Universidad
Carlos III de Madrid

Trabajo Fin de Grado

Grado en Ingeniería Informática

Especialidad: Sistemas de la Información

CAPTCHAS. DEBILIDADES Y FORTALEZAS

Autor: Patricia Román Escabias

Tutor: Arturo Ribagorda Garnacho

Madrid, 24 de junio de 2013

Título: CAPTCHAS. Debilidades y fortalezas.

Autor: Patricia Román Escabias
Director: Arturo Ribagorda

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Trabajo Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Llegados a este punto es inevitable pensar en el principio, en el principio de la carrera, hace cuatro-cinco años y ahora estamos casi acabando, este es el último paso para terminar esta etapa que, al menos en mi caso, ha sido de las mejores.

Haciendo repaso de todo, recuerdo a mis primeros compañeros, algunos se fueron, otros han continuado por otro camino, y otros han llegado conmigo hasta el final. Muchas de esas personas no son sólo compañeros, sino que también son amigos, a esas personas quiero agradecerles todo el apoyo que me han dado durante estos años. Hemos vivido muchos momentos buenos y malos, pero pesan más los buenos, sin duda. En especial quiero nombrar a Marta Parrilla y José Luis Garrido, que son lo mejor que me llevo de esta etapa. Ellos han estado conmigo tanto en mi mejor, como en mi peor momento de la carrera, y les debo mucho.

También quiero agradecer a mi familia todo el apoyo que me han dado durante estos años de carrera, estos meses de último trabajo y durante toda mi vida, ya que siempre han estado ahí, sobre todo mis padres, mi hermana y mi abuela.

Además, quiero agradecer a mis amigas que siempre están ahí, porque todo el mundo necesita despejarse, salir de la rutina y de todo lo referente a los estudios; ellas están ahí en las buenas y en las malas. En especial, nombrar a mi principal apoyo, que desde pequeñas ha estado conmigo, dándome su ayuda y cariño incondicional, mi amiga Irene.

Por último, quiero agradecer su apoyo a algunos profesores, algunos de ellos se han preocupado de verdad por nosotros, estando ahí cada vez que hemos necesitado su ayuda, por ejemplo para tutorías en momentos de máximo agobio. Y entre todos los profesores, quiero reconocer en particular a Arturo Ribagorda, mi tutor de proyecto, que me ha acompañado a lo largo de la realización del mismo.

Esta etapa termina pero comienza una nueva, y sé que la mayoría de las personas nombradas también estarán en ella.

Resumen

En el presente trabajo se tratará el tema de los CAPTCHAS, es decir, esas pruebas que suelen aparecer cuando queremos registrarnos en una página, en una encuesta, descargarnos algún tipo de fichero, etc. La funcionalidad de estos es diferenciar si se está tratando con un ser humano o con una máquina y, de esta forma, evitar que un robot sea capaz de realizar este tipo de acciones en la web.

Las pruebas de los CAPTCHAS se basan en problemas abiertos de Inteligencia Artificial. Por lo tanto, estas pruebas también estarían relacionadas con la rama de la IA, lo que lleva a investigar y trabajar en su seguridad.

Hoy en día, el tema de los CAPTCHAS como prueba de seguridad, no es 100% fiable, y es que, a pesar de los intentos de crear uno, que verdaderamente pruebe que se está tratando con un ser humano, y no con una máquina, no se ha llegado a encontrar dicho CAPTCHA "perfecto". Existen numerosas herramientas que son capaces de saltar, en cuestión de minutos, las pruebas consideradas más actuales y desarrolladas.

En la siguiente imagen vemos una realidad, y es que cada vez los CAPTCHAS son más "complicados", es decir, no son fáciles de ver y resolver por un usuario cualquiera, a golpe de vista. Esto se hace con el fin de que también sea más complicado de resolver por una máquina, pero la realidad, es que la mayoría de las veces, es ineficiente.

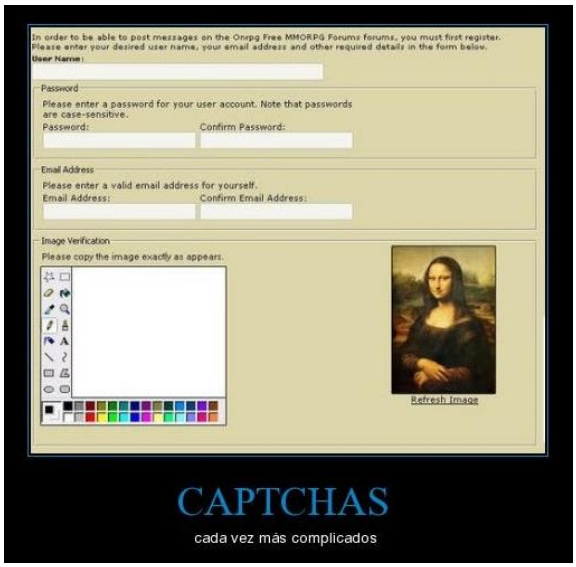

Ilustración 1

En este estudio se verán los diferentes CAPTCHAS que existen hoy en día, con sus principales fortalezas y debilidades. Se verán desde ejemplos muy sencillos, como introducir una determinada cadena de caracteres, hasta los más complicados, tanto para usuarios como para máquinas, que consisten en resolver una fórmula matemática o un puzle.

Además veremos algunas de las principales herramientas anti-CAPTCHAS, y se crearán y comentarán nuevas ideas de CAPTCHAS, que serían, a día de hoy, muy seguros con respecto a estas herramientas que intentan romperlos.

Abstract

In this paper, we address the idea of CAPTCHAS, they are tests that usually appear when we check into a page, in a survey, download any file, etc. The functionality of these test, is tell you whether you're dealing with a human or a machine, and then, you can prevent robot can perform those actions on the web.

CAPTCHA tests are based on open problems in Artificial Intelligence. So, these tests are also linked to the branch of AI, and this leads to research and work on their safety.

Today, CAPTCHAS, like security test, are not 100% reliable, despite attempts to create a CAPTCHA, which really proves that we are dealing with a human, not a machine, it has not been found, it has not been created the "perfect" one. There are many tools that are able to break in minutes the most current and developed test.

The next picture shows a reality, today, CAPTCHAS are more and more “complicated”, they are not easy to see and resolve any user, at a glance. This is because, developers want CAPTCHAS are difficult for bots too, but actually, this is often inefficient.

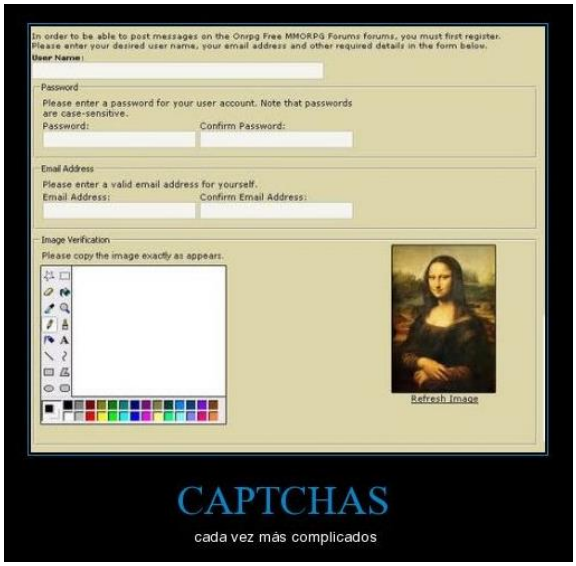

Ilustración 1

In this study, it will be seen different CAPTCHAS that exist today, with its main strengths and weaknesses. It will be seen CAPTCHAS simple, like inserting a particular character string, and CAPTCHAS more complicated, for users and machines, which consist of solving a mathematical formula or a puzzle.

Furthermore, it will be seen some anti-CAPTCHAS tools, and it will be create and comment new CAPTCHAS ideas. These ideas are, today, very safe with respect to anti-CAPTCHAS tools views.

Índice

1. Introducción 10
1.1. Objetivo 10
1.2. Motivación 10
1.3. Contenido de la memoria 10
2. Presentación de los CAPTCHAS 11
2.1. ¿Qué son? 11
2.2. La historia de los CAPTCHAS 14
2.2.1. Los peores CAPTCHAS de la historia 14
2.3. Aplicación 17
2.4. Problemas que causan los CAPTCHAS 18
2.5. CAPTCHAS y Pruebas de Turing 19
2.5.1. Pruebas de Turing 19
2.5.2. Inteligencia Artificial 20
3. Tipos de CAPTCHAS 21
3.1. Imágenes 21
3.1.1. ASIRRA 21
3.1.2. HumanAuth 23
3.1.3. HotCaptcha 24
3.2. Audio 25
3.2.1. Nueva generación de CAPTCHAS de Audio 26
3.3. Video 26
3.3.1. NuCaptcha 26
3.4. Matemáticos 28
3.5. Mini-juegos 29
3.5.1. Puzle-CAPTCHA 30
3.6. ReCAPTCHA 31
4. Anti- CAPTCHAS 33
4.1. Introducción 33
4.2. Segmentación 33
4.3. Herramienta ENT 35
4.3.1. Ataque a ASIRRA 36
4.3.2. Ataque a HumanAuth 38

4.4. Ataque a los CAPTCHAS de Hotmail y Gmail 39
4.5. DeCAPTCHA 40
4.6. Ataque al CAPTCHA de video 41
4.7. Ataque a MAPTCHA 43
4.8. JDownloader burla los CAPTCHAS 44
5. Desarrollo de un CAPTCHA 45
5.1. Consejos para crear un buen CAPTCHA 45
5.2. Propuesta de CAPTCHA 1 49
5.2.1. El CAPTCHA 49
5.2.2. ¿Cómo crear este CAPTCHA? 50
5.2.3. ¿Por qué este CAPTCHA es seguro? 52
5.3. Propuesta de CAPTCHA 2 53
5.3.1. El CAPTCHA 53
5.3.2. ¿Cómo crear este CAPTCHA? 55
5.3.3. ¿Por qué este CAPTCHA es seguro? 56
5.4. Propuesta de CAPTCHA 3 58
5.4.1. El CAPTCHA 58
5.4.2. ¿Cómo crear este CAPTCHA? 60
5.4.3. ¿Por qué es seguro este CAPTCHA? 61
5.5. Debilidades de los CAPTCHAS creados 61
5.5.1. CAPTCHA 1 61
5.5.2. CAPTCHA 2 62
5.5.3. CAPTCHA 3 62
6. Conclusiones y trabajos futuros 63
Referencias 64
Anexo 1: Terminología 67
Glosario de términos 67
Abreviaturas 67
Anexo 2: Presupuesto del Proyecto 68
Costes de personal 68
Costes de material 69
Viajes y dietas 70
Costes totales 71
Cantidad total 72

Índice de Imágenes

Ilustración 1 3
Ilustración 2 11
Ilustración 3 12
Ilustración 4 12
Ilustración 5 13
Ilustración 6 15
Ilustración 7 15
Ilustración 8 15
Ilustración 9 15
Ilustración 10 16
Ilustración 11 16
Ilustración 12 16
Ilustración 13 17
Ilustración 14 20
Ilustración 15 22
Ilustración 16 23
Ilustración 17 24
Ilustración 18 25
Ilustración 19 27
Ilustración 20 27
Ilustración 21 28
Ilustración 22 29
Ilustración 23 29
Ilustración 24 30
Ilustración 25 30

Ilustración 26..... 31

Ilustración 27..... 32

Ilustración 28..... 34

Ilustración 29..... 34

Ilustración 30..... 35

Ilustración 31..... 37

Ilustración 32..... 37

Ilustración 33..... 38

Ilustración 34..... 38

Ilustración 35..... 39

Ilustración 36..... 40

Ilustración 37..... 40

Ilustración 38..... 42

Ilustración 39..... 42

Ilustración 40..... 43

Ilustración 41..... 46

Ilustración 42..... 46

Ilustración 43..... 47

Ilustración 44..... 48

Ilustración 45..... 49

Ilustración 46..... 50

Ilustración 47..... 50

Ilustración 48..... 52

Ilustración 49..... 54

Ilustración 50..... 56

Ilustración 51..... 57

Ilustración 52..... 57

Ilustración 53..... 58

Ilustración 54..... 59

Ilustración 55..... 60

Índice de Tablas

Tabla 1 36

Tabla 2: Costes de personal 68

Tabla 3: Tipos de cotización 2013 68

Tabla 4: Grupo de cotización 69

Tabla 5: Cálculo de cuotas 69

Tabla 6: Coste total por empleado 69

Tabla 7: Costes HW..... 70

Tabla 8: Costes SW 70

Tabla 9: Costes indirectos 70

Tabla 10: Viajes y dietas 71

Tabla 11: Costes totales..... 71

Tabla 12: Beneficio y riesgos..... 71

Tabla 13: Total sin IVA 72

1. Introducción

1.1. Objetivo

El objetivo de este trabajo es hacer un estudio de los CAPTCHAS en la actualidad. Para ver la seguridad que pueden ofrecer en el entorno de Internet y las páginas web.

Estas pruebas son muy utilizadas en Internet, y mucha gente cree que con un CAPTCHA sencillo, tienen cubierta la seguridad, con respecto a la interacción de una máquina, en su página web. Pero, como veremos a lo largo del proyecto, esto no es así, y es que los CAPTCHAS tienen muchos agujeros en su seguridad.

En el trabajo se verán los principales tipos de CAPTCHAS, con ejemplos de cada tipo, y se comentarán algunas herramientas anti-CAPTCHAS existentes, de este modo, se conocerán las principales fortalezas y debilidades de aquellos.

Tras ver la situación actual de los CAPTCHAS, se propondrán nuevas ideas. Estas nuevas propuestas serán seguras con respecto a las herramientas vistas.

1.2. Motivación

La principal motivación de este trabajo, es la creación de nuevas pruebas. Ofrecer una idea innovadora, y un diseño para la posible creación de un CAPTCHA que, a día de hoy, sería más seguro que los vistos y estudiados a lo largo del trabajo.

Para llegar a este fin, se hará un estudio de los CAPTCHAS en la actualidad, y se compararán las nuevas ideas con las ya existentes.

1.3. Contenido de la memoria

En este primer apartado se hará una introducción de los objetivos y la motivación que me llevó a enfrentarme a este proyecto. Además, se hará un breve comentario de lo que trata el trabajo. Se trata de un punto introductorio a lo que veremos más adelante, en el desarrollo del mismo.

En el punto 2, “Presentación de los CAPTCHAS”, veremos más a fondo qué son los CAPTCHAS, una breve historia de estos y algunos ejemplos concretos. También veremos en este punto sus principales aplicaciones, y los problemas que estos causan, sobre todo a personas con problemas de visión. Por último, se comentará la relación entre los CAPTCHAS y las pruebas de Turing y la Inteligencia Artificial.

En el siguiente punto, “Tipos de CAPTCHAS”, se verán los diferentes tipos que nos podemos encontrar, de imagen, audio, video, matemáticos o mini-juegos, y se comentarán ejemplos de cada

uno. También se verá el reCAPTCHA, este es un CAPTCHA de tipo imagen, pero se comentará aparte debido a su fama e importancia en la web.

Tras estudiar lo que son y sus tipos, en el punto 4, "Anti-CAPTCHAS", pasamos a comentar algunas herramientas que son capaces de saltar los CAPTCHAS con relativa facilidad. Veremos herramientas que rompen pruebas de tipo imagen, audio, video y matemáticas.

Por último, tras hacer un estudio de los CAPTCHAS y herramientas Anti-CAPTCHAS, pasaremos a ver nuevas propuestas sobre estos, en el punto 5, "Desarrollo de un CAPTCHA". Se propondrán tres nuevas ideas, se describirán y se verá su funcionamiento. Además, se verá porqué son seguros con respecto a las herramientas vistas.

2. Presentación de los CAPTCHAS

2.1. ¿Qué son?

La sigla CAPTCHAS se corresponde con Completely Automated Public Turing test to tell Computers and Humans Apart, es decir, Prueba de Turing pública y automática para diferenciar máquinas y seres humanos. También es conocido como "acertijo", ya que en muchos casos superar la prueba de un CAPTCHA es similar a pasar un acertijo.

Los CAPTCHAS son una prueba de desafío cognitivo, usados a menudo como herramienta anti-*spam*. El *spam* es el correo electrónico no solicitado, no deseado, o de remitente desconocido, este la mayoría de las veces es de tipo publicitario. El nombre, en realidad, proviene de la denominación de "comida basura", a esta se le conocía como *spam,* sobre todo en América.

Ilustración 2

La función básica del CAPTCHA es diferenciar entre personas y máquinas, para comprobar que se está interaccionando con un ser humano, y proteger a los sitios web del acceso de robots con scripts automatizados. Los CAPTCHAS son pruebas que un ser humano es capaz de pasar, sin embargo, un robot lo tiene más complicado; las pruebas se basan en las capacidades inherentes de la mente humana, difíciles de imitar por una máquina. En teoría, los robots directamente no deberían poder pasar las pruebas, pero como veremos más adelante, hay herramientas que pueden llegar a romperlas y con bastante facilidad.

Diferenciar entre humanos y máquinas es la tarea básica de estas pruebas, pero en realidad, la finalidad de esto es proteger los sistemas vulnerables al *spam* en el correo electrónico o en

páginas web, también son usados para impedir listas automatizadas en algunas votaciones, siendo estas así válidas y legales. Todas sus aplicaciones las veremos más adelante.

En la siguiente imagen vemos uno de los CAPTCHAS más comunes, con el que casi todos nos hemos encontrado alguna o varias veces:

Ilustración 3

Esta prueba consiste en introducir las letras del texto distorsionado. Un ser humano podría resolverlo con facilidad, sin embrago una máquina no debería, o le sería muy complicado, distinguir las letras o números que forman el texto.

Ilustración 4

Los primeros CAPTCHAS consistían en una combinación de letras a las que se aplicaban diferentes técnicas de distorsión, y se hacía una imagen de ello que el usuario tenía que descifrar, como la mostrada anteriormente. Los CAPTCHAS de texto son muy populares ya que sólo con 5 caracteres (letras mayúsculas, minúsculas y dígitos) hay millones de combinaciones posibles.

Esto, supuestamente, era imposible de resolver por una máquina, ya que un ordenador necesita instrucciones muy precisas de lo que debe hacer para resolver un problema, y en este caso no se le pueden dar, ya que hay infinitas formas de representar una letra en una imagen. Sin embargo, hoy en día esto está muy avanzado, hay herramientas que son capaces de leer estas imágenes dividiéndolas en varias partes (segmentación) y descubrir así su contenido.

En un principio se estudió utilizar otras técnicas, como por ejemplo, que estas pruebas consistieran en contestar a preguntas sencillas, preguntas matemáticas del tipo 1+1, o preguntas de sentido común como “de qué color es la hierba”.

A la hora de programar un CAPTCHA, hay que tener en cuenta que para demostrar si es un ser humano o una máquina, este debe poder ser resuelto por un usuario. Hay CAPTCHAS tan complicados que son imposibles de resolver, tanto para unos como para otros, y es que la prueba consiste sólo en demostrar que, efectivamente, se trata de un ser humano, no probar el coeficiente intelectual de este.

