

Unidad 5

Captcha

CAPTCHA

Captcha es el acrónimo de **Completely Automated Public Turing test to tell Computers and Humans Apart** (Prueba de Turing pública y automática para diferenciar máquinas y humanos).

Se trata de una prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano. El término se empezó a utilizar en el año 2000 por Luis von Ahn, Manuel Blum y Nicholas J. Hopper de la Carnegie Mellon University, y John Langford de IBM.



Prueba de Turing

La típica prueba consiste en que el usuario introduzca un conjunto de caracteres que se muestran en una imagen distorsionada que aparece en pantalla. Se supone que una máquina no es capaz de comprender e introducir la secuencia de forma correcta por lo que solamente el humano podría hacerlo.

Como el test es controlado por una máquina en lugar de un humano como en la Prueba de Turing, también se denomina Prueba de Turing Inversa.

El Test de Turing (o Prueba de Turing) es una prueba propuesta por Alan Turing para demostrar la existencia de inteligencia en una máquina. Fue expuesto en 1950 en un artículo (Computing machinery and intelligence) para la revista Mind, y sigue siendo uno de los mejores métodos para los defensores de la Inteligencia Artificial. Se fundamenta en la hipótesis positivista de que, si una máquina se comporta en todos los aspectos como inteligente, entonces debe ser inteligente.

La prueba consiste en un desafío. Se supone un juez situado en una habitación, y una máquina y un ser humano en otras. El juez debe descubrir cuál es el ser humano y cuál es la máquina, estándoles a los dos permitido mentir al contestar por escrito las preguntas que el juez les hiciera. La tesis de Turing es que si ambos jugadores eran suficientemente hábiles, el juez no podría distinguir quién era el ser humano y quién la máquina. Todavía ninguna máquina puede pasar este examen en una experiencia con método científico.

En 1990 se inició un concurso, el Premio Loebner, una competencia de carácter anual entre programas de ordenador que sigue el estándar establecido en la prueba de Turing.

Un juez humano se enfrenta a dos pantallas de ordenador, una de ellas que se encuentra bajo el control de un ordenador, y la otra bajo el control de un humano. El juez plantea preguntas a las dos pantallas y recibe respuestas. El premio está dotado con 100.000 dólares estadounidenses para el programa que pase el test, y un premio de consolación para el mejor programa anual. Todavía no ha sido otorgado el premio principal.

Pasos para la creación de un Captcha

Estos son los siete pasos básicos para generar un sistema de Captcha:

1. Generar un código aleatorio
2. Añadir el código a una cookie, variable de sesión ó base de datos que será recuperada desde la siguiente página
3. Escribir el texto dentro de una imagen
4. Mostrar la imagen al usuario que quiera acceder al recurso
5. Proveer de un formulario, donde el usuario ingrese el código y lo envíe
6. Verificar el código de la clave enviada (en el paso 2)
7. Si el código es correcto, se permite el acceso

Generalidades

Todos estamos cansados de ver imágenes distorsionadas, generalmente con un texto a descifrar e ingresar en ciertos sitios.

En realidad, un CAPTCHA es un programa que debería generar una prueba desafío respuesta que los humanos seamos capaces de resolver pero que las máquinas no.

Esto que puede parecer algo absolutamente inútil ha cobrado importancia con el auge del spam y el abuso de los recursos ajenos por partes de personas inescrupulosas.

Cuando el spam alcanzó los niveles actuales y la invasión de los correos electrónicos ya no era un desafío para los spammers, los mismos comenzaron a explotar nuevas vías de promoción de forma tal de llevar a los usuarios a los sitios de venta de los productos que publicitan o bien lograr ser indexados mayor cantidad de veces por los buscadores.

La proliferación de blogs y la posibilidad de enviar mensajes a los mismos, sin registración y con formularios que no requieren más que el ingreso de texto proporcionó a los spammers la forma perfecta de lograr sus objetivos.

Con el desarrollo de simples bots (Un bot (abreviatura de robot) es un programa informático que realiza funciones muy diversas, imitando el comportamiento de un humano), se rastrean millones de blogs enviando direcciones de los sitios web promocionados por los spammers. Esta acción se conoce como Splog (Un splog es un blog creado con el único fin de promocionar sitios web afiliados, mejorar la posición de los mismos ante los buscadores, para publicar únicamente avisos publicitarios que pagan por cantidad de visitantes. El contenido de estos blogs es generalmente

texto escrito específicamente para obtener mayores ganancias con AdSense o también textos copiados desde otros sitios web. El término splog es un neologismo que viene de contraer la expresión spam blog del idioma inglés.).

Como se mencionó anteriormente la ventaja radica en dos puntos importantes para el spammer:

Si el blog es importante y muy visitado, miles de usuarios podrán ver los enlaces y hacer click en ellos.

Si los buscadores indexan este enlace, el mismo puede ser considerado de relevancia directamente proporcional a la importancia del blog del que fue obtenido. Si bien los buscadores implementan algoritmos (y tags HTML) para evitar estos sitios, los mismos son burlados y es común ver sitios de spammers con los buscadores.

Esta última metodología también puede ser aplicada a sitios tan populares como la Wikipedia si bien se implementan métodos para evitarlos, como el tag "noindex" que le dice a los buscadores que no indexen los enlaces contenidos en la página.

