International Journal of Computer Trends and Technology (IJCTT) - volume4 Issue5–May 2013

Reviewing Effectiveness of CAPTCHA

Ms. Priyanka, Ms.Harleen Kaur, Mr.Dileep Kumar Kushwaha
Department of MCA, M.T.U.NOIDA,
BBDIT, Ghaziabad, India

Jamia Hamdard
New Delhi, India

Abstract:

The massive use of the Internet and WWW by the government and commercial organizations for automation of work has resulted in extensive proliferation of online computing. To avoid online forms being filled by bots or malicious computers the form designers usually use a cryptic image known as CAPTHA (Completely Automated Public Turing test to Tell Computers and Humans Apart) [CAPTCHA] which is human readable and the form can be submitted only when the user has decrypted the captcha and filled in the text represented by that captcha.
This type of visual and textual verification comes at a huge price to users who are blind, visually impaired or dyslexic. In many cases, these systems make it impossible for users with certain disabilities to create accounts, write comments, or make purchases on these sites, that is, CAPTCHAs fail to properly recognize users with disabilities as human.

Also like every security system that has preceded it, this system can be defeated by those who benefit most from doing so. For example, spammers can pay a programmer to aggregate these images and feed them one by one to a human operator, who could easily verify hundreds of them each hour. The efficacy of visual verification systems is low, and their usefulness is nullified once they are commonly exploited.

Keywords : CAPTCHA, Security, Online Form Processing,effectiness of captcha, captcha for disable human.

Introduction:

CAPTCHA(Completely automated public turing test to tell computer and human apart Captcha is widely used over the world wide web to prevent the automated system(like bot) to scrape a data from web sites.

A CAPTCHA is a program that can generate and tests the human can pass but computer programs cannot. A CAPTCHA is basically just an implementation of function where it is easy to take input and compute the result. A CAPTCHA can be thought of in a simple terms as “ARE YOU A HUMAN?” test.

Initially it is a images which has some words or letters, embedded in it.

It is a challenge response test used to ensure that the challenge is generated by a human not by a computer. Users are asked to read some distorted characters (or some special images) and type the string in order to ensure that the user is human. Automation is real problem for web application, automate attacks can exploits public web services for several purposes-just like placing thousands of message on blogs, spam comments, forums, guest books and wikies, linking to forged sites for identity theft(Phishing), promoting a product, flooding a site with useless comment. As a newspaper printed out, 200 billion spams are sent every day. The spammers are using third parties to solve CAPTCHA called crowd sourcing.

Captcha is a challenge based response test used to ensure that the response is generated by a person not by computer. Users are asked to read and type a string of distorted characters in order to ensure that the user is human not a computer trying to access a website or account.

CAPTCHA provides us a secure web environment but sometimes it slow down the system where faster processing is more important than security. CAPTCHA is used where securities are more important, but we need to bypass the captcha where execution is more important. For example, a form being filled by a visually impaired person or an IRCTC agent performing many reservations, then we need to bypass the CAPTCHA.

Examples:

Fig. 1

In this paper we present a review of the effectiveness of CAPTCHA in providing online security.

Circumvention of CAPTCHA

CAPTCHA are so easy to be automatically detected and fed to the reader that they can hardly be trusted anymore. Presented here is a way to automatically detect a visual captcha by simple steps of image processing:

Automatic Detection of Captcha by Computer character recognition

A number of research projects have attempted to beat visual CAPTCHAs by creating programs that contain the following functionality:

1. Pre-processing: Removal of background clutter and noise
2. Segmentation: Splitting the image into regions which each contain a single character
3. Classification: Identifying the character in each region

Steps 1 and 3 are easy tasks for computers.[6] The only step where humans still outperform computers is segmentation. If the background clutter consists of shapes similar to letter shapes, and the letters are connected by this clutter, the segmentation becomes nearly impossible with current software. Hence, an effective CAPTCHA should focus on the segmentation.

The steps involved in the proposed methods are as follows:

Step 1: Determine the average color intensity of the image.
Step 2: Mark all pixel as white or black. This is done to remove any background noise in captcha.
Step 3: In step two there may have been generated some gaps so we eliminate horizontal gaps.
Step 4: Eliminate vertical gaps.
Step 5: It might be that steps 3 & 4 lead to inadvertent removal of some valid portion of captcha so we attempt to fill the region between two consecutive black regions in the image produced after step 4.
Step 6: Repeat the step 5.
Step 7: There may be false positives produced in step 6 (by filling in regions) which need to be cleared so we clear any false positives.
Step 8: The image produced in step 7 has characters distinctly visible but not aligned. So we find and align the characters.
Step 9: Feed this image produced in step 8 with aligned characters to optical character recognition software.
Step 10: Write the output of OCR software to a text file.
So we see that using simple image filtering loops based on observation of how a human being would approach the image and some existing OCR software the CAPTCHA has been detected and converted to text.