Por ejemplo, existen algunos foros dedicados a temas matemáticos que para entrar y verificar que es un ser humano debe resolver CAPTCHAS como los de la imagen de abajo. Esto para usuarios normales podría resultar imposible de resolver, pero para usuarios con conocimientos matemáticos no lo es.

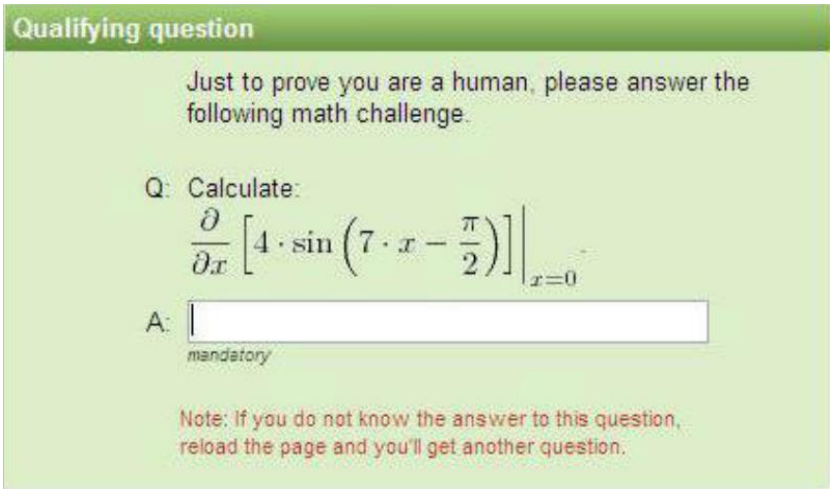

Ilustración 5

Una de las principales ventajas de los CAPTCHAS es que son automatizados, es decir, no necesitan la intervención humana para realizarlos o mantenerlos. Lo cual tiene muchos beneficios en el coste de estos.

A pesar de la importancia de los CAPTCHAS, su creciente uso y su gran investigación, no existe una metodología común y sistemática para su desarrollo y la mayoría de sitios web aún usan

CAPTCHAS ineficientes. Más adelante veremos y estudiaremos herramientas que son capaces de romper algunos o, mejor dicho, la mayoría de ellos.

2.2. La historia de los CAPTCHAS

Ya hemos visto para qué se crearon los CAPTCHAS y su funcionalidad, ahora se comentará un poco su historia.

El primero en proponer este tipo de pruebas para distinguir humanos y máquinas, y así controlar el abuso de determinadas webs de internet, fue Moni Naor en 1996. Sin embargo, los primeros CAPTCHAS fueron desarrollados un año después, en 1997, por Andrei Broder, Abadi Martin, Bharat Krishna, y Lillibridge Marcar, para evitar que robots añadiesen URL's a los motores de búsqueda. Estos CAPTCHAS eran muy primitivos.

El término "CAPTCHA" empezó a utilizarse en el año 2000. Los creadores de este tipo de prueba fueron Luis von Ahn, Manuel Blum, Nicholas J. Hopper y John Langford, de la Universidad Carnegie Mellon, ellos crearon los primeros CAPTCHAS automatizados. Yahoo fue el primero en utilizarlos.

La idea de los CAPTCHAS y el desarrollo de estos, como ya se ha comentado, es debido a que en los últimos años se ha incrementado el abuso de los servicios prestados por Internet. Principalmente el abuso de las cuentas gratuitas de correo, el abuso en sitios web donde se fomenta la publicación anónima y en las votaciones y promociones comerciales. E incluso hay casos de abuso en juegos online y ataques de denegación de servicios.

Los primeros diseños de CAPTCHAS eran básicamente imágenes distorsionadas, un usuario debía descifrarla y poner su contenido. Sin embargo, a pesar de las técnicas de distorsión, las máquinas podían resolver aproximadamente el 90% de este tipo de pruebas. Se intentaron crear algunos más complicados de resolver, para evitar que pudieran ser solucionados por máquinas (con la técnica de segmentación), sin embargo algunos de ellos también eran demasiado complicados para usuarios comunes. Eran complejos tanto por su dificultad de resolución como por su dificultad de visión.

2.2.1. Los peores CAPTCHAS de la historia

A continuación vamos a ver algunas imágenes de los peores CAPTCHAS que se han encontrado los usuarios. Algunos porque sean difíciles de ver, otros muy complicados de resolver, algunos porque son graciosos y otros directamente, indescifrables.

Inentendibles:

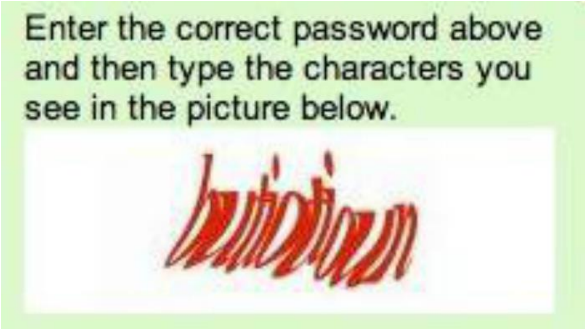

Ilustración 6

Ilustración 7

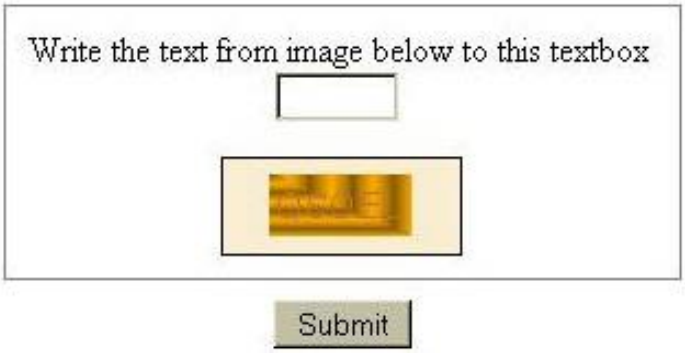

Ilustración 8

Jeroglíficos:

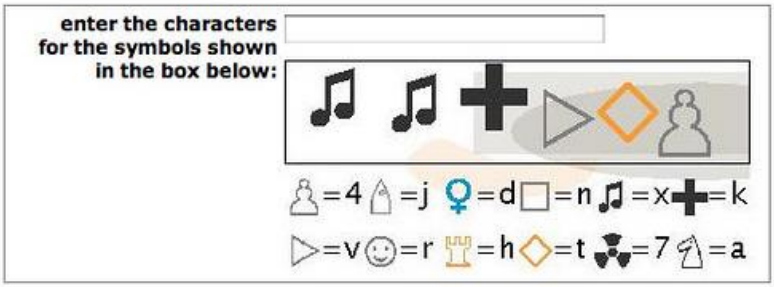

Ilustración 9

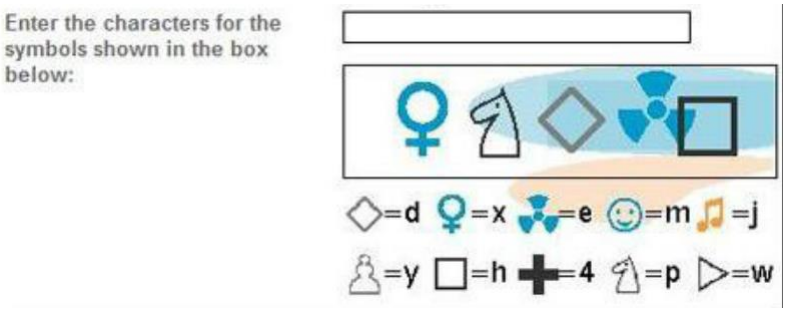

Ilustración 10

CAPTCHAS "hirientes":

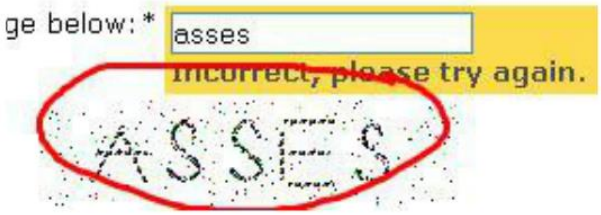

Ilustración 11

Ilustración 12

Por ejemplo los siguientes CAPTCHAS no son complicados de resolver pero hay un problema, la letra "I" (i mayúscula) y la letra "l" (L minúscula), son tan parecidas que a veces es muy difícil distinguirlas:

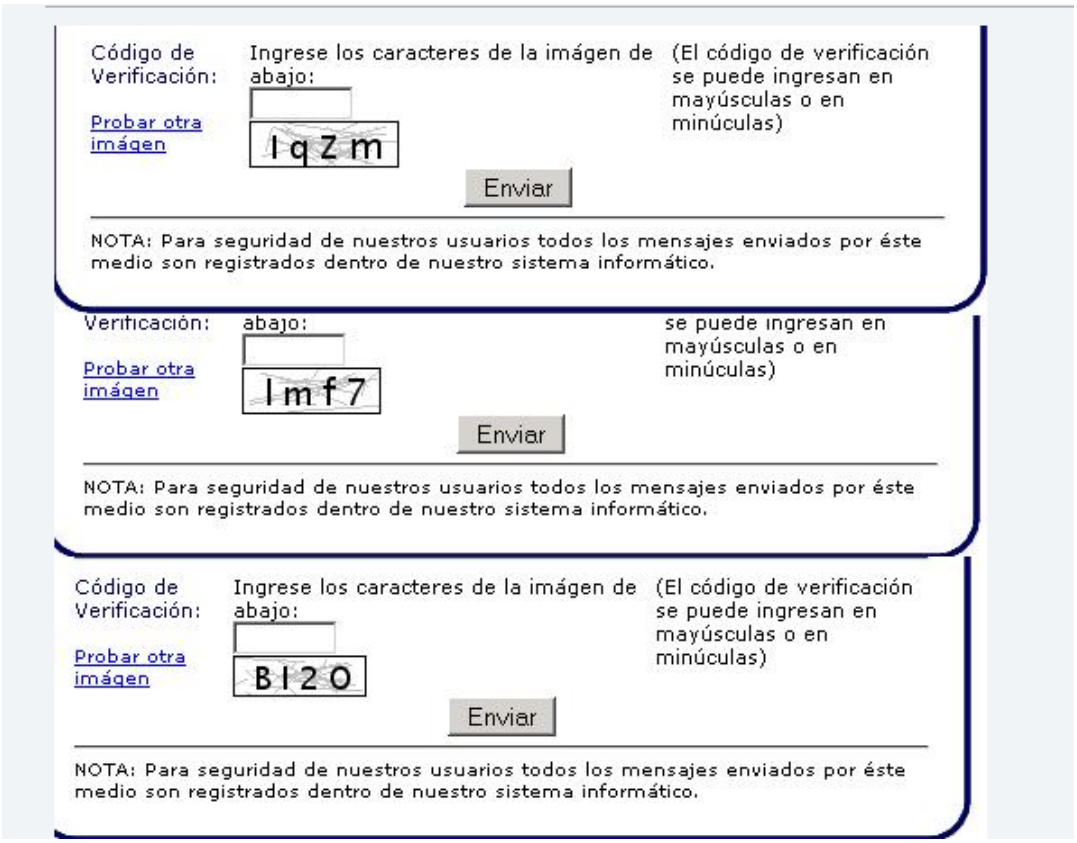

Ilustración 13

2.3. Aplicación

En un principio, para cualquier usuario resolver un CAPTCHA resulta fastidioso o una pérdida de tiempo, pero cuando ves las consecuencias de lo que ocurre cuando no hay uno, que cualquier máquina puede usar un script automático saturando una cuenta o una página, te das cuenta de lo útiles y provechosos que pueden ser.

A continuación vamos a comentar algunas de sus principales aplicaciones:

- **La protección ante el registro masivo de usuarios**. Algunas empresas, como por ejemplo Microsoft, tienen un servicio gratuito de correo. Utilizando un CAPTCHA, estas empresas se aseguran que sólo los usuarios humanos puedan obtener una cuenta de correo, y así no ser víctima del abuso de scripts automatizados que creen miles de cuentas. Hace tiempo sufrieron un ataque de este tipo, desde entonces la mayoría de los sitios gratuitos usan estas pruebas de control.

- **Protección ante *spam* en los comentarios de los blogs**. Muchos *bloggers* utilizan programas que envían comentarios falsos a los blogs. Esto lo hacen para aumentar la preferencia de un sitio web en un motor de búsqueda. Gracias a los CAPTCHAS, esto puede controlarse y sólo los usuarios humanos conseguirían escribir comentarios en los blogs y demás sitios públicos.

- **Protección en las encuestas y votaciones**. Para que una encuesta o una votación sea fiable, se debe tener la certeza de que sólo han votado usuarios reales, y una vez por persona. Esto algunas veces está controlado, utilizando la IP de un ordenador como clave, de cada IP solo se contaría un voto; sin embargo, en otros muchos casos, es el CAPTCHA la principal medida de seguridad en encuestas y votaciones.

- **Protección en los motores de búsqueda**. El uso del CAPCHA en este caso existe, pero es difícil de ver, no es una prueba que se encuentre normalmente. Aquí, el CAPTCHA lo que hace es que cuando un usuario hace clic en una página web, para verificar que es un ser humano, le pide que resuelva un CAPTCHA, así se evita que las máquinas estén continuamente entrando en un sitio web, dándole mayor relevancia de cara a su posicionamiento en los motores de búsqueda.

- **Protección en las citaciones.** También se utilizan los CAPTCHAS cuando reservas una hora (en el médico, en la comisaría…), para evitar que una máquina pueda reservar todas las horas posibles en un mismo día/semana/mes. Esto normalmente se comprueba con el DNI o datos personales de las personas, pero no está de más que también exista un CAPTCHA para controlarlo.

2.4. Problemas que causan los CAPTCHAS

Al igual que todo, los CAPTCHAS tienen su parte buena y sus inconvenientes. A parte de ser molesto para el usuario, un CAPTCHA puede ocasionar problemas más serios, como que a causa de ellos los usuarios no puedan demostrar que realmente son humanos. Esto ocurre normalmente en el caso de los visuales, en los que hay que introducir un conjunto de letras distorsionadas.

Como ejemplos de estos problemas tenemos los siguientes:

- Como ya hemos visto en el apartado anterior de los peores CAPTCHAS, uno de los problemas que pueden causar estos es que un usuario cualquiera no pueda leerlos. Porque las letras estén demasiado juntas o superpuestas, demasiado distorsionadas o que el fondo no acompañe a las letras.

- Otro de los problemas se lo causarían a las personas con dislexia. Para alguien a quien le cueste leer, tener que diferenciar este tipo de letras puede ser demasiado costoso.

- Personas discapacitadas visualmente. Para las personas con problemas de vista sería muy difícil tener que distinguir y averiguar dichas letras distorsionadas.

- Personas daltónicas. A estas personas les costaría en el caso de tener que distinguir las letras de un determinado color.

- Personas de edad avanzada. También podría costarle a muchas personas mayores ver y distinguir dichas letras.

Sin embargo, estos problemas visuales están resueltos gracias a los CAPTCHAS de audio. Existen pruebas sólo de audio y otras visuales, pero que añaden un audio precisamente para solucionar este tipo de problemas.

2.5. CAPTCHAS y Pruebas de Turing

Los CAPTCHAS son, en realidad, pequeñas pruebas de Turing que estamos acostumbrados a ver y pasar para demostrar que somos humanos; de hecho, en las letras que forman el nombre CAP**T**CHA, la T pertenece a "Turing".

La finalidad del test de Turing y el CAPTCHA es la misma, distinguir entre máquinas y seres humanos. La diferencia, es que las pruebas de Turing son analizadas por una persona y los CAPTCHAS por una máquina, por ello, a estos también se les conocen como "Pruebas de Turing inversas".

2.5.1. Pruebas de Turing

El nombre de "prueba de Turing" se corresponde con el apellido de su creador, el matemático Alan Turing. La prueba se expuso en 1950, en el artículo "Computing machinery and intelligence", para la revista Mind. La base de esta propuesta es demostrar la inteligencia en una máquina, para ello se basa en la hipótesis positivista que dice que si algo se comporta como inteligente, es que es inteligente.

Como ya se ha dicho, una de las principales diferencias entre el test de Turing y los CAPTCHAS, es que aquel es analizado y evaluado por un ser humano, al contrario de los CAPTCHAS, que son valorados por una máquina.

"El juego de la imitación", como también se le conoce a esta prueba, es un desafío en el que una máquina se hace pasar por un ser humano, y un tercero debe distinguir a dicha máquina de una persona real.

El test de Turing tiene como factores una máquina, un ser humano y un "juez", este último debe descubrir cuál de los otros dos es la máquina y cuál el ser humano. El procedimiento sería el siguiente, el juez realiza preguntas a ambos, tanto máquina como individuo pueden mentir en sus respuestas. Para que la prueba sea pasada por la máquina, el juez debe confundirse al determinar quién es la persona y quién la máquina, o no poder distinguirlos, es decir, no poder dar una respuesta fiable.

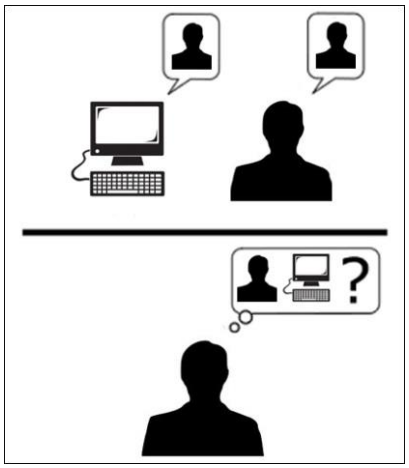

Ilustración 14

La única vez que una máquina ha superado esta prueba fue en el año 2010, cuando el robot Suzette, de Bruce Wilcox, consiguió confundir al juez.

Una de las principales aplicaciones de la prueba de Turing se equivale con otra de los CAPTCHAS, proteger al correo electrónico del indeseado *spam*, comprobando si el remitente es una máquina o un ser humano.

La propuesta de Turing fue la base de la Inteligencia Artificial, y también de la idea de si una máquina podría suplantar a un humano, en cuanto a la capacidad de pensar se refiere.

2.5.2. Inteligencia Artificial

El término Inteligencia Artificial, acuñado en 1956, apunta que la IA es: La ciencia e ingeniería de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes. Se dice que un sistema tiene Inteligencia Artificial, cuando se le incorpora conocimiento o características propias de un ser humano.

Se han hecho muchos estudios sobre este tema en las últimas décadas, porque esto es algo que llama mucho la atención, el cómo podemos dejar el trabajo de muchas personas en manos de una máquina, que ésta tenga la capacidad de pensar, y de este modo hacer dicho trabajo igual que si lo hiciera el propio ser humano.

La IA comprende varios campos, como la robótica o sistemas expertos, que tienen en común la creación y mantenimiento de máquinas capaces de pensar.

A día de hoy, las principales aplicaciones en las que participa son para la creación de videojuegos, juegos de estrategia, traducción de idiomas, robótica…

Superar el test de Turing, no sería una prueba muy significativa de la Inteligencia Artificial de una máquina. Esto sólo demostraría que, en un cierto momento, se podría confundir con un ser humano, sería una inteligencia débil; sin embargo, para hablar de una Inteligencia Artificial fuerte y

real, una máquina debería ser capaz, no sólo de pensar, sino también de actuar y comprender como un ser humano.

