Web Security using CAPTCHA: A Review

Chetan Chopra
Student, CSE
KIIT College of Engineering
Gurgaon, India
Chetan.2jan@gmail.com

Garima Harish
Student, Microbiology
Bhaskaracharya College of Applied Sciences
New Delhi, India
Garimaharish@yahoo.com

***Abstract*—CAPTCHA is used in providing the security to the websites. It is one of the best methods being implemented to provide the distinction between humans and the bots. In this paper, we have provided the information about CAPTCHA and various types of CAPTCHAS that are available to secure the websites from bots.**

***Index Terms*— CAPTCHA, Security, ReCaptcha.**

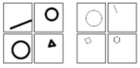
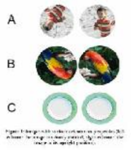
INTRODUCTION

CAPTCHA stands for Completely Automated Public Turing test to tell Computer and Humans Apart. This program is used to protect websites from bots by putting forward tests in which user is asked to read a distorted image, humans can pass but bots cannot. It basically exploits ability of human mind to perceive visual data and inability of computer to do so. The term CAPTCHA was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas Hopper and John Langford of Carnegie Mellon University [1]. CAPTCHA is very much similar to the Turing Test but the difference is that now the judge is a computer. It was first used by "ALTAVISTA" in 1997 [2]. The various types of CAPTCHA broadly include Textual CAPTCHA, Graphical CAPTCHA and Audio CAPTCHA.

To create a CAPTCHA programmer must look at different ways humans and machines process information. Machines follow sets of instructions. If something falls outside the realm of those instructions, the machines aren't able to compensate. One way to create a CAPTCHA is to pre-determine the images and the solutions it will use. This approach requires a database that includes all the CAPTCHA solutions, which can compromise the reliability of the test. Other CAPTCHA applications create random strings of letters and numbers.

In order to break these CAPTCHAs programmer could approach the problem in phases. He or she would need to write an algorithm a set of instructions that directs a machine to follow a certain series of steps. Greg Mori and Jitendra Malik published a paper detailing their approach to cracking the Gimpy version of CAPTCHA and Jennifer Tam, Jiri Simsa, Sean Hyde, and Luis von Ahn published a paper detailing their approach to break audio CAPTCHAs. CAPTCHA find its application in to combat email worms and spams, online polling, prevent dictionary attack, spam comments on blogs, to protect email addresses from scrappers and to protect website registrations [4].With many advantages of CAPTCHA, the biggest disadvantage is the evolution of Artificial Intelligence.

Table 1: Types of CAPTCHAS

Types Of CAPTCHAS		
Text CAPTCHA	**Visual CAPTCHA**	**Audio CAPTCHA**
Gimpy 	Bongo 	
Ez- Gimpy 		
Baffle Text 	PIX 	
MSN CAPTCHA 		

TYPES OF CAPTCHA

The various types of CAPTCHA include:

Text CAPTCHA: This is CAPTCHA works by putting a question or more than one words which humans can recognize easily but bots cannot. Simple questions like, what is second letter in the word SUNDAY? Not only questions, this category also include distorted text which user is asked to identify [5]. Distorted text can be of the following types:

Gimpy: This CAPTCHA was built by CMU in collaboration with YAHOO! for its messenger. It randomly takes up some dictionary words and shows two of them in overlapping form. Humans can read these words with ease but bots cannot.

Ez-Gimpy: Simplified version of Gimpy which randomly pics up dictionary words and add distortion to them.

Baffle Text: This CAPTCHA was developed by Henry Baird at University of California at Berkeley. It picks up random alphabets and makes a nonsense yet pronounceable word and later distortion is added. This CAPTCHA overcomes the drawback of Gimpy CAPTCH which uses dictionary words. Clever bots can break this CAPTCHA by finding matching word from dictionary. [11]

MSN CAPTCHA: Microsoft uses a different CAPTCHA for MSN popularly known as MSN Passport CAPTCHAs. They use eight characters (upper case) and digits. Foreground is dark blue, and background is grey. Warping is used to distort the characters, to produce a ripple effect, which makes computer recognition very difficult.