Other ways to circumvent captchas
There are several approaches available to defeating CAPTCHAs:

- exploiting bugs in the implementation that allow the attacker to completely bypass the CAPTCHA
- improving character recognition software
- using cheap human labor to process the tests (see below)

Insecure implementation

Like any security system, design flaws in a system implementation can prevent the theoretical security from being realized. Many CAPTCHA implementations, especially those which have not been designed and reviewed by experts in the fields of security, are prone to common attacks.

Some CAPTCHA protection systems can be bypassed without using OCR simply by re-using the session ID of a known test image. A correctly designed CAPTCHA does not allow multiple solution attempts at the same test, which would allow unlimited

reuse of a correct solution, or a second guess after an incorrect OCR attempt.[5] Other CAPTCHA implementations use a hash (such as an MD5 hash) of the solution as a key passed to the client to validate the answer. Further, the hash could assist an OCR based attempt. A more secure scheme would use an HMAC(Hash-based message authentication code). Another example is directly provide answer in the code such as showing four pictures to let user pickup the correct one, a spam bot can always guess the first picture to gain 25% success rate in this case. Finally, some implementations use only a small fixed pool of CAPTCHA images. Eventually, when enough image solutions have been collected by an attacker over a period of time, the test can be broken by simply looking up solutions in a table, based on a hash of the challenge image.

In February 2008, it was reported that spammers had achieved a success rate of 30% to 35%, using a bot to respond to CAPTCHAs for Microsoft's Live Mail service[7] and a success rate of 20% against Google's Gmail CAPTCHA.[8] A Newcastle University research team has defeated the segmentation part of Microsoft's CAPTCHA with a 90% success rate, and reported that this could lead to a complete crack with a greater than 60% rate.[9]

Human solvers

CAPTCHA is vulnerable to a relay attack that uses humans to solve the puzzles. One approach involves relaying the puzzles to a group of human operators who can solve CAPTCHAs. In this scheme, a computer fills out a form and when it reaches a CAPTCHA, it gives the CAPTCHA to the human operator to solve.

Spammers pay about $0.80 to $1.20 for each 1,000 solved CAPTCHAs to companies employing human solvers in Bangladesh, China, India, and many other developing nations.[22] Other sources cite a cost as low as $0.50 for each 1,000 solved.[10]

Another approach involves copying the CAPTCHA images and using them as CAPTCHAs for a high-traffic site owned by the attacker. With enough traffic, the attacker can get a solution to the CAPTCHA puzzle in time to relay it back to the target site.[24] In October 2007, a piece of malware appeared in the wild which enticed users to solve CAPTCHAs in order to see progressively further into a series of striptease images.[11][12] A more recent view is that this is unlikely to work due to unavailability of high-traffic sites and competition by similar sites.[13]

These methods have been used by spammers to set up thousands of accounts on free email services such as Gmail and Yahoo![14] Since Gmail and Yahoo! are unlikely to be blacklisted by anti-spam systems, spam sent through these compromised accounts is less likely to be blocked.

In 2010, encouraged by Ticketmaster, the U.S. Attorney in Newark, New Jersey won a grand jury indictment against Wiseguy Tickets, Inc. for purchasing tickets in bulk by circumventing CAPTCHA mechanisms.[15] Among its 43 findings, the grand jury found Wiseguy Tickets Inc defeated online ticket vendors' security mechanisms CAPTCHA.[16]

SPAM BOTS
API that Bypass Captcha

API	Description
Antigate	CAPTCHA Decoding Service
Bypass CAPTCHA	CAPTCHA Bypass Service
CaptChair	Image-based captcha service
Confident CAPTCHA	Image-based captcha service
Death By CAPTCHA	CAPTCHA Bypass Service
Ericsson Captcha	Security and Advertisement Service
I'm Human	Visual CAPTCHA service
ImageDecoders	CAPTCHA bypass service
Keypic	Image-based spam prevention service

Photo Captcha	Image based captcha service
RainCaptcha	Free CAPTCHA service
reCAPTCHA	Security and Book Digitalization Service

SUBSTITUTES to CAPTCHA

Voice CAPTCHA
In this method, the user is given the option to hear the verification code apart from seeing it in the screen. The user needs to activate a button which would play the audio file containing the code. The user can then enter the code on the provided field and proceed to the next processes.