Los niveles de la IA serían: poder pensar como un humano, que sería la inteligencia débil que se ha nombrado; poder actuar como un humano; y como último nivel, poder pensar y actuar racionalmente, que sería la inteligencia fuerte.

La Inteligencia Artificial siempre ha tenido muchas críticas, por el hecho de que se puedan crear máquinas que suplanten a las personas, y porque estas sólo serían una copia del original. Además, las máquinas que se puedan crear, sólo tendrían la capacidad de pensar, si así se le puede llamar, de los humanos, pero no llegaría a tener los sentimientos, la compresión, la conciencia, y otras muchas aptitudes y cualidades que pueda tener una persona.

3. Tipos de CAPTCHAS

3.1. Imágenes

Son los CAPTCHAS más típicos y con los que normalmente nos encontramos. El más común consiste en una imagen con letras y números distorsionados, los usuarios deben introducir los caracteres con los que creen que se corresponde la imagen. Fueron los primeros CAPTCHAS, los más sencillos y los que más se utilizan.

Este tipo de pruebas son las que más problemas de visión generan, ya que para cierto tipo de personas leer los caracteres distorsionados puede resultar muy difícil; sobre todo, cuando el fondo tiene muchas líneas, cuando los colores de base y las letras y números son muy similares, o cuando los caracteres están solapados. A menudo, para resolver este tipo de problemas se añade un audio junto con la imagen.

Además de este tipo de CAPTCHAS, existen otros también de imagen, pero que se resuelven de distinta forma que introduciendo las letras correspondientes. Algunos son los que explicamos a continuación.

Los problemas de visión que causarían los caracteres distorsionados, con estos otros CAPTCHAS visuales quedarían resueltos.

3.1.1. ASIRRA

La prueba ASIRRA es un tipo de CAPTCHA en el que aparecen una serie de fotografías de animales (en concreto de gatos y perros), y un usuario, para demostrar que es un ser humano, debe indicar cuál de las imágenes se corresponden con gatos.

El ASIRRA, además de ser más seguro que las pruebas de una sencilla secuencia alfanumérica, evita los problemas de visión causados por las letras distorsionadas.

Ilustración 15

El inicio del desarrollo de ASIRRA fue la propuesta del CAPTCHA KittenAuth. Este consistía en 9 fotografías de animales, de las cuales, el usuario tenía que marcar las 3 que fueran gatos. El problema de esta propuesta, es que su base de datos era demasiado pequeña, <100, por lo que fácilmente se podía "enseñar" a una máquina a que resolviese dicha prueba. Incluso aplicando nuevos métodos de distorsión, este ejemplo es ineficiente.

El ASIRRA se desarrolló en el año 2007, es un tipo de CAPTCHA de imagen, cuya base de datos cuenta con más de 3 millones de fotos de animales, muchas más en comparación con el KittenAuth.

Las imágenes pertenecen al servidor web Petfinder.com, el mayor sitio web encargado de la búsqueda de un hogar a animales abandonados. La prueba consiste en que el usuario debe seleccionar las imágenes que sean gatos de entre las que se le muestren, verá un conjunto de fotos de gatos y perros, en total se le exponen 12 fotos de estos animales. Este tipo de clasificación servirá para indicar si se está tratando con un ser humano. La sigla ASIRRA se corresponde con "Reconocimiento de Imagen de Especies Animales para Restringir el Acceso".

Además, este CAPTCHA tiene otra función, una función de carácter humanitario, y es que junto a la imagen del animal, te da la opción de "Adoptarme", si le interesa acoger al animal de la fotografía. Si seleccionas dicha opción te redirige a la página de Petfinder donde podrás ver el animal y tramitar la adopción.

En el caso de elegir esta opción, la prueba quedaría invalidada, para que esto no afecte a la seguridad del sitio web inicial.

Aquí vemos un ejemplo de este tipo de CAPTCHA:

Ilustración 16

Sin embargo, se han visto muchas debilidades en esta prueba, ya que ha sido vulnerada en numerosas ocasiones con ataques de aprendizaje de las máquinas y ataques de canal lateral.

3.1.2. HumanAuth

Este CAPTCHA también se basa en el funcionamiento del KittenAuth y el ASIRRA, de clasificación de imágenes, pero las fotos son de distinto tipo. Consiste en distinguir imágenes de elementos naturales, tales como el mar, el sol… y artificiales, como un coche, un ordenador, etc. El usuario debe encontrar y marcar los elementos que sean naturales.

La diferencia, y mejora, con respecto al KittenAuth y ASIRRA, es que este tiene más de dos tipos de imagen (perros y gatos), lo cual hace más difícil enseñar a una máquina a que distinga la imagen en cuestión. El internauta debe observar la imagen, que puede tener cualquier elemento en ella, y decidir si es natural o artificial.

Además, este CATPCHA tiene una mejora en cuanto a los problemas visuales, y es que junto a cada imagen tiene una descripción de lo que trata, por ejemplo, “casa roja”, con esa información el usuario debe poder clasificar la imagen.

Por otro lado, el problema de este test, es que la base de datos de sus imágenes es bastante pequeña, tiene un repositorio de 45 imágenes de la naturaleza y 68 de no-naturaleza en formato JPEG. Lo cual hace que el aprendizaje de una máquina pueda romperlo fácilmente.

Otra mejora que propone, y que intenta paliar su problema del pequeño repositorio, es que cuenta con un algoritmo de marcas de agua, que hace más difícil el reconocimiento.

Un ejemplo de este CAPTCHA y el mecanismo de la marca de agua lo vemos en la siguiente ilustración:

Ilustración 17

Como veremos más adelante, aunque sean imágenes más complicadas de adivinar que en el caso de ASIRRA y KittenAuth, y a pesar de la marca de agua, este CAPTCHA también ha sido roto en numerosas ocasiones y no es 100% efectivo. También veremos en los siguientes puntos cómo vulnerarlo.

3.1.3. HotCaptcha

Este tipo de CAPTCHA consiste en un conjunto de imágenes de personas provenientes del sitio web HotOrNot.com, en las que los usuarios deben dar una puntuación a la imagen que están viendo, según le parezca más o menos “caliente”.

Esta página tenía una gran base de datos de imágenes y cualquiera podía enviar su fotografía al sitio web. Sin embargo, esto podía ser bastante ofensivo, además que no tenía fundamento alguno, era la opinión de cada usuario y podía votar quien quisiera.

Este CAPTCHA y el sitio web Hotcaptcha.com no están disponibles desde el año 2009.

Un ejemplo de este lo vemos en la siguiente imagen:

Ilustración 18

3.2. Audio

Normalmente estos CAPTCHAS se encuentran en la misma interfaz que los visuales, y se usan para solventar algunos problemas que estos últimos causan a las personas discapacitadas o disléxicas. Además, también hay CAPTCHAS de audio sin la compañía de los visuales, que se crearon cuando estos empezaron a tener agujeros en su seguridad y era muy fácil su ruptura. Sin embargo, como veremos más adelante, romper los de audio resulta igual de sencillo.

Los CAPTCHAS de audio generalmente son más complicados de resolver que los visuales, ya que se pronuncian una serie de letras o números que el usuario tiene que, primero entender, y después memorizar y escribir, poniendo a prueba su memoria a corto plazo.

Además, otro problema sería que el sonido de estos CAPTCHAS no es nítido, tiene un ruido de fondo, que puede distraer a la máquina que intente romperlos y también dificultar su entendimiento por los humanos. Esta dificultad se ve incrementada en el caso de que el audio sea en una lengua que el usuario no está acostumbrado a hablar o escuchar.

El éxito de estos CAPTCHAS no fue el esperado ya que, por un lado, no es mucho más complicado romperlos y por otro, como ya se ha dicho, son más complejos que los visuales para el usuario. Normalmente con este tipo de pruebas se falla un 50% más que con las de simples imágenes.

3.2.1. Nueva generación de CAPTCHAS de Audio

Existe una nueva generación de CAPTCHAS de audio, que se espera sean más seguros que los habituales test de audio y que el resto de CAPTCHAS, tanto de imágenes, como de video, o cualquier otro tipo.

En este ejemplo se intenta usar lo que fue la idea del ASIRRA y HumanAuth para los CAPTCHAS visuales, es decir, se da una idea, en el primer caso una imagen, en este caso un audio, y el usuario, pensando, trata de llegar a la solución. De este modo, no hay que dar una respuesta sistemática, sino que hay que "elaborar" la solución, lo cual es más complicado para una máquina y habrá menos probabilidades de que pueda romperlo.

Este modelo consiste en que, un usuario escucha un audio, y debe adivinar a qué corresponde el sonido. Por ejemplo, si el audio es un maullido, el individuo debe oírlo y determinar que ese sonido se corresponde con el maullido de un gato. Además de este sonido concreto, al CAPTCHA se le puede añadir ruido de fondo, como por ejemplo, poner el maullido del gato en una calle muy transitada. Esto podría resultar casi imposible para una máquina.

El problema de este ejemplo y por lo que aún no está en vigor, es que es muy difícil dar una solución concreta, por ejemplo en el caso del maullido de un gato, un usuario puede poner "maullido de gato", otro puede poner "gato maullando", otro podría poner "maullar"… y así infinitas soluciones que serían muy complicadas de gestionar, ya que todas ellas serían válidas.

3.3. Video

Estos CAPTCHAS se crearon intentando llegar a una solución frente a la fácil ruptura de las pruebas de imagen y de audio. Al igual que sus predecesores, la idea del CAPTCHA de video es ser considerablemente fácil de resolver para un ser humano, pero imposible para una máquina.

En este tipo de CAPTCHAS, en lugar de ver una imagen en la pantalla (de letras distorsionadas o cualquier otro tipo de prueba de imagen vista) o un audio en concreto, están relacionados con la visualización de un video o una imagen en movimiento.

El principal CAPTCHA de video con el que nos encontramos es el que se explica a continuación.

3.3.1. NuCaptcha

Este es uno de los últimos tipos de CAPTCHAS desarrollados, uno de los más actuales. Es la primera tecnología del test de tipo video. Su principal característica es que es más fácil de resolver por los humanos que uno de imagen común de letras distorsionadas, tiene un éxito de resolución del 97%.

La compañía NuCAPTCHA fue fundada en el año 2008 y cuenta con oficinas en Vancouver y San Francisco, tiene APIs para muchos tipos de lenguajes y plataformas.

Ilustración 19

El CAPTCHA consiste en la inclusión de una pequeña pantalla, donde un usuario ve unas letras en color rojo y detrás un fondo animado, para pasar la prueba y demostrar que es un ser humano, deberá escribir debajo del CAPTCHA el contenido de las letras en rojo. Otra opción, es que el video contenga letras en rojo precedidas de letras en blanco que dicen "Type the code", en castellano algo como "Escriba el código", el procedimiento es el mismo, el usuario deberá escribir sólo el contenido de las letras en rojo.

No podemos reproducir un video de este tipo de CAPTCHA, pero sí dejamos una imagen que lo representa:

Ilustración 20

Este tipo de prueba utiliza la red NuCAPTCHA para monitorear la actividad existente con respecto al CAPTCHA en todo el mundo. La compañía tiene un equipo dedicado exclusivamente a esta prueba, ya que consideran la seguridad como un aspecto muy importante y en continuo cambio, lo que hace que haya que estar siempre al día, o muy fácilmente puede pasar de nuevo a obsoleto. Además, la compañía a la que pertenece el CAPTCHA ofrece al usuario que lo adquiera plataforma de apoyo y teléfono o correo de asistencia 24 horas.

NuCAPTCHA considera que un video puede resultar mucho más fácil de resolver para un usuario que una imagen distorsionada, ya que, al estar en continuo movimiento, se puede ver mejor cada una de las letras que lo componen; los caracteres se van juntando y separando, se ven huecos en blanco en unos y otros lados de la imagen, y así se observarían mejor cada una de las letras que lo forman.

El formato del CAPTCHA es una imagen de video, no es un programa Flash, como muchos creen, ya que resultaría bastante menos seguro. Se trata de un video Stream H.264 MPEG-4.

Otra característica que diferencia al NuCAPTCHA, y que mejora con respecto al resto, es que da más facilidades a los usuarios "fieles", es decir, a internautas que suelen visitar el sitio web y que, podría decirse, ya conoce. Para ello, utiliza un sistema de análisis de comportamiento, que supervisa

todas las interacciones en la plataforma, la información obtenida se usa para dar más o menos complejidad a los CAPTCHAS. A los usuarios que "reconoce" se le proponen pruebas que son más sencillas de resolver y así se les da más facilidad para acceder. A los usuarios que no reconoce, se le proponen pruebas algo más complicadas, para fortalecer la seguridad con respecto a un posible ataque. A pesar de todo esto, los CAPTCHAS siempre son posibles de resolver por un ser humano.

En la siguiente imagen podemos ver el grado de dificultad que tiene según el tipo de usuario que se encuentra:

Ilustración 21

Sin embargo, y al igual que el resto, este CAPTCHA no es infalible, y su principal punto débil es que, al ser un video, y no una fotografía concreta, un atacante puede sacar varias imágenes de un mismo caso, y así, con varias imágenes de las mismas letras, tendrá más posibilidades de averiguar cual es la secuecia correcta. Esto lo veremos más adelante con más detalle.

3.4. Matemáticos

Otro CAPTCHA muy conocido sería el "MAPTCHA", este es de tipo matemático. La prueba consiste en que un usuario resuelva una fórmula matemática, normalmente un límite, e introduzca el número correcto en la solución.

Esta prueba no es muy eficiente, debido a que, para una máquina sería fácil memorizar muchas fórmulas matemáticas, haciendo un aprendizaje del CAPTCHA, y así podría resolverlas cuando se le presenten. Sin embargo, hay veces que las fórmulas que se proponen son bastante complicadas para un usuario.

Este es el caso que se ha comentado en el primer punto de, por ejemplo, foros matemáticos que utilizan la prueba de límites matemáticos para demostrar que es un usuario y no una máquina. En la siguiente imagen podemos ver otro ejemplo de esto:

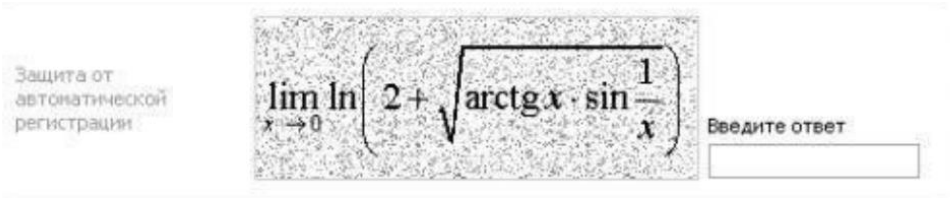

Ilustración 22

Normalmente este CAPTCHA no es creado por personas expertas en seguridad, sino por matemáticos, o personas que llevan los sitios web que pretenden proteger. Lo cual lleva a que su creación tenga la intención de ser más interesante de resolver que realmente seguro para el sitio web.

Algunos casos de MAPTCHA no son fáciles como para que los resuelva un usuario cualquiera, sino con ciertos conocimientos matemáticos, sin embargo, en muchos sitios donde se encuentra este CAPTCHA, si un usuario no puede resolver la prueba que se le ofrece, puede recargar la página y encontrarse con una prueba más sencilla, ya que son varias opciones de prueba y de mayor y menor dificultad las que este CAPTCHA ofrece. Aquí vemos otros ejemplos con los que un usuario se puede encontrar:

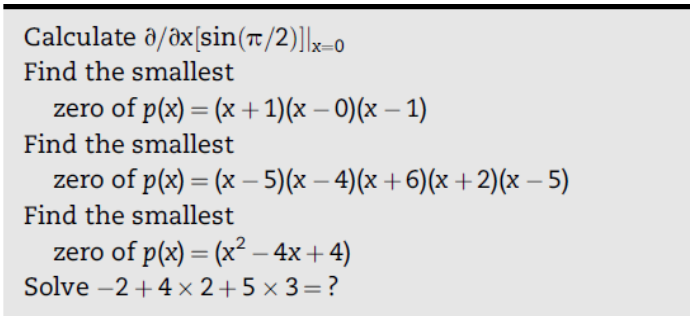

Ilustración 23

El MAPTCHA es un servicio gratuito y, al igual que todos los comentados anteriormente, es aleatorio y automatizado, los números que forman la ecuación matemática son fortuitos.

Cuando salió a la luz, destacó por la novedad y la idea revolucionaria de este CAPTCHA, pero ya se ha comprobado que no es realmente seguro, ya que es fácilmente vulnerado por las herramientas anti-CAPTCHAS.

3.5. Mini-juegos

Además de los típicos CAPTCHAS que al usuario le molesta tanto tener que descifrar, se han creado nuevas alternativas, digamos más "entretenidas". Son los CAPTCHAS PlayThru, esta es una iniciativa de "Are You A Human".

Este tipo es una especie de mini-juego; en lugar de tener que insertar una palabra, en este caso el usuario tendría que "jugar" con el sistema. Las pruebas que se proponen son del tipo:

arrastrar un coche a una plaza libre de aparcamiento, añadir ingredientes a una pizza, cazar mariposas u otros juegos similares.

Además, existe un sitio web, puzzlecaptcha.com, que comercializa con CAPTCHAS de tipo juegos. A pesar del nombre, los CAPTCHAS que contiene este sitio web no son puzles como tal, es decir, de colocar piezas, sino que utiliza una especie de juegos, en los cuales el usuario tiene que hacer clic en alguna parte determinada de la imagen para pasar la prueba.

El sitio web da una prueba gratuita para los usuarios que les interese adquirirlos, y también desde la propia página se pueden probar los juegos que hay. Además, se proporcionan los colores y diseños del CAPTCHA a elegir, según quiera el usuario para su página web o donde necesite utilizarlo.

Aquí tenemos un par de ejemplos de los juegos que vemos en esta página:

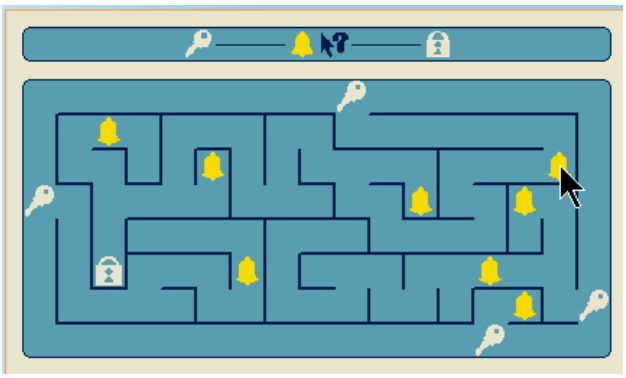

Ilustración 24

Ilustración 25

3.5.1. Puzle-CAPTCHA

Esta es otra variante de los CAPTCHAS de imágenes y en concreto del tipo "juegos", al contrario que en el caso de la página web puzzlecaptcha.com, este tipo de prueba sí radica en resolver un puzle común.

Según su creador, esta variante es la solución a todos los problemas que causan los CAPTCHAS visuales y los de audio. Es fácil de resolver por un usuario, sin embargo, más complejo para una máquina.