Graphic CAPTCHA: These are visual puzzles generated and graded by computers, but computer is unable to solve it itself. These puzzles involve pictures and objects that have sort of similarities, which can be easily judged by humans.

Bongo: BONGO is named after M.M. Bongard [6]. This program asks the user to solve a visual pattern recognition problem. In particular, Bongo displays two series of blocks, the left and the right series. The blocks in the left series differ from those in the right, and the user must find the characteristic that sets the two series apart.

PIX: This program has a large database of labelled images. The program picks an object at random, finds six images of that object from its database, presents them to the user and then asks the question “what are these pictures of?” Current computer programs should not be able to answer this question.

Audio CAPTCHA: Nancy Chan of the City University in Hong Kong was the first to implement this sound based system [5]. The program picks a word or a sequence of numbers at random, converts it into a sound clip and distorts the sound clip. This clip is presented to the user and asked to enter its contents. This CAPTCHA is based on the difference in ability between humans and computers in recognizing spoken language.

CONSTRUCTING CAPTCHAS

The first step to create a CAPTCHA is to look at different ways humans and machines process information. Machines follow sets of instructions. If something falls outside the realm of those instructions, the machines aren’t able to compensate. A CAPTCHA designer has to take this into account when creating a test. For example, it’s easy to build a program that looks at **metadata** – the information on the Web that’s invisible to humans but machines can read. If you create a visual CAPTCHA and the images’ metadata includes the solution, your CAPTCHA will be broken in no time. Similarly, it’s unwise to build a CAPTCHA that doesn’t distort letters and numbers in some way. An undistorted series of characters isn’t very secure. Many computer programs can scan an image and recognize simple shapes like letters and numbers. One way to create a CAPTCHA is to pre-determine the images and solutions it will use. This approach requires a database that includes all the CAPTCHA solutions. If a spammer managed to find a list of all CAPTCHA solutions, he or she could create an application that bombards the CAPTCHA with every possible answer in a **brute-force** attack. The database would need more than 10,000 possible CAPTCHAs to meet the qualifications of a good CAPTCHA. Other CAPTCHA applications create random strings of letters and numbers. Using randomization eliminates the possibility of a brute-force attack — the odds of a bot entering the correct series of random letters are very low. Some CAPTCHAs rely on **pattern recognition** and **extrapolation**.

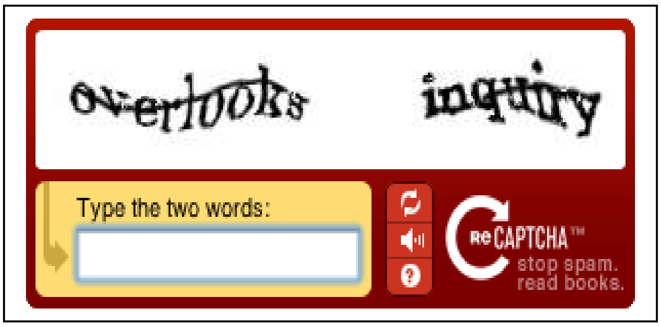


Figure 1: ReCAPTCHA

CAPTCHA builder, which will provide the authentication services remotely. Most such services are free. The original text (pre-altered) is persisted somewhere, as this is the correct answer to the question. There are different ways to persist the answer, as a server-side session variable, cookie, file, or database entry. The generated CAPTCHA is presented to the user, who is prompted to answer it. The back-end script checks the answer supplied by the user by comparing it with the persisted (correct) answer. If the value is empty or incorrect, we go back to step 1 and a new CAPTCHA is generated. Users should never get a second shot at answering the same CAPTCHA. If the answer supplied by the user is correct, the form post is successful and processing can continue. If applicable, the generated CAPTCHA image is deleted. [10]