This option is appropriate for blind Internet users and those with visual impairments. However, deaf people and individuals with cognitive disabilities may face problems in using it.

Math Questions
Instead of a graphical code, the user has to answer a simple math question. A common example would be "2 + 2". The user then has to enter the answer and if it is correct, the system loads the succeeding page.

This option is good for most people. Developers should nonetheless ensure that the math questions are all basic ones. Math questions that are complex may cause difficulties for persons with cognitive disabilities.

Simple Questions
Apart from math questions, developers have tried using simple questions. An example of this is a question such as "What is the second letter of the English alphabet?"

Simple questions generally work well for Internet users. But concerns may still arise among certain people. For instance, the above example may make it difficult for non-English speakers to quickly provide the answer.

Easy Tasks
Apart from questions, easy tasks can also be used in place of CAPTCHA. For example, a registration form can contain a checkbox that is checked by default. And the checkbox has a label such as "Uncheck the box if you are a human". Developers who choose this option should make sure that the tasks are truly easy to follow.

Hidden Fields
This option builds on the idea that robots enter information in a field regardless of whether or not it is visible. Developers use CSS to hide a particular field, rendering it invisible to humans. Here, the system checks if a hidden field has data. If it does, it is most likely filled in by a robot. Otherwise, a human has entered information in the form.

The main problem of this option is that it may be confusing for someone who has deactivated CSS.

Verification via SMS
In this option, the site asks for the user's mobile number. After the user has provided this information, the site will send an SMS containing the verification code to the mobile phone. The user then can enter the code in the specified field.

The main problem of this option is that not all persons have mobile phones. In addition, a blind person, for example, may have a mobile phone but it doesn't have the needed screen reading software. Another issue is that people may use the page from another country.

Confirmation Page
Here, the user enters information in one page and activates a Submit button to proceed to the next page. The next page contains the information the user previously entered, and a Confirm button. This prevents robots from successfully entering information in the system because they normally focus only on the page containing fields.

Developers who choose this option should see to it that the page clearly explains to the users what would happen. Otherwise, some users might think that after clicking on the Submit button, they can leave the site.

Determining Time taken to fill the form
A human normally takes at least half a minute to fill out a form with three to five fields. On the other hand, most robots complete a form automatically, so it doesn't take more than a few seconds.

The system can determine if the form was filled out in a very short amount of time (e.g. ten seconds for three fields). Through the acquired information, the system can determine if the form is filled out by a robot or a human.

Determine if a Java Script is Loaded
This option is for pages running a java script. When the java script has been executed, one can be certain that a human is using the page. Developers can create a way to check this activity. The main problem of this option takes place when the user has deactivated Java scripts.

Use human intervention
Here, the user is asked to send a short request email to the person handling the website. That person can then verify if the sender of the email is a human. Although this option can be very effective, registration may take a longer time to complete.

Verification through Pictures
The user is presented with a set of pictures; say a cat, a dog, and a bird. The page then tells the user to click on one of the pictures, the dog, for instance.

To help blind users answer this test, developers can include alternate descriptions in the pictures. In this case, the dog picture says "dog", but robots would find it hard to know what the page wants the user to select.

Identifying Sound
The user clicks on a button to play a sound. The page presents a set of buttons or links containing names of sounds. The user has to click the button whose caption best describes the sound.

The main issue of this option is that it would exclude deaf people from answering the test.

Devising a test for effectiveness of CAPTCHA
1) Test should be administered where the human and the server are remote over the network.
2) Test should be simple for humans to pass. Humans should fail less than 0.1% on the first attempt.
3) Test should be solvable by humans in less than a several seconds.
4) Test should only be solvable by the human to which it was presented.
5) Test should be hard for computer to pass correctly guessing the answer should be less than 1 in 1,000,000, even after 24-hours of analysis.
6) Knowledge of previous test questions, answers, results, or combination thereof should not impact the predictability of following tests.
7) Test should not discriminate against humans with visual or hearing impairments.
8) Test should not possess a geographic, cultural, or language bias.

Instances of CAPTCHA Breaches

1. **Busting Google's Captcha**

Spammers in these attacks managed to create bots that are capable of signing up and creating random Gmail accounts for spamming purposes.

Signing up for an account with Google allows access to its wide portfolio of services. Second, Google's domains are unlikely to be blacklisted. Third, they are free to sign up. And fourth, it may be hard to keep track of them as millions of users worldwide are using various Google services on a regular basis.