Es soportado por cualquier sistema y desarrollado por las principales herramientas. Los puzles que se generan en cada ocasión son aleatorios y automatizados.

El CAPTCHA consiste en un rompecabezas o puzle, el internauta debe visualizar la imagen y descubrir qué pieza va en cada sitio vacío. El usuario arrastra el trozo que cree que va en cada hueco para colocarlo. El rompecabezas utiliza el mecanismo Drag and Drop para colocar las piezas.

Para verificar la solución dada al rompecabezas, se utiliza AJAX, por lo que esto puede ser integrado en la rutina de validación de formularios JavaScript.

Aquí vemos un ejemplo:

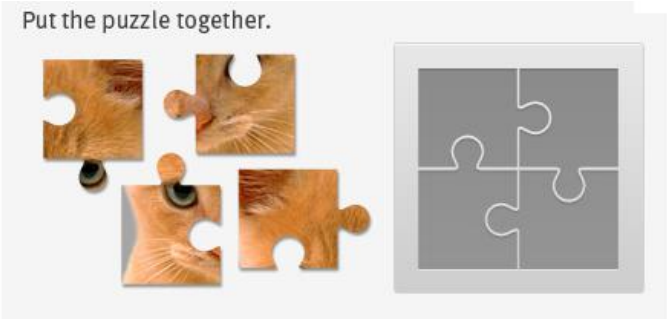

Ilustración 26

3.6. ReCAPTCHA

Se trata de un tipo de CAPTCHA visual y de audio, pero lo comentamos aparte ya que es bastante importante y merece la pena tratar en profundidad.

Los primeros CAPTCHAS que se crearon, tienen como única función la que ya se ha contado, asegurarse de que se está interaccionando con un ser humano y no con una máquina, y de esta forma proteger los sitios web. Pero más adelante, los CAPTCHAS podría decirse que evolucionaron, y se les propuso una nueva tarea, a estas nuevas pruebas se les llamó reCAPTCHA.

Cuando un usuario resuelve un CAPTCHA, esto le lleva unos 10-15 segundos de su tiempo, bien, pues a su creador principal, Luis von Ahn, esto le hizo pensar en cómo utilizar ese tiempo que cada usuario pierde cuando se encuentra con uno delante, y se le ocurrió que cuando una persona resolviese un CAPTCHA, en realidad estuviese resolviendo un algoritmo. Quiso utilizar, como él mismo dice, la capacidad de pensar, capacidad de la mente humana, que está por encima de la de cualquier máquina, para algo que fuese beneficioso para la humanidad.

Según sus estimaciones, los seres humanos de todo el mundo escriben más de 100 millones de CAPTCHAS cada día, la idea de este experimento era, intentar hacer un uso positivo del tiempo empleado para resolverlos.

El algoritmo de los CAPTCHAS en realidad, es que las palabras que lo forman provienen de libros escritos que se quieren digitalizar, de este modo, cuando resolvemos uno de ellos, en realidad estamos descubriendo una palabra de algún libro que se está digitalizando.

El libro escrito se escanea, una máquina descubre las palabras que puede, y el resto, que la máquina en un primer momento no ha podido descifrar, se le pasa a los usuarios en forma de CAPTCHA. Esto ocurre sobre todo en el caso de libros antiguos, con tinta borrosa y páginas amarillentas. La herramienta no puede reconocer aproximadamente el 20-30% de las palabras. Por el contrario, los seres humanos son más exactos en la transcripción de la impresión.

Se ha demostrado que, gracias a este método, se han transcrito libros con una precisión de palabras superior a 99%, casi igualando la garantía de profesionales transcriptores humanos. El trabajo de estas personas es muy caro, por lo que su uso se limitaba a transcripciones de suma importancia.

Esta nueva idea de CAPTCHAS ha llevado a que ahora se tengan que utilizar dos palabras en una misma imagen, es decir, el CAPTCHA está formado por dos vocablos, para poder comprobar las respuestas que se obtienen. Una de las palabras es la que debemos descubrir, desconocida y proveniente directamente de un libro de texto, y la otra, es una ya conocida por el sistema. Si el usuario escribe correctamente la palabra que el sistema sí conoce, asume que es un ser humano y el CAPTCHA sería válido. Más adelante, cuando se obtenga el resultado de más internautas para esa misma pareja de palabras, si todos o una gran mayoría de ellos dan la misma solución para la palabra desconocida, se tomará por válida y será de confianza para la digitalización del libro. Por supuesto que el usuario no sabe cuál es el vocablo conocido y cuál el desconocido del sistema, él debe resolver ambos por igual.

Un ejemplo de este CAPTCHA es el de la siguiente imagen:

Ilustración 27

Este nuevo CAPTCHA está desplegado en más de 40.000 sitios web y ha transcrito más de 440 millones de palabras, aunque este dato es de hace tiempo, probablemente ya se hayan transcrito muchas más.

Se ha comprobado que gracias a este sistema se han digitalizado libros mucho más rápido y con mucha más facilidad.

4. Anti- CAPTCHAS

4.1. Introducción

Como ya sabemos, los CAPTCHAS no son 100% seguros, la idea de crear una prueba para demostrar que un usuario es una persona y no una máquina, es buena, siempre y cuando, de verdad la persona esté resolviendo dicha prueba. Sin embargo, se han creado herramientas que son capaces de superar estas pruebas sin apenas problemas, lo cual es un grave inconveniente en la seguridad del CAPTCHA y, por tanto, del sitio web donde se encuentre.

Para romper un CAPTCHA, la idea es encontrar un algoritmo automatizado capaz de superar la prueba, de manera similar a como lo haría un ser humano. De esta forma, el CAPTCHA le confundiría con un usuario y le tomaría como tal, por lo que le dejaría entrar al sitio web y poner su opinión en un foro, realizar una encuesta o cualquier otra tarea que esté restringida a los humanos por el propio CAPTCHA.

La mayoría tienen muchas lagunas en su seguridad, y resulta irónico que no se preste demasiada atención a esto, sabiendo lo útiles y eficaces que pueden ser si son seguros y realmente cumplen con su tarea.

En los siguientes sub-apartados vamos a ver diferentes casos de herramientas para la ruptura de estos, siendo el más común la segmentación, que rompe los CAPTCHAS visuales que utilizan letras y números distorsionados, descubriendo uno a uno cada carácter que contiene. Pero además, veremos otros casos, tanto de tipo imágenes, como de audio, video y matemáticos también.

Las herramientas que se van a comentar tienen su descripción en Internet, aunque la mayoría de ellas no podemos encontrarlas tan fácilmente, ya que su uso no es legal. Los creadores de estas herramientas anti-CAPTCHAS han comentado los resultados obtenidos con los propios desarrolladores de los CAPTCHAS originales, con el fin de ayudar a estos en el desarrollo de nuevas pruebas, más seguras que las que encontramos actualmente.

Antes de la realización del proyecto, se han estudiado los ejemplos de las herramientas que se comentan a continuación, con el fin de diseñar un buen CAPTCHA, innovador, e inmune a estas. Sin embargo, en este trabajo no se va a realizar el estudio de dichas herramientas, simplemente se comentará su función y qué tipo de pruebas son capaces de romper.

4.2. Segmentación

El principal ejemplo de ruptura de un CAPTCHA corriente, es el de descifrar los caracteres que contiene la imagen, escribirlos, y de este modo, superar la prueba. La gran mayoría de los CAPTCHAS que nos encontramos son visuales y con caracteres distorsionados, por lo que, este tipo de herramienta es capaz de romper la prueba más común y utilizada.

Una de las partes más importantes para la ruptura de un CAPTCHA de este tipo, se basa en la segmentación, aunque esto no sea todo, pero es muy importante. Se aprovecha que la longitud de

código de la imagen sea fija, para hacer una conjetura de dónde y cómo segmentar este código, además, si se conoce el número de caracteres fijo que contiene, la segmentación simplemente necesita buscar dicho número e ir separándolos uno a uno.

El código de la imagen automatizada se divide en cinco pasos genéricos: pre-procesamiento, segmentación, post-segmentación, reconocimiento y post-procesamiento.

El pre-procesamiento consistiría en eliminar patrones de fondo u otras adiciones a la imagen, lo conocido como “ruido”, que pueda interferir o causar problemas en la segmentación. La segmentación, y los pasos intermedios, estarían encargados de realizar lo que es el proceso de división en sí.

El proceso de segmentación básicamente consiste en dividir los caracteres que se encuentran en el CAPTCHA. En un principio, se realiza el pre-procesamiento, así que para el proceso de segmentación ya se tendría el CAPTCHA limpio, sin "ruido" externo. Una vez tengamos la imagen dividida por caracteres, se pasaría al estudio y reconocimiento de cada uno.

Las siguientes imágenes describen mejor cómo sería el proceso de segmentación de un CAPTCHA.

Segmentación sin pre-procesamiento:

Ilustración 28

Segmentación con pre-procesamiento:

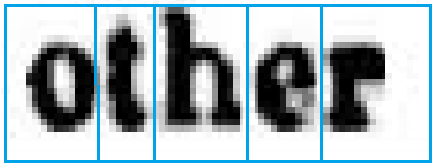

Ilustración 29

Tras el reconocimiento de caracteres, el post-procesamiento mejora la precisión del resultado, por ejemplo, aplicando reglas ortográficas si el resultado de una imagen consiste en palabras.

4.3. Herramienta ENT

La herramienta ENT es utilizada para romper algunos CAPTCHAS de tipo imagen, como el ASIRRA o el HumanAuth.

Esta herramienta es un algoritmo que hace una serie de pruebas para buscar la aleatoriedad de una secuencia de bytes y, de este modo, hacer una clasificación de imágenes. Las pruebas que contiene esta herramienta serían las siguientes:

- Tamaño: el tamaño propio de la imagen.
- Entropía: esto es "la cantidad de información promedio que contienen unos símbolos", aplicándolo a este caso concreto, la entropía de una imagen es la medida de su contenido de información. Si la entropía de una imagen es alta, quiere decir que la imagen es imprevisible, aleatoria, sin embargo, si su entropía es baja, la imagen es más predecible. Cuando hallas la entropía de una imagen, calculas la probabilidad de su ocurrencia, calculas un valor que representaría el número de bits necesarios para representar la probabilidad.
- Chi-Cuadrado: calcula el p-valor estadístico de cada atributo con respecto a la clase, y así, obtiene el nivel de correlación entre la clase y cada atributo.
- Compresión: esta prueba nos da un porcentaje del tamaño que se podría obtener, si el archivo es comprimido usando un algoritmo de compresión sin pérdida de datos, como por ejemplo, el algoritmo Lempel-Ziv.
- Media aritmética: calcula el valor medio de bytes del fichero de entrada.
- Algoritmo de Monte-Carlo: es una técnica numérica para calcular probabilidades y otras cantidades relacionadas, utilizando secuencias de números aleatorios. En este caso, se utiliza este algoritmo para calcular el valor de Pi, utilizando el archivo de entrada como fuente de aleatoriedad.

Ilustración 30

- Correlación serial (autocorrelación): mide cómo un byte puede ser dependiente del byte anterior, los bytes están vinculados entre sí.

La herramienta ENT es capaz de tratar y trabajar con formatos ASCII, BMP, JPGE o MP3, sin embargo, en este caso simplemente nos interesan los archivos BMP y JPGE, ya que vamos a estudiar esta herramienta para vulnerar el ASIRRA y el HumanAuth, que son de tipo imagen.

En la siguiente tabla 1 (27) podemos ver algunos resultados que obtendríamos de la herramienta ENT, cuando le introducimos estos dos formatos de imagen:

Atributo	BMP	JPGE
Tamaño	1683594	4914423
Entropía	7.24	7.91
p-valor del chi-cuadrado	>0.01	>0.01
Compresión	9%	1%
Media	73.21	137.88
Monte-Carlo (PI)	3.45	2.84
Autocorrelación	0.537042	0.004862

Tabla 1

4.3.1. Ataque a ASIRRA

Los creadores de ASIRRA han puesto su CAPTCHA a prueba de la herramienta ENT, con el fin de estudiar y probar su seguridad. Para esta prueba se usaron 25.000 imágenes, de gatos y perros, mitad de cada tipo.

Las imágenes originales se trataron con la herramienta ENT, esta proporciona un fichero de salida con formato arff. Este fichero se abre con la herramienta Weka, ahora veremos los resultados que ofrece el estudio.

Tras abrir el fichero obtenido por ENT, se comprueba que, únicamente usando esta herramienta, y sin que tuvieran que usar ninguna otra técnica de reconocimiento de imágenes, ya se puede hacer una clasificación de gatos y perros con un acierto de casi el 60%.

Haciendo ensayos sobre esto, se comprueba que, usando un árbol simple de decisión, que se basa únicamente en el tamaño del archivo, puede distinguir una imagen de gato o perro con una precisión del 57%. Esto ya es bastante mejor al resultado del 50% que obtendríamos del azar, hay dos opciones, por lo que, ½ de probabilidad de acierto. Y recordemos que, únicamente se basa en el atributo “tamaño” de la fotografía.

Usando un clasificador más complejo, se ha llegado a obtener un 58,0326% de acierto en la clasificación. Tenemos algunas imágenes que nos proporciona el estudio que se hizo sobre ASIRRA. En la siguiente ilustración 30 (27) vemos un acierto de casi el 60% usando un clasificador complejo:

```
=== Run information ===
Scheme:       weka.classifiers.meta.LogitBoost
              -P 100 -F 0 -R 1 -L -1.7976931348623157E308 -H 1.0 -S 1 -I 10
              -W weka.classifiers.trees.DecisionStump
Relation:     catsdogs
Instances:    24998
Attributes:   9
              entropy
              size
              compressionrate
              chisqstatistic
              arithmean
              montepi
              errmontepi
              corr
              class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

LogitBoost: Base classifiers and their weights

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       14507               58.0326 %
Incorrectly Classified Instances     10491               41.9674 %
Total Number of Instances            24998

=== Confusion Matrix ===

    a    b   <-- classified as
 8288 4211 |    a = Cat
 6280 6219 |    b = Dog
```

Ilustración 31

Como vemos en la imagen, del total de imágenes, 24.998 ya que hubo dos imágenes corruptas y, por tanto, descartadas de las 25.000; acierta en 14.507 y falla en 10.491, lo que supone un 58% de acierto.

Al ver estos resultados, parece claro que el atributo más significativo de ASIRRA es el tamaño de las imágenes, sin embargo, a través de Weka, también se puede comprobar qué atributo es más significativo y en qué medida.

```
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

average merit      average rank  attribute
729.115 +-23.245     1   +- 0       2 size
223.755 +-19.423     2   +- 0       4 chisqstatistic
171.656 +-17.037     3   +- 0       5 arithmean
144.709 +-10.347     4   +- 0       1 entropy
81.333 +- 6.923     5.1 +- 0.3     8 corr
73.121 +- 4.512      6   +- 0.45    6 montepi
66.109 +-10.686      6.9 +- 0.3     7 errmontepi
16.297 +- 8.621      8   +- 0       3 compressionrate
```

Ilustración 32

Como se ve, efectivamente, el tamaño es el atributo más significativo, seguido del Chi-cuadrado y la media aritmética. La comprensión sería el atributo menos significativo, y además con poco valor, lo cual quiere decir, que este atributo va a afectar muy poco a la clasificación de las imágenes.

Como nota a los creadores del ASIRRA, y para futuras mejoras del CAPTCHA, deberían controlar más el tamaño de las imágenes, hacer que sea el mismo o muy similar para ambos tipos de

fotografías. Aún así, quitando este atributo, la herramienta ENT es capaz de clasificar las imágenes con un acierto del 56,8%, lo que ya es bastante mejor que el azar.

4.3.2. Ataque a HumanAuth

Ahora veremos el estudio de un caso similar, el HumanAuth, que también puso su trabajo en manos de la herramienta ENT, para su estudio y valoración de la seguridad.

Para esta prueba cogieron la base de datos de HumanAuth, y la herramienta ENT dio un fichero de salida de formato arff, que estudiaron con Weka, igual que para el caso de ASIRRA. Como veremos a continuación, los resultados de la prueba dieron un alto porcentaje de acierto por parte de la herramienta ENT, superiores incluso a los del CAPTCHA anterior.

Utilizando el mejor clasificador para este caso, el randomForest, obtenemos un muy buen resultado, hasta casi un 78% de acierto por parte de la herramienta (31):

```
=== Run information ===
Scheme:         weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1
[.....]
Test mode:     10-fold cross-validation
Random forest of 10 trees, constructed considering 4 random features.
=== Summary ===
Correctly Classified Instances          88               77.8761 %
Incorrectly Classified Instances        25               22.1239 %

=== Confusion Matrix ===
  a  b   <-- classified as
 34 11 |  a = nature
 14 54 |  b = nonnature
```

Ilustración 33

Esta información la obtenemos usando todos los atributos. En la siguiente imagen podemos ver una clasificación de estos:

```
=== Run information ===
Evaluator:     weka.attributeSelection.ChiSquaredAttributeEval
[.....]
Evaluation mode:     10-fold cross-validation
=== Attribute selection 10 fold cross-validation (stratified),
seed: 1 ===
average merit          average rank  attribute
38.987 +- 5.975         1.9 +- 1.58     8 corr
34.58  +- 3.872         2.6 +- 1.2      3 compressionrate
32.936 +- 6.025         2.7 +- 0.78     2 size
29.844 +- 4.878         3.9 +- 0.83     1 entropy
29.885 +- 4.57          4.6 +- 1.85     4 chisqstatistic
27.228 +- 1.393         5.3 +- 0.46     5 arithmean
 0      +- 0            7.2 +- 0.4      7 errmontepi
 0      +- 0            7.8 +- 0.4      6 montepi
```

Ilustración 34

Los atributos más significativos serían: correlación, comprensión y tamaño. Los creadores de la herramienta también hicieron pruebas quitando estos atributos, y, aun quitando los tres atributos más significativos, se obtiene un acierto de aproximadamente el 75%.

4.4. Ataque a los CAPTCHAS de Hotmail y Gmail

Hotmail y Gmail son sitios web donde millones de internautas crean su cuenta personal de correo. Son sitios muy conocidos y muy utilizados por los usuarios, por ello su sistema de correo debería ser fiable, para ello habría que darle mucha importancia a la seguridad en su sitio web. Sin embargo, esto no es así; estos sitios incluyen CAPTCHAS para intentar protegerse de los ataques bots, que crean millones de cuentas en cuestión de minutos, pero sus pruebas no son suficientemente seguras. Utilizan CAPTCHAS muy sencillos y que, como veremos ahora, son vulnerados fácilmente.

En cuanto a Hotmail, se ha comprobado que ya existe una nueva herramienta que puede atacarlo en aproximadamente 6 segundos, lo cual es un peligro, ya que se pueden crear muchísimas cuentas en cuestión de escasos minutos.