Los CAPTCHAs han sido, por un tiempo, una aproximación bastante acertada a la solución del problema hasta que los spammers encontraron la forma de saltar algunos de ellos.

La forma de pasar un CAPTCHA es el reconocimiento óptico de caracteres (OCR), la misma técnica actualmente utilizada desde hace años en los scanners y la misma utilizada para detectar spam gráfico en los correos electrónicos.

Es así que actualmente pueden verse decenas de CAPTCHA distintos, algunos de los cuales suelen ir desde soluciones muy sencillas, por lo cual no cumplen su función, a extremadamente difíciles de descifrar incluso para humanos. Esto sin mencionar que las personas con deficiencias visuales tienen problemas con su reconocimiento.

Es importante remarcar que la "fortaleza" de un CAPTCHA puede ser distinta según la utilización del mismo: no tiene la misma importancia el registro a un sitio web como paypal que un formulario de contacto en un weblog.

A modo de ejemplo podemos citar proyectos en los cuales el objetivo es mejorar los CAPTCHAs actuales, mostrando debilidades, fortalezas, eficacia e ineficiencia de muchos de los métodos utilizados.

Otras corrientes prefieren variar estos métodos mediante la utilización de algunos de los siguientes métodos:

- Resolver un problema del tipo matemático - ¿Cuánto es siete menos cuatro? Presenta problemas con el idioma y por alguna razón es rechazado por muchos usuarios.
- ¿Cuál de estas personas es mayor? y se muestran un bebé y una persona adulta. Presenta problemas en las personas con deficiencias visuales.
- Responder preguntas del tipo ¿cuál es la temática de este sitio? Pueden presentar problemas con el idioma o con la interpretación de la pregunta realizada.
- Establecer la relación que guardan ciertas imágenes. Mostrar cuatro imágenes de gatos y la persona debe seleccionar "gato" desde una lista desplegable. Presenta los mismos problemas del anterior y además algunas imágenes pueden ser difíciles de relacionar.
- Establecer relación entre imágenes y caracteres en un captcha complejo. Pueden presentar los problemas de los anteriores e incluso llegar a ser muy complejos.
- Resolución de Puzzles y rompecabezas. Pueden ser complejos. Presenta problemas en las personas con deficiencias visuales.
- Crear CAPTCHAs auditivos. Presentan problemas con las personas con discapacidades auditivas.

Si bien las alternativas son muchas, algunos profesionales afirman que este no es el camino correcto debido a los problemas de accesibilidad que presentan y que cualquier método técnico a la larga es "hackeable" por un robot (o por una máquina que pase el Test de Turing).

Más allá de no ser la solución final, lo cierto es que actualmente son un filtro importante en la mayoría de los sitios webs y todo parece indicar que los CAPTCHAs nos acompañaran por un tiempo más hasta que se perfeccionen algunos métodos de detección heurísticos.

Pasemos a un ejemplo sencillo:

Lo primero que necesitamos para mostrar un CAPTCHA es crear aleatoriamente la secuencia de caracteres que va aparecer en la imagen y guardarla en una variable de sesión (\$_SESSION) para comprobarla luego con la ingresada por el usuario y un formulario para que el usuario ingrese la clave.

```
<?php
session_start(); // Iniciamos la sesión
?>
<!DOCTYPE html PUBLIC>
<head>
<meta charset="UTF-8" />
<title>Ejemplo Captcha</title>
</head>
<body>
```

```
<?php
$numero1= rand(0,9);
$numero2= rand(0,9);
$numero3= rand(0,9);
$minus =
array("a","b","c","d","e","f","g","h","i","j","k","l","m","n","ñ","o","p","q","r","s","t","u","v",
"w","x","y","z");
$mayus =
array("A","B","C","D","E","F","G","H","I","J","K","L","M","N","Ñ","O","P","Q","R","S","T","U",
"V","W","X","Y","Z");
$signos = array("!", "#", "$", "%", "&", "=");
$generador_min = rand(0,26);
$generador_may = rand(0,26);
$generador_sig = rand(0,5);
$_SESSION["codigos"] =
($numero1).($minus[$generador_min]).($numero2).($mayus[$generador_may]).($signos[
$generador_sig]).($numero3);
// Guardamos el texto del CAPTCHA en la session
print "<img src=imagen2.php>";
?>
<form action="confirmacion2.php" method="post"> Escriba codigo de seguridad:
<label>
<input name="confirmacion" type="text" id="confirmacion" />
</label>
<p>
<label>
<input type="submit" name="Submit" value="Comprobar" />
</label>
</p>
</form>
</body>
</html>
```

Dentro del script anterior, encontramos una llamada a una archivo php que genera dinámicamente la imagen tal cual viéramos en las unidades anteriores. Lo transcribimos a continuación:

```
<?php session_start();
header ("Content-type: image/jpeg");
$im = @imagecreate(100, 30);
$color_fondo = imagecolorallocate ($im, 240, 240, 240);
```

```
$color_texto = imagecolorallocate ($im, 0, 128, 6);  
imagestring ($im, 25, 25, 5, $_SESSION["codigos"], $color_texto); imagejpeg ($im);  
?>
```