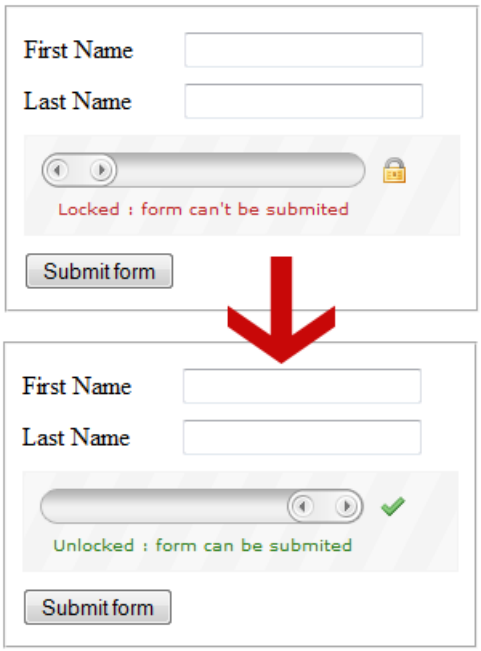
Alternativas al método Captcha



Introducción

Una compañía de Telefónica

acens
the Cloud services company

En muchas ocasiones nos habremos encontrado delante de **un formulario en una página web con un campo donde debemos introducir una serie de caracteres**, en ocasiones casi ininteligibles, para poder realizar el envío del formulario. Se trata de los conocidos captcha, cuyo principal objetivo es distinguir si el que va a realizar el envío del formulario es una persona o un "robot informático", con el objetivo de evitar el envío de spam en los blogs, foros o cualquier otra aplicación web.

El nombre de Captcha corresponde a las siglas 'Completely Automated Public Turing test to tell Computers and Humans Apart' (test público de Turing completamente automatizado para distinguir los ordenadores de los humanos) en honor a la prueba que creó en 1950 el matemático Alan Turing, y que viene a demostrar que las personas son capaces de realizar ciertas tareas mejor que cualquier máquina, siendo una de estas tareas el reconocimiento de imágenes. Aunque hoy en día ya hay aplicaciones que son capaces de indicar las letras que aparecen en las imágenes con un alto porcentaje de acierto.

El término Captcha se empezó a utilizar en el año 2000 por Luis von Ahn de la Universidad Carnegie Mellon, y pronto su uso se extendió a grandes empresas del sector informático. Hoy en día es raro encontrar un formulario web que no haga uso de este sistema.

Pero el Captcha tiene sus inconvenientes, ya que en muchas ocasiones puede llegar a frustrar al usuario si tras varios intentos no es capaz de dar con la palabra correcta que aparece en el recuadro. O bien si queremos que el formulario no tenga demasiados campos, para que el usuario lo rellene rápidamente.

ReCaptcha. Evolución y negocio

Poco después de desarrollar el método Captcha para Internet, su creador Luis von Ahn evolucionó ese sistema creando el método reCaptcha, un sistema que muchos de vosotros conoceréis ya que es uno de los más utilizados hoy en día y que se caracteriza porque en vez de aparecernos una palabra, son dos las que tiene que introducir el usuario.

Lo curioso de este sistema, es que sólo una de esas dos palabras es la que pertenece a la base de datos de los Captcha correctos. Pero entonces, ¿qué pasa con la segunda? Aquí es donde viene lo llamativo, ya que Luis von Ahn creó este sistema para digitalizar libros antiguos que por culpa del paso del tiempo los escáneres más potentes no eran capaces de digitalizar.

Lo que hizo von Ahn fue introducir en otra base de datos todas aquellas palabras que no habían podido ser digitalizadas. De esa base de datos saca la segunda palabra. El programa realiza un control de esas dos palabras, y si el usuario introduce la palabra perteneciente al Captcha correctamente, el sistema ya descarta que sea una máquina, y da como válida la otra palabra (la correspondiente al libro antiguo), que pasa a ser digitalizada.

Gracias a este original sistema, entre todos los usuarios del mundo estamos ayudando a digitalizar miles de libros cada año.

Alternativas a los tradicionales Captcha

En los últimos años, la aparición de programas capaces de adivinar con un elevado porcentaje de acierto las letras y números que aparecen en los cuadros de texto de este sistema ha provocado el desarrollo de otras estrategias, también de reconocimiento, para combatir el spam y cualquier tipo de información no deseada.

Veamos a continuación algunos métodos antispam desarrollados como alternativa al código Captcha.

1.- Juegos y puzles

Una de las alternativas que han aparecido es el uso de juegos o puzles para verificar que se trata de un humano. En este caso se muestra al usuario de la web un sencillo juego que debe resolver para verificar que no se trata de un robot, como introducir una serie de objetos en un recipiente, plantar un jardín o colocar los cubiertos en una mesa, por ejemplo.

2.- Operaciones matemáticas

En este caso el funcionamiento es muy sencillo, ya que el usuario tendrá en su pantalla una expresión matemática que deberá resolver para demostrar que no se trata de un robot. Las operaciones pueden ser desde simples sumas o restas de dos números, hasta complejas ecuaciones matemáticas.

3.- Reconocimiento de imágenes

Este método antispam, en vez de mostrar una imagen con un texto, presenta varias imágenes reales. Luego, sobre esas imágenes, se realizará una pregunta simple. Por ejemplo, en la imagen superior se muestran dos animales y para verificar que eres un humano deberás de introducir el nombre de esos animales.

La fortaleza de este método radica en que para los robots es muy complicado reconocer imágenes y más aún resolver una pregunta sobre esas imágenes.