Al ir a crearnos una cuenta nueva en Hotmail, vemos que este es el CAPTCHA que nos aparece para resolver, tras completar el formulario de creación de cuenta nueva:

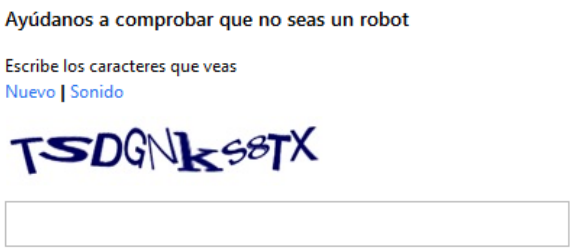

Ilustración 35

Nos aparece un gran número de caracteres, sin embargo, las “trabas” que se ponen para averiguar estos caracteres, son más bien escasas. Algunas técnicas como la distorsión, el solapamiento, el fondo o líneas que crucen el CAPTCHA, ni siquiera existen. No son técnicas que garanticen un 100% su seguridad, pero en este caso, no existen ni estas, ni otras técnicas de protección, lo cual deja muy al descubierto el CAPTCHA, y muy fácil de romper para cualquier herramienta sencilla.

En cuanto al CAPTCHA de Gmail, este ha sido modificado hace poco tiempo, ya que la seguridad del antiguo, brillaba por su ausencia.

Este fue vulnerado multitud de veces veces, tanto la imagen como el audio. En la siguiente imagen podemos ver un ejemplo de él:

Ilustración 36

Ahora, cuando vamos a crearnos una cuenta nueva en Gmail, este es la prueba que nos aparece:

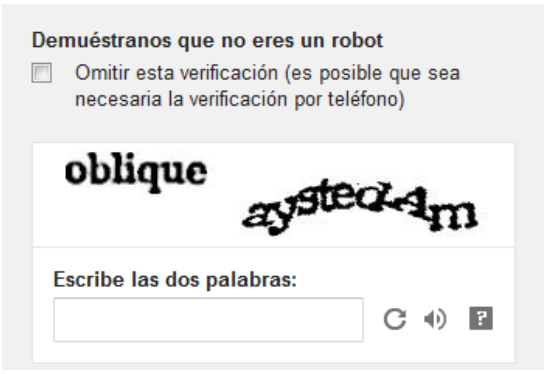

Ilustración 37

Por un lado, está la opción de demostrar que es un ser humano dejando el número de teléfono, y a través de un mensaje de texto o una llamada, confirman que es un usuario. Sin embargo, la opción de resolver el CAPTCHA, es tan válida como esta, así que, si se trata de una máquina quien quiere vulnerarlo, tomaría la opción de resolver este.

Pasando directamente a la resolución del CAPTCHA, (en este caso se trata de un reCAPTCHA) se escribirían las letras distorsionadas que aparecen en la imagen, además, existe la opción del audio, si la persona tuviera problemas de visión.

4.5. DeCAPTCHA

DeCAPTCHA se utiliza para la ruptura de múltiples CAPTCHAS, por ejemplo, para vulnerar el antiguo CAPTCHA de Gmail, que se acaba de comentar, sin embargo, en este punto vamos a tratar la ruptura de los de tipo audio en concreto.

La base del funcionamiento de esta herramienta, es la misma que la de la herramienta ENT, utilizada para romper los visuales.

Consta de varios pasos:

- Primero se "limpia" el CAPTCHA. En este caso, al ser de audio, lo que se hace para limpiarlo, es quitar el ruido de fondo, para quedarse con la parte que hay que descifrar. Para separar el audio, se analiza la energía detectada en cada momento de onda, y nos quedamos con las partes de mayor energía.

- Después se descifra el mensaje y se escriben los caracteres que se han extraído de la grabación.

- El último paso, es pasar los dígitos extraídos por una fase de clasificación, y de este modo, darle algún sentido a la secuencia.

Para llegar a romper un CAPTCHA de este tipo, es necesario el entrenamiento de la máquina, para ello, los creadores de la herramienta, utilizaron 300 ejemplos ya resueltos.

Con el fin de probar la efectividad de esta herramienta, se hicieron numerosos experimentos. En el caso de la página Authorize.net, tuvo un éxito de aproximadamente el 90%, con el CAPTCHA de Ebay, tuvo un éxito del 80% y con el de Microsoft, del 50%. Se llegaron a tener mejores resultados, incluso que si fuera un ser humano el que resolviese estas pruebas.

4.6. Ataque al CAPTCHA de video

En este punto, vamos a ver la facilidad con la que una herramienta es capaz de vulnerar una prueba de tipo video. Para este caso, nos vamos a centrar en el CAPTCHA de video más conocido y desarrollado, y que ya se ha visto en este mismo trabajo, el NuCAPTCHA.

Al igual que para los casos anteriores, no contamos con la herramienta en cuestión que desarrollaron para romperlo, pero vamos a ver en qué consiste. El éxito de esta, para el NuCAPTCHA en concreto, es de más del 90% de acierto, lo que demuestra la escasa seguridad de este.

El algoritmo de la herramienta consiste en lo siguiente:

1. Primero se toman diferentes imágenes del video.

2. Pre-procesamiento: Se hace una limpieza de la imagen y se pasa a blanco y negro, para que sea más fácil de tratar.

3. Análisis: en este punto se extraería el CAPTCHA en sí, del resto de caracteres que pueda contener la imagen. Vamos a ver más detenidamente este apartado ya que es el más importante de la herramienta. Este a su vez, se divide en dos puntos:

    - Análisis de la imagen: se busca "el objeto más interesante", es decir, se busca la secuencia de caracteres que la herramienta cree que es el CAPTCHA. Para ello, estudia las imágenes que ha capturado, buscando qué parte le puede interesar. Esto lo hace comprobando el tamaño (lo forman 4 letras), la relación de altura/anchura entre las letras, y estudiando los bordes y esquinas de las letras (ya que estas giran durante el

video). La herramienta va eliminando opciones que superen unos ciertos umbrales hasta que decide cuál cree que es la parte que necesita. Aquí vemos una imagen de lo que esto representaría:

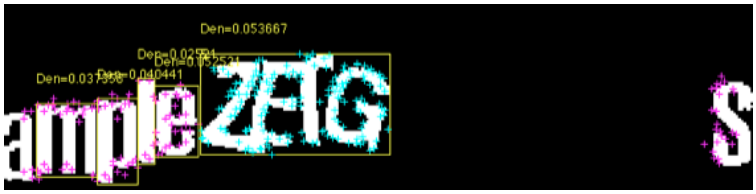

Ilustración 38

- Análisis cross-frame: una vez se tiene, o se cree tener el CAPTCHA en cuestión, se hace un estudio más a fondo, con 50 fotografías de la misma parte, teniendo en cuenta los cuadros que ha marcado en un primer momento como posible CAPTCHA. Vemos en la siguiente imagen cómo trabaja la herramienta, siendo el punto rojo más alto, el indicativo del cuadrado que considera como solución:

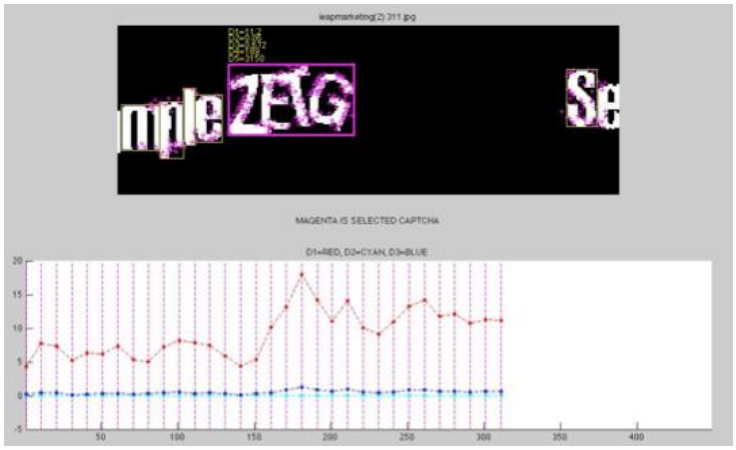

Ilustración 39

4. Segmentación. Tras este proceso, tendremos separado cada carácter del cuadrado obtenido como solución. Este punto es mucho más fácil que cuando se trata de un CAPTCHA común, ya que tenemos 50 copias diferentes para comparar y obtener letra a letra.

5. Reconocimiento: en esta última fase se reconoce cada letra individual utilizando un algoritmo de aprendizaje automático.

   Los desarrolladores ofrecen la siguiente imagen representativa del algoritmo:

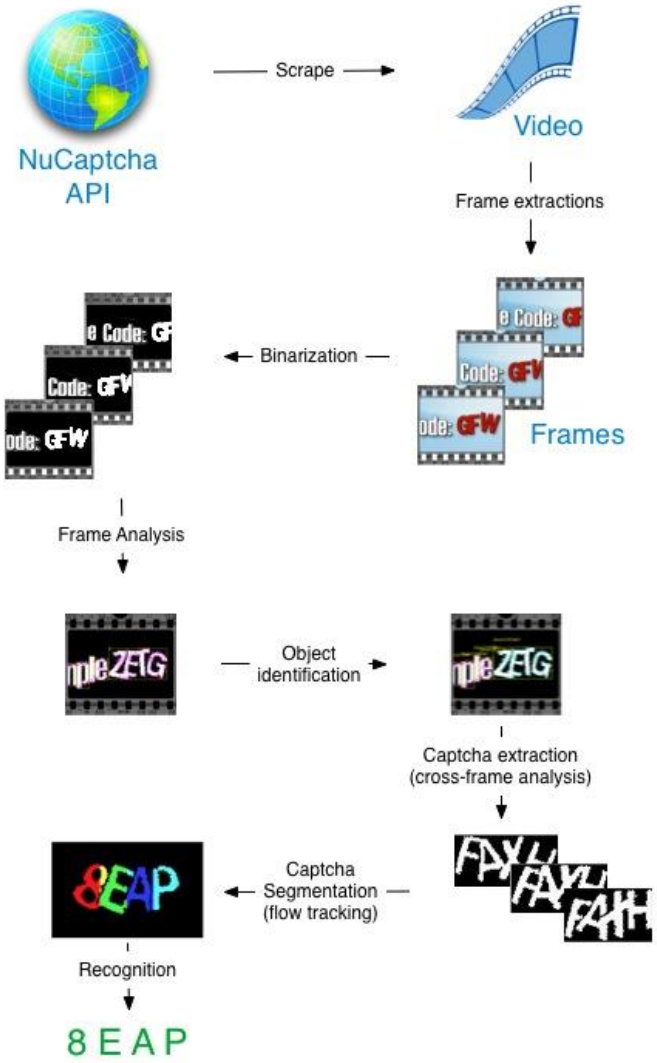

Ilustración 40

4.7. Ataque a MAPTCHA

Como ya se ha explicado en el punto 3, de los tipos de CAPTCHAS, el MAPTCHA consiste en una prueba matemática. Ofrece al usuario una cuestión matemática a resolver, para demostrar que es un ser humano. Sin embargo, este ejemplo no es un CAPTCHA muy seguro, y es saltado con relativa facilidad. A continuación veremos algunos de sus defectos que lo hacen inseguro ante un ataque.

La principal resistencia de los CAPTCHAS que nos encontramos habitualmente, es la de los caracteres distorsionados y unidos o superpuestos, que se intenta romper utilizando la segmentación; en este caso, esto no es necesario, ya que el MAPTCHA pone sus números y símbolos de forma natural, tal y como nos los encontraríamos en un teclado por ejemplo, esto da mucha facilidad a la hora de leerlos y descifrarlos por parte de una herramienta.

Sin embargo, la dificultad en cuanto a los caracteres que forman el CAPTCHA, es su baja resolución, que a veces le resulta complicado de ver, tanto a usuarios, como a máquinas, y esto puede hacerlo algo más seguro con respecto a las herramientas.

Para descifrar los caracteres del MAPTCHA, lo que hace es ir buscando uno a uno todos los símbolos y números que estén formando la fórmula. Para esto, utiliza algoritmos de reconocimiento de caracteres, a través de los píxeles de la imagen que contiene la fórmula.

Una vez reconocido cada símbolo, se colocan en el orden que formula la ecuación original, y pasaría a resolverse.

Una de las principales faltas, en cuanto a la seguridad de este CAPTCHA, es la posibilidad de recargar la página las veces que sean necesarias, y que de este modo, cada vez aparezca un tipo de prueba distinta.

Además de esto, una herramienta puede identificar el tipo de MAPTCHA que aparece, gracias al tamaño de la imagen que ve, y la cadena que aparece en la página web antes de la prueba; con ello, la máquina puede recargar las páginas tantas veces como quiera, hasta llegar a una prueba que conoce.

Otra debilidad muy importante del MAPTCHA, es que tiene retos repetidos, así la máquina, con resolverlo una vez, ya lo tendría resuelto para futuras veces que le pueda aparecer. Además de todo esto, la imagen que decimos repetida, quedaría con el mismo nombre, lo cual hace que sea mucho más sencillo de reconocer para la herramienta.

4.8. JDownloader burla los CAPTCHAS

JDownloader es uno de los más conocidos programas de descargas, utilizado por muchas personas. Descarga videos, música, etc. de Internet, y todo esto lo hace “resolviendo” los CAPTCHAS que las páginas de donde descarga le proponen. La cuestión es, ¿cómo JDownloader consigue superar los CAPTCHAS siendo una máquina?

Para ello, el sistema que emplea es el Janticaptcha, creado por los mismos desarrolladores del software JDownloader, el JDTeam. Según estos, el Janticaptcha, es una herramienta para encontrar objetos en imágenes.

Básicamente lo que hace este sistema es comparar el CAPTCHA que le pide en cada descarga, con los habituales que muestra cada sitio web, y que tiene almacenados en su base de datos. El repertorio de la herramienta cuenta con cientos de imágenes de CAPTCHAS.

5. Desarrollo de un CAPTCHA

5.1. Consejos para crear un buen CAPTCHA

En este punto daremos algunos consejos para crear un CAPTCHA que sea seguro, viendo las principales deficiencias que tienen los de hoy en día. Sólo trataremos los de tipo imagen, ya que son los más utilizados, dejando a un lado otros tipos de CAPTCHAS estudiados en este trabajo.

El principal método para vulnerar CAPTCHAS visuales con letras distorsionadas, es la segmentación, por ello, esto es lo que hay que tener en cuenta, en mayor medida, a la hora de crear un nuevo CAPTCHA, hacerle seguro ante un proceso de segmentación.

Este proceso, como ya se ha comentado en el punto anterior, y hemos visto personalmente en las herramientas que se comentaban, lo que hace, básicamente, es dividir la imagen en partes e ir descifrando una a una. Antes del proceso de segmentación, lo que interesa es obtener sólo la imagen que queremos dividir, por lo que una buena herramienta anti-CAPTCHAS, lo primero que hace es dejar el esqueleto de este.

El primer paso para la ruptura sería limpiar el fondo de la imagen, esto es conocido como pre-procesamiento. Se toma un CAPTCHA original y se limpia el ruido externo a la imagen que nos interesa descifrar (los caracteres que contiene), de manera que quede lo más simple posible.

¿Cómo podemos crear un CAPTCHA que sea seguro ante esta limpieza inicial? Para empezar, usaremos el ruido como técnica más básica, pero eficiente, para confundir a la máquina a la hora de limpiar la imagen. Para que el ruido sea eficaz, logrando equivocar a la máquina, este debería ser del mismo color, o similar, al de las letras del CAPTCHA, sino, en lugar de proteger, incluso podría ser de ayuda para lo contrario, para hacerlo más vulnerable ante un ataque.

El principal ataque al ruido de una imagen es el conocido algoritmo de Gibbs. Este algoritmo iterativo funciona con el cálculo de la energía de los píxeles que conforman la imagen, eliminando los que están por debajo de un cierto umbral. En cada iteración, va eliminando los que considera necesarios, hasta que no quedan más píxeles adicionales por descartar, quedando la imagen que le interesa para descifrar.

Una buena opción de usar el ruido de base, sería hacer que el dibujo, o las líneas que formen el fondo, sean capaces de confundirse con los caracteres que forman el propio CAPTCHA. Sin embargo, un mal ejemplo de esto, sería utilizar siempre las mismas imágenes, o usar fotografías aleatorias pero con el mismo tipo de colores, tanto para la imagen de base como para los caracteres, ya que, para vulnerarlo, simplemente valdría con un aprendizaje de los colores por parte de la máquina.

Otra buena idea para crear un fondo de imagen seguro, sería que este tuviera colores similares al color de los caracteres. De hecho, la buena práctica, sería usar colores que fueran muy diferentes para el ojo humano pero que, sin embargo, sean muy similares para una máquina, es decir, que se encuentren muy cerca en el espectro de colores RGB. El espectro RGB podemos verlo en las siguientes imágenes:

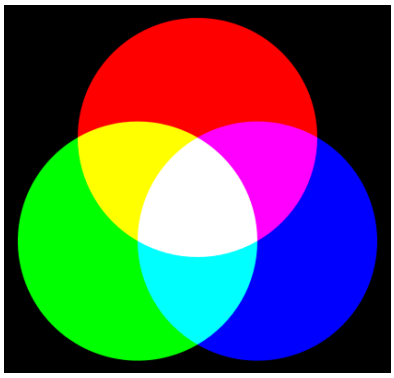

Ilustración 41

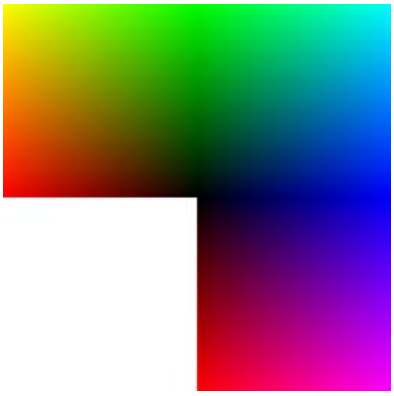

Ilustración 42

En muchos CAPTCHAS, para tratar de hacerlos más seguros, se combinan múltiples formas de confusión en el fondo de la imagen, es decir, incluyen todos, o gran parte, de los consejos que se han explicado y esto, al contrario de hacerlo más seguro, en realidad lo hace más inestable ante un ataque.

Sin embargo, no podemos confiar ciegamente en que usando estas técnicas se pueda crear un CAPTCHA seguro 100%, ya que, con un cierto aprendizaje de la máquina, tarde o temprano, es muy probable que sean capaces de romperlo.

Partiendo del punto en que la máquina ya haya conseguido superar el ruido de fondo, ya tiene el esqueleto de los caracteres que lo forman, el siguiente paso para descifrar el CAPTCHA sería la segmentación.

Ahora veremos técnicas para hacer, o intentar hacer, que nuestra prueba sea segura ante la segmentación.

Un ejemplo para esto, sería usar una línea que atraviese varios caracteres, se pueden utilizar líneas que atraviesen sólo algunos caracteres, o una línea grande que atraviese todo el conjunto.

En cuanto a la primera opción, una línea que cruce algunos caracteres, esta puede hacer que la máquina se confunda al leer cada carácter y puede creer que forma parte de la letra o número. Por ejemplo, tenemos una “l”, si usamos una línea que la cruce, podríamos confundir a la máquina y crea que es una “t”.

Sin embargo, esto es vulnerado fácilmente por el algoritmo de Gibbs, ya que tiene la facilidad de que, la línea que cruza los caracteres del CAPTCHA, es más fina que cada letra y cada número original; más adelante comentaremos sobre el grosor de la línea. El algoritmo de Gibbs, como ya se ha comentado antes, va limpiando píxel a píxel el carácter y se queda con el que supone que es el real.

