2017

Diseño Orientado a Objetos

1595054
CARLOS ABEL LUGO GONZALEZ

FCFM | ciudad universitaria

Asignación 5

Ensayo sobre seguridad en aplicaciones web: comunicación entre cliente y servidor.

Para usar Internet de manera responsable y tener todo el control posible sobre nuestros datos, es importante saber qué información compartimos con los sitios web que visitamos.

La gran mayoría de los usuarios en Internet se han topado alguna vez en su camino con el término "cookies", y desde hace no mucho tiempo es casi imposible no haberse encontrado con un sitio en donde se muestre un aviso informando al internauta de que se almacenarán cookies en su navegador. Muchos probablemente cerraron el mensaje y no le dieron importancia a las "galletitas" que aceptaron.

Dado que nunca se está a salvo de los problemas de seguridad y privacidad que conlleva el uso de Internet, es importante saber qué es lo que estamos aceptando cuando decimos que no nos importa que un sitio web guarde cookies en nuestro navegador.

- ¿Qué son las 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, de manera que se le pueda ofrecer el contenido más apropiado según sus hábitos. Esto quiere decir que 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.

Estas cookies pueden tener una finalidad simple, como saber cuándo fue la última vez que el usuario entró a cierta página web; o algo más importante como es guardar todos los artículos puestos en el carrito de compras de una tienda, una acción que se va guardando en tiempo real.

¿Con qué finalidad se creó la primera cookie?

La primera cookie se creó en 1994 cuando un empleado de Netscape Communications decidió crear una aplicación de e-commerce con un carrito de compras que se mantuviese siempre lleno con los artículos del usuario sin requerir muchos recursos del servidor. El desarrollador decidió que la mejor opción era usar un archivo que se guardara en el equipo del receptor, en lugar de usar el servidor del sitio web.

- ¿Cuáles son los tipos de cookies más comunes?

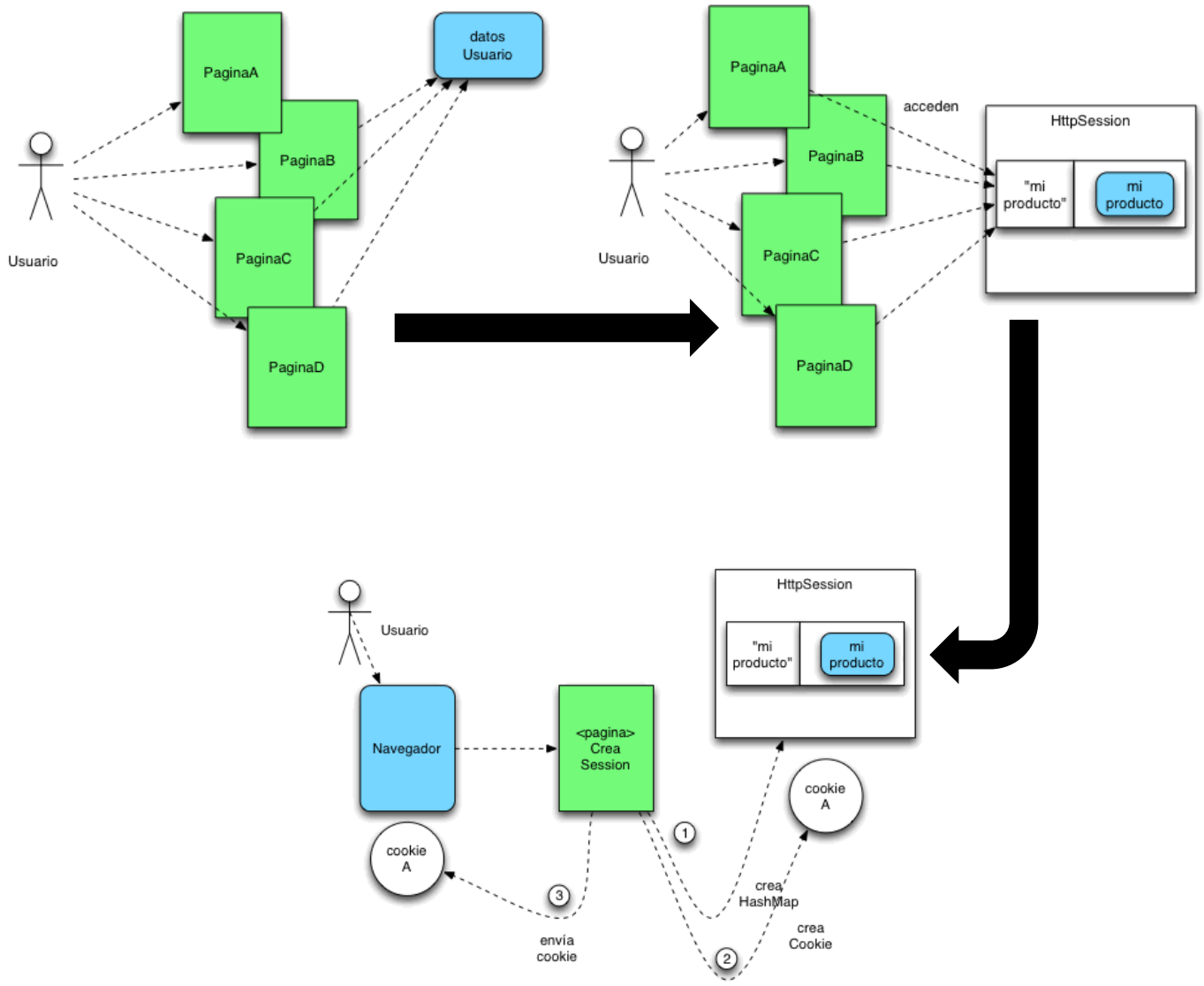
Las secure cookies o cookies seguras almacenan información cifrada para evitar que los datos almacenados en ellas sean vulnerables a ataques maliciosos de terceros. Se usan sólo en conexiones HTTPS.