2. Attack on Microsoft Captcha
A simple attack has achieved a segmentation success rate of higher than 90% against the captcha developed and used by Microsoft websites. It took on average ~80 ms for the attack to completely segment a challenge on a desktop computer with a 1.86 GHz Intel Core 2 CPU and 2 GB RAM. This Microsoft scheme can be broken with an overall (segmentation and then recognition) success rate of more than 60%. On the contrary, its design goal was that "automatic scripts should not be more successful than 1 in 10,000" attempts (i.e. a success rate of 0.01%).

Conclusion
Although we have seen that CAPTCHA are very easy to circumvent with some programming and there are a number of SPAMBOTS and API working around the Internet to counteract the security provided by CAPTCHA yet almost 90% of websites continue to use CAPTCHA in some form and this will continue till next breakthrough in online security is achieved.

Captchas are vulnerable to attacks. A so called good captcha scheme can broken with an overall (segmentation and then recognition) success rate of more than 60%. Therefore, we find that captchas provide only a week security. Even if segmentation resistance is a sound principle for designing secure text-based CAPTCHAs, it is critical to make sure that a design is not vulnerable to any known (and ideally unknown) segmentation method. Designing CAPTCHAs that exhibit both good robustness and usability is much harder that it might appear to be because current collective understanding of this topic is small and the requirements, tools and methodologies for assessing captcha designs are almost nill.

Acknowledgement:
This research paper is made possible through the help and support from everyone, Including: Teachers, family, friends, and in essence, all sentient beings.

Especially, please allow me to dedicate my acknowledgment of gratitude toward the following significant advisors and contributors:
First and foremost, I would like to thank Dr. Harleen Kaur for his most support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper.
Second, I would like to thank as well as all the other faculty members who have taught me about image processing and captcha over the past three years of my pursuit of the master degree.
Finally, I sincerely thank to my friends, who provide the advice. The product of this research paper would not be possible without all of them.

References:

[1] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: using hard AI problems for security. In Eurocrypt, 2003.
[2] Luis von Ahn, Manuel Blum, Nicholas J. Hopper and John Langford. TheCAPTCHA Web Page: http://www.captcha.net. 2000.
[3] Luis von Ahn, Manuel Blum and John Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI. To appear in Commu-nications of the ACM.
[4] Mihir Bellare, Russell Impagliazzo and Moni Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In 38th IEEE Symposium on Foundations of Computer Science (FOCS' 97),
[5] "Breaking CAPTCHAs Without Using OCR". *Howard Yeend (pureMango.co.uk)*. 2005. Retrieved 2006-08-22.
[6] Kumar Chellapilla, Kevin Larson, Patrice Simard, Mary Czerwinski (2005). *Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs)* (PDF). Microsoft Research. Archived from the original on 2006-06-13. Retrieved 2006-08-02.
[7] Gregg Keizer, "Spammers' bot cracks Microsoft's CAPTCHA: Bot beats Windows Live Mail's registration test 30% to 35% of the time, says Websense", *Computerworld"', February 7, 2008*
[8]Prasad, Sumeet (2008-02-22). "Google's CAPTCHA busted in recent spammer tactics". Websense. Archived from the original on 2008-08-22. Retrieved 2008-12-21.
[9]Jeff Yan; Ahmad Salah El Ahmad (April 13, 2008). *A Low-cost Attack on a Microsoft CAPTCHA* (PDF). School of Computing Science, Newcastle University, UK. Retrieved 2008-12-21.
[10] M. Motoyama, K. Levchenko, C. Kanich, D. McCoy,G. M. Voelker, and S. Savage. "Re: CAPTCHAs:understanding CAPTCHA-solving services in an economic context". University of California, San Diego. Retrieved 17 March 2011.
[11] Robertson, Jordan (2007-11-01). "Scams Use Striptease to Break Web Traps". San Jose, California. Associated Press. Archived from the original on 2007-11-06.
[12] Vaas, Lisa (2007-11-01). "Striptease Used to Recruit Help in Cracking Sites". PC Magazine. Retrieved 2008-12-
[13] "Captcha.net". Captcha.net. Retrieved 2011-03-22.
[14] "Spam filtering services throttle Gmail to fight spammers". 2008-04-10. Retrieved 2008-04-10.
[15] Zetter, Kim (March 1, 2010). "Wiseguys Indicted in $25 Million Online Ticket Ring". *Wired.com*. Retrieved 2012-01-02.
[16] "UNITED STATES of AMERICA vs KENNETH LOWSON, KRISTOFER KIRSCH, LOEL STEVENSON". *Federal Indictment*. February 23, 2010. Retrieved 2012-01-02.