas URL. Esta función proporciona un método sencillo para evitar que se rastree contenido de un sitio por duplicado. Ahora es posible rastrear un sitio más eficientemente, reduciendo el uso de ancho de banda y probablemente permitiendo la indexación de más contenido único del sitio.

Usar esta función puede ser una buena idea en aquellos casos en que se sospecha que la cobertura de un sitio por parte del robot de Google no es todo lo buena que podría ser. Pero una gran capacidad conlleva una gran responsabilidad, Esta función solo se debe usar cuando se esté seguro del comportamiento de los parámetros de URL del sitio. En caso contrario, se podría

impedir, por error, el rastreo de algunas URL, con lo cual el contenido dejaría de estar accesible para el robot de Google.

Hay un ataque para este método de comunicación:

Ataque Semántico de la URL

Este tipo de ataque implica que el usuario modifique la URL con el fin de descubrir si se puede hacer algo. Por ejemplo, si el usuario Jorge pincha sobre un enlace de nuestra aplicación web y llega a: http://www.ejemplo.com/privado.php?usuario=jorge, se puede considerar que este usuario intentará ver qué pasa cuando el valor usuario cambie. Por ejemplo, podrá visitar la URL http://www.ejemplo.com/privado.php?usuario=luis para ver si puede tener acceso a la información de otra persona. Por esto, los datos GET son un target más frecuente para atacadores sobre todo nuevos.

Para todo tipo de método de comunicación hay vulnerabilidades y riesgos, aunque aún no se encuentren, por eso es que es importante ser muy precavidos y saber manejar bien todo este tipo de información.