Las zombie cookies son interesantes porque se recrean a sí mismas luego de que son borradas. Esto quiere decir que el navegador realmente no tiene ningún poder sobre ellas porque continuarán regenerándose, de ahí el nombre tan creativo que tienen. Las cookies zombis se guardan en el dispositivo y no en el navegador, usualmente con la finalidad de que se pueda acceder a ellas sin importar qué navegador se esté usando. Esta misma característica puede convertirlas en una amenaza para la privacidad y seguridad del usuario, y en muchas ocasiones son usadas con fines ilegítimos y malintencionados.

Uno de los conceptos que más problemas produce cuando comenzamos a trabajar con aplicaciones web en Java es el concepto de java session (HttpSession) que sirve para almacenar información entre diferentes peticiones HTTP ya que este protocolo es stateless (sin estado). Así pues en muchas ocasiones nos encontraremos con el problema de compartir estado (datos usuario) entre un conjunto amplio de páginas de nuestra Aplicación.

Java Session

Para solventar este problema en la plataforma Java EE se usa de forma muy habitual la clase HttpSession que tiene una estructura de HashMap (Dicccionario) y permite almacenar cualquier tipo de objeto en ella de tal forma que pueda ser compartido por las diferentes páginas que como usuarios utilizamos.



Funcionamiento

El funcionamiento del sistema de sessiones es relativamente sencillo. Cada vez que un usuario crea una session 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 r accedido desde cualquier otra página permitiéndonos compartir información.

Usuarios y Sessiones

El concepto de Session es individual de cada usuario que se conecta a nuestra aplicación y la información no es compartida entre ellos. Así pues cada usuario dispondrá de su propio HashMap en donde almacenar la información que resulte útil entre páginas.

¿Qué es una Sesión?

Una sesión es una serie de comunicaciones entre un cliente y un servidor en la que se realiza un intercambio de información. Por medio de una sesión se puede hacer un seguimiento de un usuario a través de la aplicación. El tiempo de vida de una sesión comienza cuando un usuario se conecta por primera vez a un sitio web pero su finalización puede estar relacionada con tres circunstancias:

-Cuando se abandona el sitio web.

-Cuando se alcanza un tiempo de inactividad que es previamente establecido, en este caso la sesión es automáticamente eliminada. Si el usuario siguiera navegando se crearía una nueva sesión.

Manejo de las Sesiones

En JSP las acciones que se pueden realizar sobre las sesiones se lleva a cabo mediante la interface HttpSession y los métodos que implementa. Esta interfaz está incluida dentro del paquete javax.servlet.http y es utilizada por el contenedor de páginas JSP para crear una sesión entre el servidor y el cliente. Para obtener la sesión de un usuario se utiliza el método getSession() que devuelve una interfaz de tipo HttpSession.

```
<%HttpSession sesion=request.getSession();%>
```

Una vez creado el objeto de tipo sesión es posible acceder a una serie de datos sobre la misma. Uno de estos datos es idSession que devuelve un identificador único asociado a una sesión:

```
<%HttpSession sesion=request.getSession(); out.println("IdSesion: "+sesion.getId()); %>
```

Es posible conocer el momento en el que se creó la sesión:

```
<%@page import="java.util.*" session="true"%> <% HttpSession sesion=request.getSession();
out.println("Creación: "+sesion.getCreationTime()); Date momento=new
Date(sesion.getCreationTime()); out.println("Creación: "+momento); %>
```

También se puede conocer la fecha y hora de la última vez que el cliente accedió al servidor con el que se creó la sesión, utilizando el método getLastAccesedTime():

```
<% Date acceso=new Date(sesion.getLastAccessedTime()); out.println("Último acceso:
"+acceso+"); %>
```

Teniendo en cuenta el momento en el que ser creó la sesión y la última vez que se accedió al servidor, se puede conocer el tiempo que lleva el

cliente conectado al servidor, o lo que es lo mismo el tiempo que lleva el usuario navegando por la páginas JSP:

```
<% long longDuracion=sesion.getLastAccessedTime(); sesion.getCreationTime();

Date duracion=new Date(longDuracion);

out.println("Duracion:"+duracion.getMinutes()+"min."+duracion.getSeconds()+"seg"); %>
```

La interfaz HttpSession ofrece el método isNew() mediante el cual es posible saber si la sesión creada es nueva o se está tomando de una previamente creada:

```
<% HttpSession sesion=request.getSession(); out.println("nueva: "+sesion.isNew()); %>
```

Si se ejecuta el ejemplo la primera vez el método devolverá true, ya que previamente no había ninguna sesión y ha sido creada en ese instante. Si se recarga la página devolverá false ya que la sesión ya ha sido creada. Las sesiones no necesitan ningún tipo de mantenimiento, una vez creadas no es necesario utilizarlas de forma obligatoria o tener que refrescar los datos asociados a las mismas, se deben ver como una variable más con la

diferencia que pueden ser utilizadas en cualquier página independientemente del lugar en el que hayan sido creadas.