En cuanto a la opción de la línea grande que cruza el CAPTCHA, no es vulnerado tan fácilmente por el algoritmo de Gibbs. Para romper esto, lo que se hace es una búsqueda de dicha línea en el CAPTCHA, fijándose en los bordes de este, y se intenta eliminarla entera. Esto de

"eliminarla entera" no es tan fácil, ya que hay que tener cuidado de no eliminar también parte de algún carácter, volviéndolo así irreconocible. Para eliminar la línea, se hace píxel a píxel, fijándose en los de su alrededor. Este trabajo hace la opción de la línea grande algo más segura que la línea que cruza determinados caracteres.

Veamos un ejemplo gráfico de esto:

Ilustración 43

Este mecanismo anti-segmentación puede ser bastante útil, si se aplica bien. Viendo lo comentado anteriormente, sobre cómo se puede vulnerar un CAPTCHA a través de esta línea, es decir, cómo hacen las herramientas para eliminar la raya, llegamos a la conclusión de algunos puntos que es necesario tener en cuenta a la hora de usar dicha línea para crear una prueba segura:

- Lo primero y principal, es que la línea sea del mismo color que el CAPTCHA, sino podrían usar el color de la línea para eliminarla.
- Las líneas deberían ser del mismo grosor que los caracteres, para hacerla más difícil de distinguir. Esto sobre todo es bueno aplicarlo a las líneas que cruzan sólo algunas letras y números. Cuando una raya cruza sólo determinados caracteres, es más fina que estos, y esa es su principal debilidad, haciendo que la línea sea del mismo grosor, podría ser confundible con parte del propio carácter.
- La longitud total de la línea no debería exceder por delante, ni por detrás, a los caracteres, o que supere a este lo mínimo, así no podrían utilizar los bordes para localizarla.
- Además, la longitud debería variar en cada prueba, con el fin de evitar un aprendizaje de su tamaño por parte de la máquina.
- Procurar que la línea mantenga la misma pendiente que el CAPTCHA original, que no tenga una inclinación extraña que denote que esta existe, o una inclinación muy diferente a la de los caracteres.

Otra opción para prevenir la segmentación, sería el solapamiento de los caracteres. Esta puede ser una buena técnica si se utiliza correctamente.

Un fallo muy común, y que, aunque exista solapamiento, no quedaría resuelto, es que en muchas ocasiones se conoce el número de caracteres que contiene, porque siempre se utiliza la misma cifra, por lo tanto, aunque los caracteres se solapen o estén muy juntos, la máquina busca el número concreto que debe tener, y así, por tanteo va descubriendo dónde se encuentra cada uno, separándolo del resto para descifrarlo.

La principal solución para este problema, es que la máquina no conozca el número de caracteres que tiene el CAPTCHA, y además, que no conozca el tamaño de dichos caracteres; de este modo, tampoco podrá tantear dónde está cada número o letra, tomando su tamaño como unidad de medida. La mejor técnica es que el tamaño y la longitud del CAPTCHA sean aleatorios.

Una vez resueltos estos problemas de tamaño y longitud, hacer un solapamiento de caracteres, podría ser una buena técnica, porque le sería costoso de reconocer. También hay que tener cuidado con el solapamiento, ya que, un exceso de este, puede hacerlo tan complicado que los propios usuarios no sean capaces de resolverlo.

Un ejemplo gráfico del solapamiento sería:

Ilustración 44

Para hacer un CAPTCHA seguro, a mi parecer, la principal técnica debería ser utilizar la mente humana. No usarla simplemente para leer unas letras y cifras o escuchar un audio, sino para pensar y averiguar el resultado de un algoritmo. La capacidad de pensar es la principal virtud de los seres humanos frente a las máquinas, que se basan en un aprendizaje de las características de los CAPTCHAS, tales como su tamaño, número de caracteres…, a un ser humano real no es necesario enseñárselo para que lo aprenda, sino, simplemente, mostrárselo.

CAPTCHAS basados en esto, serían los que realizan preguntas sencillas que todos conocemos y sabríamos responder, como ¿de qué color es el cielo?, sin embargo, se debería crear un conjunto infinito de dichas preguntas, para hacer difícil el aprendizaje por parte de una herramienta anti-CAPTCHA. Si el conjunto es relativamente pequeño, fácilmente será memorizado por una de ellas.

Existen muchas otras posibilidades, que no sean estas simples preguntas, y que, sin embargo, se necesite la visión y capacidad de pensar de un usuario, sin hacerlo tan complicado que, por ejemplo un niño o un anciano, no sean capaces de responder.

Una de las principales características del CAPTCHA seguro, sea del tipo que sea, es que existan multitud de casos, para evitar el aprendizaje de la máquina. Y para hacer uno muy seguro, sería conveniente crearlo de tal forma, que no sea posible crear un algoritmo capaz de resolver todos los CAPTCHAS del mismo tipo.

En los puntos siguientes, se explican varias propuestas, en todas ellas se utiliza la capacidad humana de pensar, y hay múltiples opciones dentro de ellas para que le aparezcan al usuario. Todas las pruebas serán visuales, ya que es el tipo de CAPTCHA que se ha tratado en este punto.

A día de hoy, las herramientas que se han encontrado y estudiado, no serían capaces de romperlos, ya que son de estilo diferente a los ya existentes. Esto no impide que se pueda crear un nuevo algoritmo que consiga vulnerarlos en cuestión de segundos.

5.2. Propuesta de CAPTCHA 1

5.2.1. El CAPTCHA

En este punto se comentará una de las propuestas de CAPTCHA seguro, es de tipo imagen, como ya se ha dicho, además, contendrá letras distorsionadas, así que veremos todos los consejos que hemos comentado antes para este caso.

La idea básica de la prueba es que el usuario coloque alfabéticamente las letras que verá en varias imágenes.

El primer paso consiste en que, el usuario deberá distinguir las letras que forman cada imagen, estas podrán aparecer distorsionadas, superpuestas, descoloridas, con líneas que las crucen, etc., todo ello para dificultar su visión y descubrimiento.

Aquí vemos un ejemplo de las imágenes que formarían la prueba:

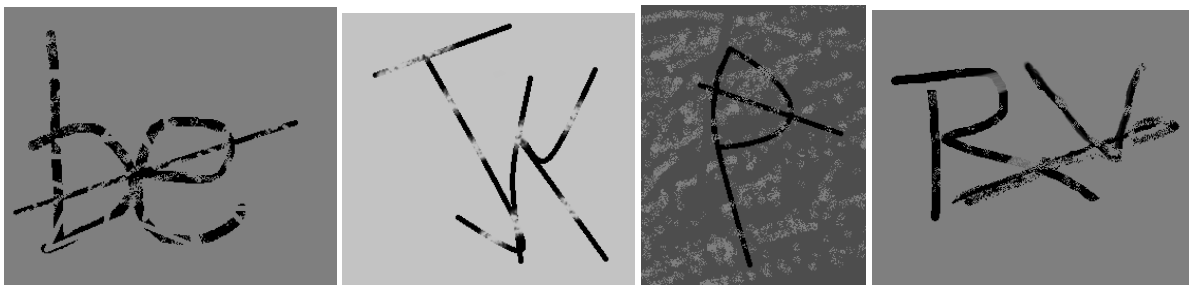

Ilustración 45

Las letras que forman la imagen, estarán seguidas entre sí, pero no tienen por qué estar seguidas inmediatamente en el abecedario, esto lo explicaremos con más detalle en el apartado de "Cómo crear este CAPTCHA". Las imágenes que forman el CAPTCHA en conjunto, no estarán colocadas, de eso se trata la prueba, y esto es lo que en realidad tiene que hacer el usuario, poner en orden las imágenes para que queden las letras colocadas alfabéticamente.

Por lo que aquí vemos un ejemplo más real de lo que se encontraría un usuario, con las imágenes colocadas aleatoriamente:

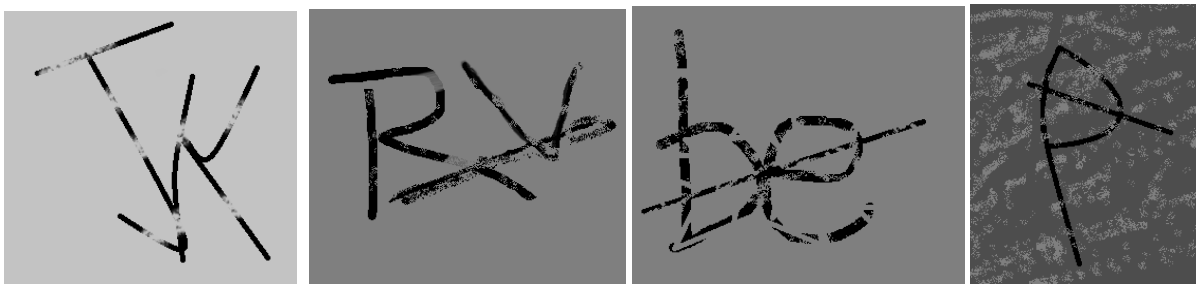

Ilustración 46

Una vez el usuario sepa el orden alfabético en el que pretende colocar las letras, y por tanto las imágenes, llegamos al último paso; el usuario no debe escribir exactamente las letras que ve, sino que cada imagen, formada por una o dos letras, se equivale a un número. El usuario, en lugar de escribir las letras en la solución, deberá escribir el número que corresponde a cada imagen en el orden que considere correcto.

El número aparecerá colocado en la parte superior de cada fotografía.

Aquí vemos un ejemplo de cómo quedaría finalmente el CAPTCHA:

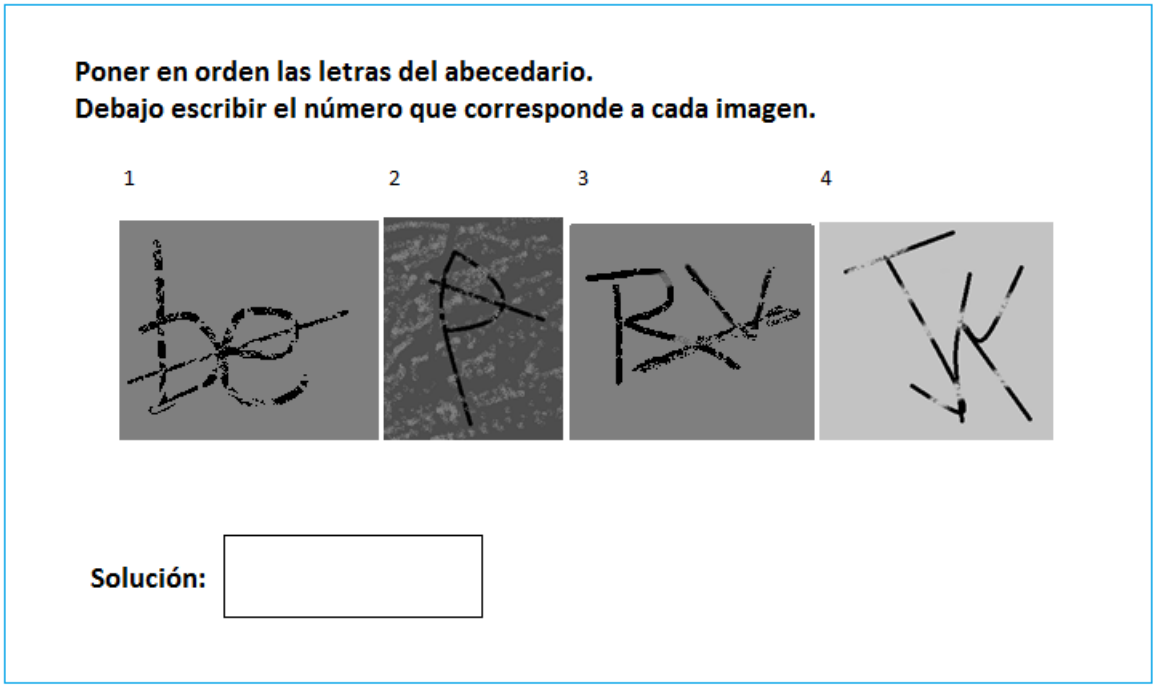

Ilustración 47

El usuario en la solución debería escribir: **1423**

5.2.2. ¿Cómo crear este CAPTCHA?

Imágenes del CAPTCHA

Para crear el CAPTCHA, primero tomamos las 27 letras del abecedario, en orden. Para formar cada imagen se necesitan una o dos letras, esta cifra será elegida aleatoriamente por la máquina, por lo que podemos tener fotografías de una o de dos letras, esto lo hará más seguro frente a la segmentación. El total de imágenes será siempre cuatro, sin embargo, el total de caracteres puede variar entre ocho y cuatro.

Para crear las imágenes, primero se cogerán ocho letras al azar, siempre en orden entre sí, pero no tiene por qué ser orden inmediato. Para hacer cada imagen, se irán cogiendo las letras seleccionadas de una en una o de dos en dos, según se ha dicho antes. Por lo que se pueden utilizar los ocho caracteres preseleccionados, o que finalmente sobre alguno.

En cuanto a las letras seguidas pero no inmediatamente, esto quiere decir que podemos tener los caracteres B, D. Para formar una imagen podemos coger el par B-D, siempre y cuando, otra imagen no contenga la letra C.

Pongamos un ejemplo más completo de esto, se seleccionan las letras (siempre en orden): A, D, F, H, J, P, W, Z. Las imágenes se deben ir formando en ese mismo orden: A-D; F; H-J; P. Sobrarían la W y la Z. El total de todos los caracteres juntos sería, de cuatro a ocho letras colocadas alfabéticamente entre sí: A, D, F, H, J, P.

Una vez tengamos las cuatro imágenes con sus letras correspondientes, estas se colocan al azar dentro del CAPTCHA.

Distorsión y técnicas anti-segmentación

Una vez tengamos las imágenes con las letras, el segundo paso consiste en “codificarlas”, es decir, seguir las técnicas de distorsión que conocemos, para intentar dificultar su visión y consiguiente descifrado.

Las técnicas de distorsión que vamos a emplear para las imágenes serán:

- Colores similares en letras y fondo. Para la base, se utilizarán siempre colores que sean similares a los de la las letras, estos colores deberán estar cerca en el espectro de colores RGB. El ejemplo dado es bastante simple, ya que sólo se han utilizado el gris y el negro.
- Imágenes de fondo. El fondo de las imágenes puede ser simple o con algún dibujo en él, que haga más complicada la visualización de las letras.
- Letras en varios colores. Para que sea más difícil distinguir cada carácter, quitando el fondo con el algoritmo de Gibbs, algunas letras pueden estar en varios colores.

Ya que el ejemplo propuesto es bastante sencillo, en cuanto a las técnicas de distorsión se refiere, aquí vemos una imagen de otro caso, que recoge todas las técnicas que se acaban de explicar:

Ilustración 48

Cuando tenemos distorsionadas las letras, además, añadiremos en algunas de ellas una línea, esta puede atravesar una sola letra, o ambas letras. A la línea también se le aplicará el difuminado o distorsión al que se someta la imagen, y se tendrán en cuenta los consejos explicados en el punto anterior, en cuanto a que la línea será del mismo grosor que las letras, tendrá la misma pendiente y se procurará que no exceda el tamaño de las letras. Además de la raya, se incluirá el solapamiento entre ambas letras en algunas de las imágenes.

Solución

El sistema tendrá la solución a cada prueba, porque conoce lo que contiene cada imagen y el orden que estas deben seguir, esto lo asocia al número que se corresponda con cada fotografía, y compara el resultado real con el proporcionado por el usuario.

Si los resultados coinciden, el CAPTCHA supone que está tratando con un usuario y habrá pasado la prueba. Si no lo supera, le dará otra oportunidad, mostrando un nuevo ejemplo con cuatro imágenes distintas.

5.2.3. ¿Por qué este CAPTCHA es seguro?

La principal, y mejor característica que tiene este CAPTCHA, es que el usuario debe pensar el orden de las letras, y no sólo eso, sino que debe escribir el número que se corresponde con la imagen, y no solo las letras que esta contiene.

Desde un principio, se intenta que la prueba sea segura, haciendo que distinguir las letras sea complicado, tanto para un usuario, como para una máquina. Para ello, se emplean las técnicas de distorsión de la imagen, explicadas en el punto anterior (punto 5.1).

Lo principal para evitar la segmentación, es el tema de los colores para confundir las letras, tanto con el fondo, como combinando varios colores en la misma letra, e intentar hacer creer a la máquina que son caracteres distintos, o que son parte del fondo. A esto se le suma la distorsión de los caracteres, el solapamiento y una posible línea que cruce las letras.

Además de esto, tenemos la posibilidad de que en la imagen haya una o dos letras, de este modo, la máquina no irá buscando siempre un número de caracteres concreto, ya que podría encontrarse con varias posibilidades.

Como ya sabemos por experiencia con otros CAPTCHAS, las técnicas de distorsión, o las técnicas de anti-segmentación, no son suficientes, por eso se ha añadido la tarea de colocar las letras e imágenes. En un CAPTCHA normal, la máquina únicamente tiene que descifrar y reproducir los caracteres; en este caso, además de eso, tiene la dificultad añadida de colocar alfabéticamente todas las letras con las que se encuentra.

Una vez tenga colocadas todas las letras, y por tanto, las fotografías, deberá saber la cifra que tienen asignada en su parte superior, y dar como resultado, estos números en el orden correcto.

5.3. Propuesta de CAPTCHA 2

5.3.1. El CAPTCHA

En la segunda propuesta de CAPTCHA, se pretende utilizar la mente humana capaz de distinguir perfectamente diferentes objetos, aunque estos estén unidos o superpuestos.

La prueba consiste en que, el usuario ve una imagen y debe distinguir dentro de ella unos objetos y colores, para superarla, el usuario debe decir el color que pinta la mayoría de los cuerpos que aparecen. Por ejemplo, si en la fotografía hay cinco objetos, de los cuales tres son azules y dos verdes, el usuario deberá escribir “azul” en la solución.