4.- Uso de plugin para jQuery

jQuery es una de las librerías de scripts más utilizadas en la actualidad gracias a su sencillez. Otra de las ventajas de utilizar esta librería son los muchos plugins que nos podemos encontrar creados y que aportan funcionalidades extras.

Entre los muchos plugins que nos podemos encontrar, también hay algunos creados para luchar contra los spammers. Entre estos plugins podemos destacar:

a) Plugin QapTcha

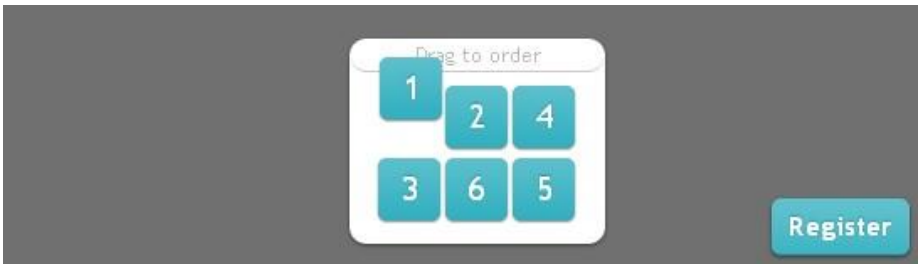
Este plugin lo que nos permite crear es un **botón deslizante el cual hay que mover hacia la derecha** para poder hacer el envío del formulario. Mientras que este elemento no es desplazado, el envío permanece bloqueado.

b) Ajax Fancy Captcha

El funcionamiento de este sistema es muy intuitivo, ya que para poder enviar el formulario el usuario deberá arrastrar uno de los iconos hacia el círculo, pero no cualquier icono, sino el que le indique el sistema. Por ejemplo, en el caso de la imagen superior, se pedía mover el icono correspondiente al "lápiz".

Este sistema es tan intuitivo que es una **buena opción para utilizar en sitios infantiles o bien para personas de edad avanzada**, que pueden tener problemas a la hora de leer los caracteres que aparecen en un captcha tradicional.

c) jQuery Fancy Draggable Captcha



En este caso, el usuario deberá ordenar los números o letras que aparezcan. De esta forma verificará que se trata de un humano y no de un robot informático.

d) Motion Captcha

Este plugin de jQuery lo que añadirá al formulario será un trazo que deberemos **sobrepintar con el ratón**. Dependiendo de lo parecido que sea el trazo con el original, se determinará si es un humano quien lo ha hecho o bien un robot.

Método HoneyPot

Si los métodos vistos en el punto anterior no te gustan porque estás cansado de que los usuarios tengan que interaccionar con la página para poder realizar el envío, te presentamos un método donde no hará falta que el usuario haga nada.

Este sistema recibe el nombre de "Honey Pot" y su funcionamiento es muy sencillo. La idea consiste en **añadir a nuestro formulario un campo de texto extra y ocultarlo mediante el uso de CSS**. De esta forma, los usuarios que visiten el portal no lo verán y no lo rellenarán. Por el contrario, un robot automático sí que lo completará. Antes de enviar el formulario sólo deberemos comprobar que ese campo oculto está vacío. Si no es así, el envío de la información se cancela.

Veamos a continuación un ejemplo de uso. Lo primero que deberemos hacer será crear nuestro formulario:

```
<form method="post" action="enviar.php">
    <label for="nombre">Nombre:</label> <input name="nombre" value="" size="20" />
    <label for="email">Email:</label> <input name="email" value="" size="25" />
    <label for="verificacion" class="oculto">¡Si ves esto, no completes el siguiente campo!</label> <input name="verificacion" class="oculto" />
    <input type="submit" value="Enviar" />
```

```
</form>
```

Como podéis observar en el código anterior, hemos añadido tres campos de texto: dos de ellos válidos como son el nombre y el email, y un tercero al que hemos llamado verificación que será el campo oculto y al que le hemos asignado la clase “oculto”. Lo hemos puesto así en el ejemplo para que lo veáis más claro, pero es recomendable llamarlo de otro modo para no dar pistas a los robots.

Lo siguiente será ocultar ese campo verificación para que los usuarios no lo vean. Para ello utilizaremos propiedades CSS que aplicaremos a la clase “oculto”.

```
.oculto{
        display: none;
}
```

Por último, y ya en el archivo encargado de realizar el envío, comprobamos que ese campo oculto venga vacío. En nuestro ejemplo esta acción se realizará en el archivo “enviar.php”.

```
<?php
if ($_POST['verificacion'] != ""){
  // Es un Robot
  exit();
}else{
  // Es un usuario real
}
?>
```

Como te indicamos antes, este método también es útil porque ahorra rellenar un campo al usuario, por tanto es más probable que lo cumplimente. **En este formulario del informe SEO gratis de acens tienes un ejemplo real de uso del método HoneyPot**.

Método Timegate

Otro método novedoso es el 'timegate'. Se trata de programar que si el formulario se rellena en menos de X segundos (por ejemplo 5) desde que el usuario empieza a escribir, no se da por válido porque un humano no puede tardar tan poco en rellenarlo, y por tanto se trataría de una máquina.

Además de utilizar algunos de los métodos que hemos visto a lo largo de este White Paper, es conveniente combinarlos con algún tipo de filtrado a nivel de navegador o bien a nivel de servidor, para reducir lo máximo posible el envío de spam, ya sea bloqueando direcciones IP desde donde se hace el envío o filtrando por palabras claves.