Guardar objetos en una Sesión

Para guardar un objeto en una sesión se utiliza el método setAttribute(), que ha sustituido al método putValue(). Este método utiliza dos argumentos:

-El primero es el nombre que identificará a esa variable.

-El segundo es el dato que se va a guardar.

```
SetAttribute(java.lang.String name, java.lang.Object value)
```

Un ejemplo de cómo guardar una cadena de texto en la sesión:

```
<%@page import="java.util.*" session="true" %> <% HttpSession sesion=request.getSession(); sesion.setAttribute("trabajo","Paginas de JSP"); %>
```

Si se quiere pasar un parámetro que no sea un objeto es necesario realizar una conversión:

```
<%@page import="java.util.*" session="true" %>

<% HttpSession sesion=request.getSession(); Integer edad=new Integer(26); sesion.setAttribute("edad",edad); %>
```

Si se hubiera utilizado el valor entero en vez del objeto Integer, el resultado habría sido similar al siguiente. Incompatible type for meted. Can't convert int to java.lang.Object.

Seguridad

En caso de las cookies encontré un video que explica cómo se debe de proteger en el cual pondré el link a continuación:

https://www.youtube.com/watch?v=WPKLBgEF-PU

Identificador de sesión predecible

Un ataque de predicción de sesión consiste en intentar adivinar el identificador de sesión de algún usuario activo.

Dependiendo del algoritmo de generación de los identificadores de sesión la predicción será más o menos difícil. Cuanto más complejo e imprevisible sea el ID, mas difícil será la tarea. De ahí la insistencia en que el identificador sea totalmente aleatorio. Para que el atacante no tenga ningún patrón de partida.

Ataque:

El concepto de predicción o adivinación es algo engañoso. Ya que normalmente el código buscado no se acierta en un primer intento (excepto que el algoritmo de generación sea totalmente inseguro).

El ataque habitualmente consiste en probar mediante fuerza bruta una serie de IDs previstos, hasta dar con uno valido. La efectividad dependerá del número de intentos que haya que realizar hasta encontrar un identificador valido.

Si el identificador es totalmente aleatorio, el número de intentos necesarios será igual a todo el rango de posibles valores del identificador. En este caso será también importante la longitud del ID de sesión, ya que si el rango de posibles valores no es lo suficientemente grande, el atacante podría intentar probar todos los posibles valores.

Patrones vulnerables:

Tenemos por tanto que el éxito de un ataque de predicción de sesión depende de:

1. La longitud del ID. O sea el rango de todos los posibles valores.
2. El patrón de generación de este ID.
3. Los patrones que se suelen emplear para la generación de sesiones y que son predecibles, de menor a mayor dificultad de análisis:
4. Fijo: El ID tiene un valor fijo. No es necesario adivinarlo.
5. Basado en datos del usuario: Su nombre, su dirección IP. Sera diferente para distintos usuarios, pero constante para el mismo cliente. Puede estar ofuscado para complicar el análisis.

6. Rango de valores: El ID se elige entre un rango de valores prefijado. Tras un periodo de observación la predicción es trivial.
7. Basada en tiempo: Cada nuevo ID se genera incrementando el anterior en base a un reloj. La predicción es fácil.
8. Incremental: Similar al basado en tiempo, pero que solo se incrementa con cada nuevo ID generado. Si la aplicación tiene mucho tráfico será más complejo de atacar.
9. Pseudo-aleatorio: La generación intenta ser aleatoria, pero el algoritmo contiene vulnerabilidades que permiten predecir su comportamiento. Normalmente el atacante para tener éxito necesitara obtener los datos de partida que usa el algoritmo para generar la serie de IDs.

Protección

Ofuscar o combinar varios de estos patrones puede dificultar el análisis, pero la única protección efectiva consiste en generar de forma aleatoria los ID y utilizar un rango de valores lo suficientemente grande.

Bibliografía

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

http://dis.um.es/~lopezquesada/documentos/IES_1213/IAW/curso/UT5/ActividadesAlumnos/12/sesiones.html

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

https://www.osi.es/es/actualidad/blog/2015/09/11/te-explicamos-la-relacion-entre-las-cookies-y-tu-privacidad-mientras-navegas

https://www.youtube.com/watch?v=WPKLBgEF-PU

https://www.s21sec.com/es/blog/2009/02/seguridad-en-el-manejo-de-sesiones-web-v/