Veamos un ejemplo de la imagen que aparecería en este CAPTCHA:

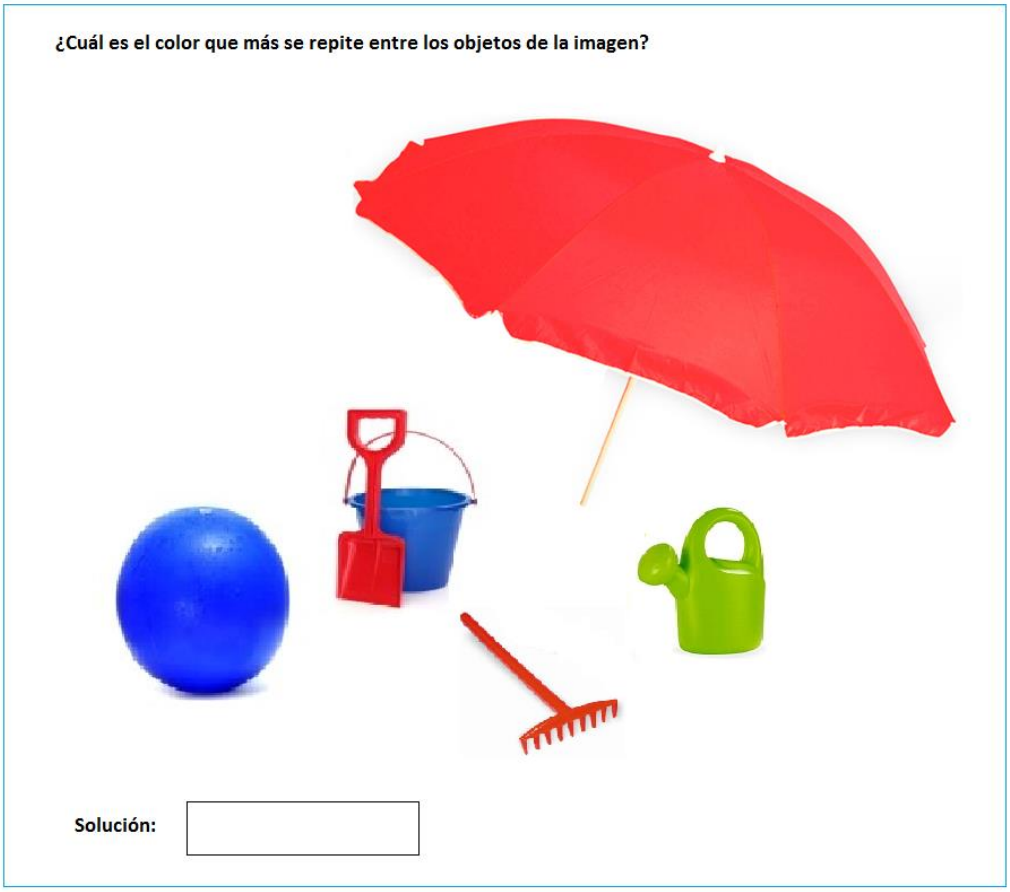

Ilustración 49

En la foto vemos una sombrilla, una pala y un rastrillo rojos, un cubo y una pelota azul y una regadera verde, la solución que debe escribir el usuario sería: “rojo”.

Este CAPTCHA intenta utilizar la capacidad de visualización y conocimiento humana, para distinguir, no sólo los colores, sino los objetos de cada color. Los objetos serán siempre de un único color, para evitar confusiones, sin embargo, los cuerpos pueden estar superpuestos o unidos; el usuario podrá distinguir fácilmente que son dos objetos distintos, sean del mismo o de distinto color entre sí, sin embargo, para una herramienta no será tan fácil.

El CAPTCHA parece sencillo, y por lo tanto puede parecer fácil de romper por una herramienta, pero no lo es tanto, ya que, para empezar, no siempre servirá coger el color más abundante, porque puede darse el caso de que los objetos con el color más repetido sean más pequeños, y haya un objeto muy grande de otro color, pero sólo uno, por lo que nunca sería mayoría, y la solución que da la máquina sería errónea. También puede ocurrir que los cuerpos estén superpuestos, y de este modo, la herramienta lo trataría como sólo un objeto, siendo varios en realidad. En siguientes apartados explicaremos estos y veremos más imágenes con ejemplos.

5.3.2. ¿Cómo crear este CAPTCHA?

Selección de objetos

Para empezar, nos harán falta muchas imágenes de objetos de diferentes colores, siempre de un sólo color cada uno. Las imágenes básicas serán los cuerpos pintados de un color, la foto que aparece en el CAPTCHA con varios de ellos se formará después.

Una vez que tengamos los objetos, cogemos una cantidad de 6-7-8, tampoco compensa que sean demasiados y el usuario pierda mucho tiempo contando. Cogemos esa cantidad, y juntamos todos los objetos aleatoriamente en una imagen con fondo blanco.

Los cuerpos serán colocados aleatoriamente, por lo tanto, habrá veces que estén algunos superpuestos, otras no, pueden estar unidos objetos del mismo color o de distinto color… infinidad de opciones que la máquina no puede controlar.

Control de objetos

Los objetos se colocan aleatoriamente en la imagen, pero el algoritmo del CAPTCHA deberá controlar los objetos que se colocan, ya que siempre tiene que haber algún color con más entidades que de los demás.

La idea es, primero colocar aleatoriamente seis o siete objetos sobre un fondo blanco, el propio CAPTCHA elegirá el número concreto. Una vez se tienen los seis o siete objetos iniciales, tendremos un posible objeto número 7/8 dependiendo de los ya colocados, este objeto se colocará si ocurre lo siguiente:

- El CAPTCHA conoce el color de cada objeto, con esa información, cuenta cuántos hay de cada color, si existe el caso de que hay dos colores como los más repetidos (empatados), entonces añade un nuevo objeto de alguno de esos dos colores, y así, el color más repetido será sólo uno de los dos. Pongamos un ejemplo de esto: se ponen seis objetos en la imagen, tres son rojos y tres amarillos, entonces se añadirá un nuevo objeto que sea rojo, así el color más repetido será únicamente este último.

- También se añadirá el objeto número 7 u 8 en el caso de que todos los objetos sean del mismo color, el objeto nuevo que se añadirá será de cualquier color excepto del color del que son todos los demás. Un ejemplo de este caso: se ponen seis objetos de color rojo, entonces se añadirá uno nuevo amarillo.

Por ello, las imágenes del CAPTCHA siempre serán de 6-7-8 objetos, sin ser nunca una cifra fija, de este modo lo hacemos más seguro ante el posible aprendizaje de una herramienta.

Solución

El CAPTCHA siempre sabrá cuántos objetos contiene la imagen, y de qué color es cada uno de ellos, con esta información, llevará la cuenta de cuántos objetos son de cada color, y por lo tanto, sabe cuál es el más repetido.

Cuando el usuario inserte un color en la solución, comparará este con el que tiene registrado como solución correcta, si coinciden, permitirá el acceso, sino, volverá a mostrar otro CAPTCHA con una nueva imagen y nuevos objetos.

5.3.3. ¿Por qué este CAPTCHA es seguro?

Vamos a ver algunas razones por las que este CAPTCHA es más seguro que los ya existentes.

Para empezar, parece que visualizar colores es relativamente fácil para una máquina, pero no se trata sólo de distinguir un color en un cuerpo, hay que distinguir y controlar qué objetos son de cada color.

El primer punto a favor, es que el tamaño de los objetos es muy variable, por lo tanto, si una herramienta intenta buscar el color que más tamaño ocupa, no siempre acertará, porque puede darse el caso de tener dos objetos muy grandes de un color, y tres objetos mucho más pequeños de otro color, por lo tanto, aunque el tamaño en superficie del segundo color sea menor, hay más objetos pintados de dicho color y sería el que más se repite. Para que quede más claro veamos un ejemplo concreto de este caso:

Ilustración 50

Aquí el color que predomina claramente es el rojo, ya que la casa roja es bastante más grande que todo el resto de objetos juntos, sin embargo, sería el color verde el que más entidades tiene, y esta sería la solución correcta.

Otra característica que lo hace más seguro, es que puede ocurrir que los objetos estén unidos o superpuestos. En este caso, si son distintos colores, no habría problema para la máquina, pero si se trata de objetos del mismo color unidos, la herramienta lo trataría como un único objeto, y el usuario vería perfectamente que son dos distintos. Aquí vemos varios ejemplos de esto:

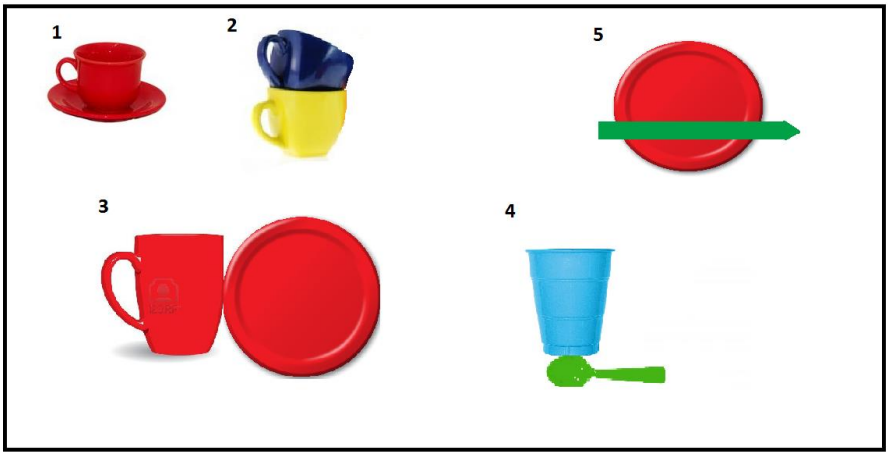

Ilustración 51

Aquí tenemos varios casos que comentar, primero, la imagen 1 de la taza y el plato rojo, un usuario puede ver claramente que se trata de dos objetos, aunque estén uno encima del otro, sin embargo, una máquina esto lo trataría como el mismo. En el caso de la imagen 2, también son dos objetos superpuestos, pero son de distinto color, por lo que una máquina sí sería capaz de reconocerlo. En la imagen 3, son dos objetos unidos, siendo de exactamente el mismo color, a una máquina le resultaría muy difícil distinguir si son dos objetos o uno solo; sin embargo, en la imagen 4, son dos objetos unidos pero de distinto color, por lo que sí podría diferenciarlos fácilmente. Por último, en la imagen 5, un objeto está cortado por otro, un usuario puede ver que se trata de dos, un plato con una flecha cruzada, sin embargo, ¿una máquina sería capaz de reconocer los dos objetos, o lo trataría cómo si fueran tres?

Para que no haya mucha dificultad para el usuario, las entidades serán siempre de un único color, sin embargo, los tonos del color pueden variar, esto también puede ser un problema para una herramienta a la hora de distinguirlos. Además, dentro de un mismo dibujo, puede darse el caso de que lo formen varios tonos del mismo color, ponemos el caso del árbol que se vio en la ilustración 50:

Ilustración 52

En este ejemplo, una máquina, al ver distintos tonos del mismo color, probablemente piense que son dos objetos unidos, o uno que corta al otro, y lo tome como dos cuerpos distintos, siendo sólo uno.

Además, en este CAPTCHA, como ya se ha explicado, el número de objetos será variable entre 6-7-8, esto nos sirve para controlar que siempre haya un color que supere al resto en número de objetos, y también para hacerlo más seguro. Haciendo que haya un número variable de cuerpos, evitamos que la máquina haga un aprendizaje de esto, y por lo tanto evitamos, que siempre busque dicho número de objetos, lo que haría que unirlos o superponerlos no fuera provechoso. De este modo, si en un principio ve seis objetos por ejemplo, no sabrá si ese es el número real o si hay algún cuerpo sobre otro.

Este CAPTCHA, sin embargo, tiene una pega, y es para las personas daltónicas, para estos usuarios, la prueba sería imposible de resolver.

5.4. Propuesta de CAPTCHA 3

5.4.1. El CAPTCHA

En esta tercera y última propuesta, el usuario verá una secuencia de letras y números, esta imagen será dada por el CAPTCHA, y una vez vistos y reconocidos los caracteres, deberá reproducirlos en una especie de pizarra.

La prueba en sí, consiste en escribir lo que se ve en la imagen dada, pero la dificultad es que no será escrito con un teclado, sino que el usuario va a dibujar los caracteres que ve en una imagen en blanco, utilizando el ratón.

Veamos un ejemplo de esto, en la imagen aparece el CAPTCHA inicialmente, lo que se encontraría el usuario:

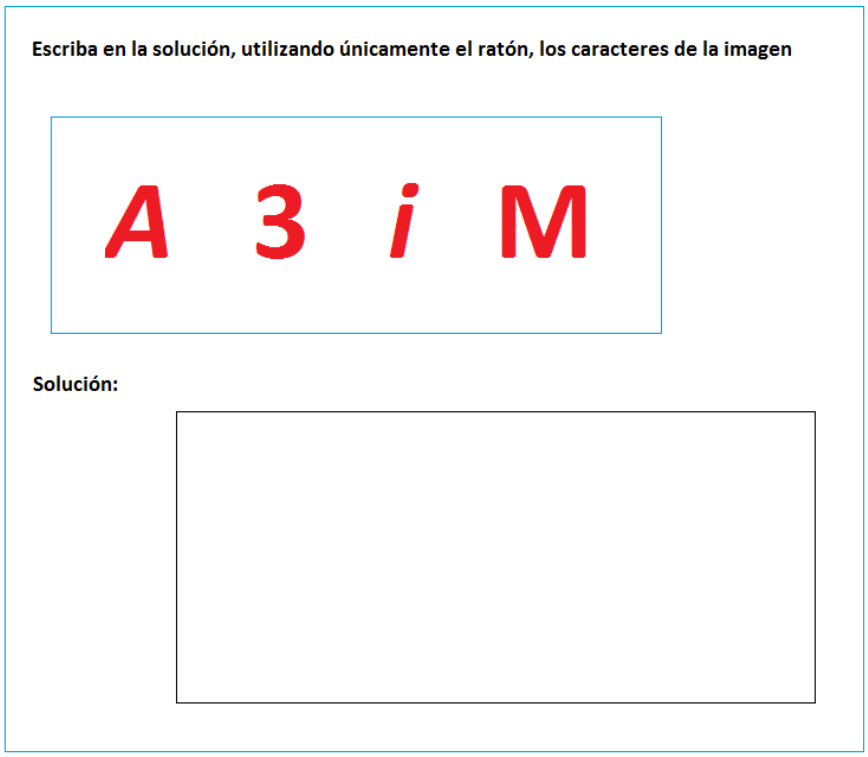

Ilustración 53

Para escribir la solución se usará el ratón, de forma similar a la herramienta Paint de Windows, cuando se utiliza el lápiz propio de dicho programa.

Y en la siguiente imagen vemos cómo quedaría después de ser solucionado por el usuario:

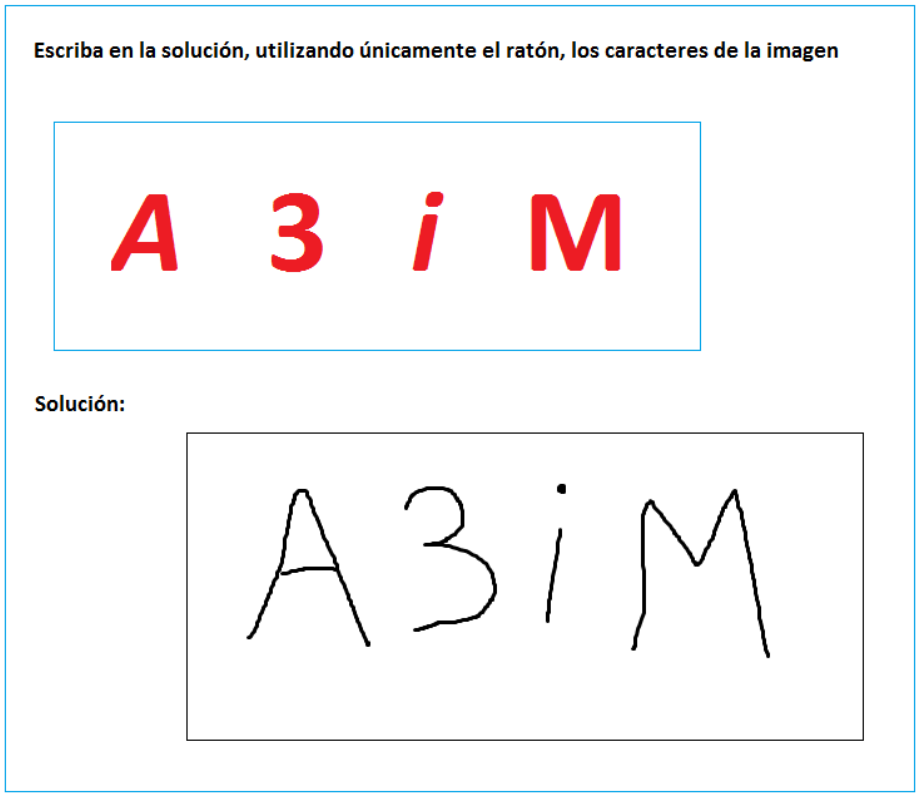

Ilustración 54

La imagen que el internauta tiene que reproducir aparecerá normalmente, sin distorsionado, ni líneas que la crucen, ni superposiciones, ya que todas estas fórmulas, como hemos visto, no son 100% seguras, de este modo lo hacemos más fácil para el usuario, que ya tiene la "tarea" de escribirlas personalmente. Recordemos que, la idea fundamental de este CAPTCHA no es descubrir cuál es la imagen que se da inicialmente, sino escribirla, por tanto, que la herramienta sepa cuál es la imagen original, no es un problema para su seguridad.

La tarea que tiene el usuario de escribir la imagen en la pizarra que se da, es precisamente la dificultad para la máquina, que para superar la prueba, también tendría que pintar los caracteres que visualiza.

Los caracteres que forman la imagen estarán separados, para que el usuario los escriba también de esta forma, y sean fáciles de reconocer, una vez escritos, por el propio CAPTCHA. Lo formarán letras, tanto en mayúsculas, como en minúsculas, y números.

5.4.2. ¿Cómo crear este CAPTCHA?

Selección de imagen

Lo primero, es generar una imagen que contenga los caracteres que el usuario debe reproducir. Como ya se ha comentado, los caracteres estarán separados, ni unidos, ni superpuestos, y siempre serán letras y números.

Los caracteres serán siempre de color rojo o negro, y el tamaño será medio-grande, para facilitarle la visión a la persona.

Una vez tengamos la imagen creada, se le muestra al usuario, y debajo de esta imagen, se pondrá una pizarra en blanco, donde podrá escribir la solución utilizando el ratón.

Solución

Cuando el usuario reproduce la imagen en la pizarra, el CAPTCHA comprueba la solución.

Para comprobar esta solución, el funcionamiento del algoritmo será similar al de las propias herramientas para romperlos, es decir, va a tratar de descifrar lo que el usuario ha escrito, aunque su trabajo será algo más fácil.

Lo que va a hacer primero es la segmentación, el CAPTCHA ya sabe cuántas letras ha tenido que escribir el usuario, porque él es quien ha dado la imagen inicial, por lo tanto, en la segmentación, simplemente va a tener que buscar el número de caracteres que ya conoce.

Vemos un ejemplo de lo que haría el proceso de segmentación para sacar cada carácter:

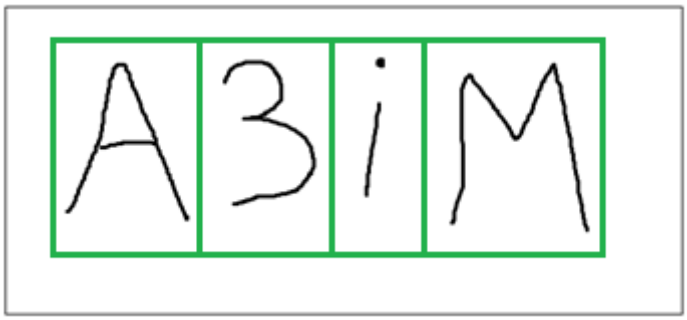

Ilustración 55

Una vez tenga los caracteres por separado, compara uno a uno con los de la imagen original, si coinciden todos ellos, lo da por válido y la prueba estará superada.

Si falla en tan sólo una letra o número, la prueba será fallida y volverá a mostrar una nueva imagen.

5.4.3. ¿Por qué es seguro este CAPTCHA?

Este CAPTCHA, a mi parecer y según lo estudiado para este proyecto, es de los más seguros que he visto hasta ahora, no he encontrado ninguna herramienta capaz de reproducir en una imagen en blanco los caracteres que visualiza.

Para tratar esta prueba, habría que crear una herramienta de cero capaz de romperla, primero utilizar alguna ya existente capaz de leer los caracteres, y luego crear una que sea capaz de escribir como si un usuario utilizase un ratón, escribir caracteres propios de un teclado no vale para superarla, el CAPTCHA simplemente va a reconocer la escritura a través del cursor del ratón.

5.5. Debilidades de los CAPTCHAS creados

En el siguiente punto vamos a ver con qué facilidad se podrían saltar los CAPTCHAS creados, en referencia a las herramientas vistas en el trabajo. También contaremos con algún factor adicional que se le pueda añadir fácilmente a las herramientas.

5.5.1. CAPTCHA 1

Para comenzar a romper este CAPTCHA, nos serviría, en principio, cualquier herramienta existente, que sea capaz de leer y descifrar caracteres distorsionados.

La primera idea propuesta consiste en unas imágenes distorsionadas, que una herramienta sencilla es capaz de resolver, sin embargo, sólo con leer las imágenes y averiguar su contenido, no es suficiente, el siguiente paso es colocar las letras que contiene, en orden alfabético.

Para pasar este segundo paso, sería suficiente incluir, en la herramienta elegida, un abecedario, y que fuera comparando imagen a imagen, letra a letra y colocando dichas letras en orden. Este apartado no lo incluye ninguna de las herramientas estudiadas, ya que no se ha visto, hasta el momento, ningún ejemplo de CAPTCHA como este, sin embargo, sería relativamente fácil incluirlo.

Además, una debilidad importante de este sería que, con conocer una de las letras que aparece en la imagen (tanto si la componen una o dos), ya podemos comparar esta letra con otra, de una imagen distinta, y así saber el orden de estas dos fotografías. Si la primera de ellas empieza por la letra O, y la segunda imagen empieza por la letra A, ya sabemos que están en orden inverso, sin importar cuál es el segundo carácter de ambas.

Por último, aún nos quedaría el paso de que, no hay que colocar y escribir las letras, sino el número con el que se corresponde cada imagen. Esto sería algo más complicado, ya que la herramienta debe tener en cuenta, en todo momento, el orden de las imágenes, tanto el orden inicial en el que las encuentra, como el orden final en el que quedan colocadas, y escribir dichos números en la solución.

Las herramientas estudiadas, actualmente, no serían capaces de romper este ejemplo de CAPTCHA propuesto, sin embargo, con un par de cambios en ellas, la prueba no sería muy difícil de vulnerar.

5.5.2. CAPTCHA 2

Esta propuesta, se diferencia más que la anterior de los CAPTCHAS más comunes y populares hoy en día, por lo tanto, las herramientas conocidas tendrían más dificultad para superar la prueba. Su porcentaje de acierto, probablemente, disminuiría considerablemente.

Para comenzar a romper el CAPTCHA, se tendría que hacer una búsqueda de objetos, una de sus principales debilidades es que ya sabe que, como mínimo, van a ser seis objetos y como máximo ocho. Por lo tanto, si la herramienta encuentra de entrada ocho "bultos", posiblemente esos sean los objetos. Luego sería necesario el estudio de los colores.

Encontrar y distinguir los diferentes colores, resultaría sencillo para una herramienta, pero la principal fortaleza de este CAPTCHA es la superposición o unión de los objetos. Esto también puede ocurrir con los caracteres, como se ha visto con otras pruebas, pero existen muchos más ejemplos de objetos, que de letras y números, por lo que resultaría más difícil el aprendizaje de una máquina en este caso.

Hoy por hoy, esta propuesta no sería vulnerada por ninguna de las herramientas vistas. Además, no serviría con añadir un par de cambios a las ya conocidas, porque es un CAPTCHA muy diferente a lo que podemos encontrar.

5.5.3. CAPTCHA 3

La principal fortaleza de este CAPTCHA sería saber escribir en una pizarra, simulando el puntero de un ratón.

Distinguir los caracteres que se proponen para escribir, sería muy fácil de ver por parte de una herramienta, de hecho, los caracteres que se proponen, no tienen distorsionado, ni ninguna otra técnica que dificulte su visualización, ya que ese no es el principal reto. Una vez descubiertos los caracteres hay que escribirlos "a mano".

Ninguna de las herramientas estudiadas sería capaz de superar esta prueba. Sería necesaria la creación de una herramienta, completamente nueva, que fuera capaz de escribir simulando el puntero de un ratón.

6. Conclusiones y trabajos futuros

Los CAPTCHAS normalmente se utilizan para proteger los sitios web de ataques “bots”, es decir, de máquinas que se hacen pasar por usuarios interaccionando en la web.

Como hemos visto a lo largo del trabajo, existen muchos tipos de CAPTCHAS, muy diferentes entre sí, CAPTCHAS de imágenes (de caracteres o fotografías para diferenciar), de audio, de video, matemáticos, pruebas que consisten en jugar con la máquina… sin embargo, todos, o la gran mayoría de ellos, son ineficientes.

Existen máquinas desarrolladas expresamente para superar estas pruebas, y la mayoría de las veces, consiguen su objetivo con un alto porcentaje de acierto. Cuando aparece un nuevo CAPTCHA, hay un nuevo reto para los desarrolladores de estas herramientas.

Esto también supone otro reto, y es la creación de un CAPTCHA seguro ante las herramientas existentes, y con la intención de seguirlo siendo ante futuras amenazas. Esto es lo que se ha intentado en este trabajo, con la creación de tres nuevas ideas de CAPTCHAS. Todos son de tipo imagen y presentan novedades con respecto a lo ya conocido.

Como posible trabajo futuro, se propondría crear una herramienta anti-CAPTCHA, que fuera capaz de superar, con un alto grado de acierto, los propuestos aquí.

Referencias

1- http://www.captcha.net/ On date: 10/01/2013

2- http://en.wikipedia.org/wiki/CAPTCHA On date: 10/01/2013

3- http://www.hscripts.com/scripts/php/HCIC/form.php On date: 10/01/2013

4- http://notebookypc.com/la-verdadera-historia-de-los-captcha On date: 21/01/2013

5- http://www.svcommunity.org/forum/galeria-tecnologia/los-peores-captchas-de-la-historia/ On date: 21/01/2013

6- http://jose-juan.computer-mind.com/jose-juan/Captcha-PHP.php On date: 25/01/2013

7- http://www.decodigo.com/2009/09/verificacion-viusal-en-formularios.html On date: 30/01/2013

8- http://research.microsoft.com/en-us/um/redmond/projects/asirra/ On date: 15/02/2013

9- http://research.microsoft.com/en-us/um/redmond/projects/asirra/installation.aspx On date: 15/02/2013

10- http://sourceforge.net/projects/humanauth/ On date: 20/02/2013

11- http://blog.wintercore.com/2008/05/04/toward-a-new-generation-of-audio-captchas/ On date: 27/02/2013

12- http://www.nucaptcha.com/ On date: 01/03/2013

13- http://www.puzzlecaptcha.com/ On date: 08/03/2013

14- http://www.google.com/recaptcha/learnmore On date: 15/03/2013

15- http://campusvirtual.unex.es/cala/epistemowikia/index.php?title=Prueba_o_test_de_Turing On date: 25/03/2013

16- http://inteligenciaartificialudb.blogspot.com.es/2008/01/concepto-caractersticas-y-metodologas.html On date: 25/03/2013

17- http://www.fourmilab.ch/random/ On date: 10/04/2013

18- http://soydondenopienso.wordpress.com/2011/05/28/los-captchas-de-audio-a-punto-de-hacerse-inservibles/ On date: 25/04/2013

19- http://www.vsantivirus.com/16-04-08.htm On date: 26/04/2013

20- http://decaptcha.biz/ On date: 30/04/2013

21- http://elie.im/blog/security/how-we-broke-the-nucaptcha-video-scheme-and-what-we-propose-to-fix-it/ On date: 01/05/2013

22- http://jdownloader.org/knowledge/wiki/gui/configuration/advanced-view/janticaptcha On date: 04/05/2013

23- http://alt-tab.com.ar/como-jdownloader-salta-los-captchas/ On date: 04/05/2013

24- http://130.203.133.150/viewdoc/summary;jsessionid=634366C411BFC4700150A770308C0F3F?doi=10.1.1.211.151 On date: 15/02/2013

25- Jeffrey P. Bigham and Anna C. Cavender. *Evaluating Existing Audio CAPTCHAs and an Interface Optimized for Non-Visual Use*. CHI 2009. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

26- Carlos Javier Hernandez-Castro, Arturo Ribagorda. *Remotely telling humans and computers apart: an unsolved problem*. In Proceedings of the iNetSec 2009, IFIP AICT 309.

27- Carlos Javier Hernandez-Castro, Arturo Ribagorda, Yago Saez. *Side-channel attack on labeling CAPTCHAs*. In *http://arxiv.org/abs/0908.1185/* .

28- Elie Bursztein, Matthieu Martin, John C. Mitchell. *Text-based CAPTCHA Strengths and Weaknesses*. ACM Computer and Communication security 2011.

29- Elie Bursztein, Steven Bethard, John C. Mitchell, Dan Jurafsky, and Celine Fabry. *How good are humans at solving captchas? A large scale evaluation*. In Security and Privacy, 2010.

30- Carlos Javier Hernandez-Castro, Arturo Ribagorda, Julio Cesar Hernandez-Castro. *ON THE STRENGTH OF EGGLUE and other Logic CAPTCHAs*. In proceeding of: SECRYPT 2011 - Proceedings of the International Conference on Security and Cryptography, Seville, Spain, 18 - 21 July, 2011. SECRYPT 2011: 157-167.

31- Carlos Javier Hernandez-Castro, Arturo Ribagorda, Yago Saez. *SIDE-CHANNEL ATTACK ON THE HUMANAUTH CAPTCHA*. In proceeding of: SECRYPT 2010 - Proceedings of the International Conference on Security and Cryptography, Athens, Greece, July 26-28, 2010. SECRYPT 2010: 59-65.

32- Carlos Javier Hernandez-Castro, Arturo Ribagorda. *Pitfalls in CAPTCHA design and implementation: The Math CAPTCHA, a case study*. Computers & Security 29. 2010.

33- Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, Manuel Blum. *ReCAPTCHA: Human-Based Character Recognition via Web Security Measures*. Science Magazine, 2008, Vol. 321. no. 5895, pp. 1465 - 1468.

34- K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski. *Designing human friendly human interaction proofs (HIPS)*. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 711-720). ACM.

35- Monica Chew and J. D. Tygar, UC Berkeley. *Image Recognition CAPTCHAs*. In Proc. of ISC 2004, pp. 268-279. A longer version as UC Berkeley Computer Science Division technical report UCB/CSD-04-1333.

36- Athanasopoulos, Elias and Antonatos, Spiros. *Enhanced CAPTCHAs: Using Animation to Tell Humans and Computers Apart*. IFIP International Federation for Information Processing 2006.

37- L. von Ahn, M. Blum, N.J. Hopper, and J. Langford. *CAPTCHA: Using hard AI problems for security*. In EUROCRYPT 2003, Warsaw, Poland, v. 2656 of LNCS, pp. 294{311. Springer, 2003.

38- G. Mori and J. Malik. *Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA*. In Proc. of CVPR03, pp. 134{144. IEEE, 2003.

39- Simard PY, Szeliski R, Benaloh J, Couvreur J, and Calinov I (2003). *Using Character Recognition and Segmentation to Tell Computers from Humans*. Intl. Conf. on Document Analysis and Recognition (ICDAR), IEEE Computer Society, pp. 418-423, 2003.

Anexo 1: Terminología

Glosario de términos

Internauta: Usuario de una red informática de comunicación internacional.

Automatizado: Convertir en automático un determinado proceso o sistema.

Script: Se trata de un archivo de órdenes, un programa, generalmente simple, que por lo general se almacena en un archivo de texto plano.

Servidor web: programa informático que procesa una aplicación del lado del servidor. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web.

Ataque de aprendizaje: Ataque informático basado en la enseñanza de un determinado comportamiento o hecho a una máquina.

Ataque de canal lateral: Ataque informático basado en la existencia de fugas de información no deseada por un lado del canal.

Interfaz: Se utiliza para nombrar la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo, dando una comunicación entre distintos niveles.

Drag and Drop: Término muy utilizado y conocido en inglés, que en castellano quiere decir "arrastrar y soltar".

Algoritmo: conjunto de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos.

Abreviaturas

URL: Uniform **R**esource **L**ocator (localizador uniforme de recursos). Secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización.

API: **A**pplication **P**rogramming **I**nterface (interfaz de aplicaciones de Internet). Conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a algunos servicios desde los procesos y representa un método para conseguir abstracción en la programación.

IP: Internet **P**rotocol (protocolo de Internet). Estándar que se emplea para el envío y recepción de información mediante una red que reúne paquetes conmutados.

IA: Inteligencia **A**rtificial.

AJAX: Asynchronous **J**avaScript **A**nd **X**ML (JavaScript asíncrono y XML). Técnica de desarrollo web para crear aplicaciones interactivas.

Anexo 2: Presupuesto del Proyecto

En este apartado se comentarán todos los detalles económicos sobre la realización de este trabajo. Se explicarán y detallarán los costes tanto de material como de personal y otras valías adicionales a tener en cuenta.

Para calcular dichos costes se asumirá que el proyecto ha sido realizado por una empresa propia.

Para el coste final del trabajo se tendrá en cuenta el total de gastos, los impuestos y cierto porcentaje de beneficio y de riesgo.

Hay que tener en cuenta para el desarrollo de este apartado, que la unidad monetaria que utilizaremos a lo largo de todo el punto será el Euro, moneda oficial de la Unión Europea, y el redondeo será siempre a dos decimales.

Costes de personal

En este apartado veremos los costes relacionados con las personas que participan en el proyecto. Dentro del personal incluido para su desarrollo, estaría Patricia Román, como autor principal, y Arturo Ribagorda, tutor del proyecto.

El coste de la ayuda ofrecida por el tutor del proyecto no será analizado en el trabajo.

En la siguiente tabla podemos ver los detalles del personal:

Personal	Categoría	Euros/hora	Dedicación (horas)	Coste bruto (euros)
Patricia Román	Ingeniero Junior	30	400	12.000€

Tabla 2: Costes de personal

El coste del personal sería una cifra de **12.000€** brutos.

Además, debemos tener en cuenta el tipo de cotización al sistema de Seguridad Social referente al año 2013. En la siguiente tabla vemos más detalles sobre la parte que debe asumir la empresa y la parte que asume cada trabajador:

Contingencias	Empresa	Trabajadores	Total
Comunes	23,60	4,70	28,30
Horas Extraordinarias Fuerza Mayor	12,00	2,00	14,00
Resto Horas Extraordinarias	23,60	4,70	28,30

Tabla 3: Tipos de cotización 2013

En la siguiente tabla vemos las bases de cotización según la categoría del personal encargado del proyecto:

Grupo de Cotización	Categorías Profesionales	Bases mínimas euros/mes	Bases máximas euros /mes
1	Ingenieros y Licenciados. Personal de alta dirección no incluido en el artículo 1.3.c) del Estatuto de los Trabajadores	1.051,50	3.425,70
2	Ingenieros Técnicos, Peritos y Ayudantes Titulados	872,10	3.425,70

Tabla 4: Grupo de cotización

Con esta información calculamos el coste total y real del personal:

Personal	Base máxima	Base cotizada	Tipo	Cuota
Patricia Román	3.425,70	3.425,70	23,60	808,46

Tabla 5: Cálculo de cuotas

Personal	Coste bruto	Cuota	Coste total
Patricia Román	12.000€	808,46	12.808,46€

Tabla 6: Coste total por empleado

El coste total del personal para desarrollar el proyecto sería de: **12.808,46€.**

Costes de material

Ahora calcularemos los costes del material empleado para la realización del proyecto. Para calcular dichos costes tendremos que tener muy en cuenta la amortización de los materiales empleados.

En la siguiente tabla veremos el material hardware empleado:

Material	Coste/unidad	Unidad	Periodo de amortización (meses)	Duración del proyecto (meses)	Coste asociado
Ordenador de sobremesa HP Pavilion	600€	1	48	5	62,50€
Impresora HP Deskjet 840C	130€	1	36	5	18,06€

Tabla 7: Costes HW

No se han tenido en cuenta los gastos de papelería ya que se considera un gasto mínimo y poco relevante.

Coste total: 62,50+18,05=**80,56€.**

Además, también hay costes de software, que detallamos en la siguiente tabla:

Material	Coste/unidad	Uso (porcentaje)	Periodo de amortización (meses)	Duración del proyecto (meses)	Coste asociado
Microsoft 2010	119€	100%	12	5	49,58€

Tabla 8: Costes SW

También tenemos que tener en cuenta otros costes indirectos, que vemos a continuación:

Material	Coste/mes	Duración del proyecto (meses)	Coste asociado
Conexión a internet	50	5	225
Luz	40	5	200

Tabla 9: Costes indirectos

Esto hace un total de **425€**.

Viajes y dietas

En este punto se tendrán en cuenta los desplazamientos necesarios para la realización del proyecto y las comidas realizadas debido a dichos desplazamientos.

Estos desplazamientos y dietas constituyen los días de tutorías.

Se ha hecho un cálculo estimado de lo que costarían ambas partes, siendo el trayecto en trasporte público a la universidad de 7€ ida y vuelta, y las dietas serían 10€ cada una.

Material	Coste	Uso (días)	Coste total
Viaje	7€/viaje	8	56€
Dietas	10€/dieta	8	80€

Tabla 10: Viajes y dietas

La cifra de viajes y dietas asciende a un total de **136€**.

Costes totales

En la siguiente tabla vemos el coste total de proyecto, incluyendo todos los puntos anteriores.

Descripción	Coste
Personal	12.808,46
Costes de HW	80,56€
Costes de SW	49,58€
Costes indirectos	425€
Viajes y dietas	136€
Total	**13.499,60€**

Tabla 11: Costes totales

Además, a este trabajo tenemos que añadirle el beneficio que queremos obtener y un porcentaje de riesgos. Ambos costes los explicamos en la siguiente tabla:

Descripción	Coste	Porcentaje	Total
Beneficio	**13.499,60**	30%	4.049,88
Riesgos	**13.499,60**	20%	2.699,92

Tabla 12: Beneficio y riesgos

Cantidad total

Cantidad final sin IVA:

Descripción	Coste
Proyecto	13.499,60€
Beneficio	4.049,88
Riesgos	2.699,92
Total	20.249,40

Tabla 13: Total sin IVA

Si a esta cifra de 20.249,40 euros le añadimos el 21% de IVA reglamentario (4.252,37€), tenemos una cantidad total de **24.501,77 euros** (veinticuatro mil quinientos un euros con setenta y siete céntimos) de coste para el desarrollo de este proyecto.