DIGITIZATION OF BOOKS AND RECAPTCHA

To archive human knowledge and to make information more accessible to the world, multiple projects are currently digitizing physical books that were written before the computer age. The book pages are being photographically scanned, and then transformed into text using "Optical Character Recognition" (OCR). The transformation into text is useful because scanning a book produces images, which are difficult to store on small devices, expensive to download, and cannot be searched. The problem is that OCR is not perfect. ReCAPTCHA improves the process of digitizing books by sending words that cannot be read by computers to the Web in the form of CAPTCHAs for humans to decipher. More specifically, each word that cannot be read correctly by OCR is placed on an image and used as a CAPTCHA. This is possible because most OCR programs alert you when a word cannot be read correctly. Each new word that cannot be read correctly by OCR is given to a user in conjunction with another word for which the answer is already known. The user is then asked to read both words. If they solve the one for which the answer is known, the system assumes their answer is correct for the new one. The system then gives the new image to a number of other people to determine, with higher confidence, whether the original answer was correct.

GUIDELINES OF CAPTCHA

If a site has to be protected from abuse, it should have CAPTCHA and following are the guidelines which are strongly recommended for a CAPTCHA code:

Accessibility: CAPTCHAs must be accessible. CAPTCHAs which involve only visual-perception tasks prevent visually impaired users from accessing the protected resource. Such CAPTCHAs may make a site incompatible with Section 508 in the United States [1]. Hence, the site should provide an audio or sound CAPTCHA.

Security of images: CAPTCHA images of text should be properly distorted before being presenting to the user. Many sites use undistorted text, or text with only minor distortions as CAPTCHAs, such texts are highly susceptible to bot attack. Over distortion should also be avoided. 3. Script Security. Building a secure CAPTCHA code is not easy. In addition to making the images unreadable by computers, the system should ensure that there are no easy ways around it at the script level. Common examples of insecurities in this respect include:

Systems that pass the answer to the CAPTCHA in plain text as part of the web form.

Systems where a solution to the same CAPTCHA can be used multiple times (this makes the CAPTCHA vulnerable to so-called "replay attacks"). Most CAPTCHA scripts found freely on the Web are vulnerable to these types of attacks [1].

Security Even after Wide-Spread Adoption: There are various "CAPTCHAs" that would be insecure if a significant number of sites started using them. Since a parser could easily be written that would allow bots to bypass this test, such "CAPTCHAs" rely on the fact that few sites use them, and thus that a bot author has no incentive to program their bot to solve that challenge. True CAPTCHAs should be secure even after a significant number of websites adopt them.

APPLICATIONS

CAPTCHA basically deals with security and has several applications. Few of its applications are as follow [7]:

Online polls: Online polls need a lot of security as bots can wreak havoc to any unprotected online poll. They might create a large number of votes which would then falsely represent the poll winner. This problem came into light when http://www.slashdot.org released an online poll asking which was the best graduate school in computer science. So, in order to protect them from being accessed by bots, CAPTCHA is needed.

Preventing comment spams: In order to improve search engine ranking, many bloggers employ programs that submit bogus comments. To overcome this issue, CAPTCHA is used. Only humans can enter their comments that too without signing in.

Protecting website registrations: Companies like Google offer its users a free email service. These services are protected with a CAPTCHA in order to prevent abuse by automated scripts.

Preventing Dictionary Attacks: To prevent a computer from being able to literate itself through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.

Preventing Dictionary Attacks: To prevent a computer from being able to literate itself through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.

E-Ticketing: CAPTCHA help prevent ticket scalpers from bombarding the service with massive ticket purchases for big events. Without some sort of filter, it's possible for a scalper to use a bot to place hundreds or thousands of ticket orders in a matter of seconds. Legitimate customers become victims as events sell out minutes after tickets become available. Scalpers then try to sell the tickets above face value. While CAPTCHA applications don't prevent scalping; they do make it more difficult to scalp tickets on a large scale.

Worms and Spam: To avoid spam emails, users are asked to solve a CAPTCHA to make sure it’s a human sending that particular mail.

As a tool to verify digitized books: This is a way of increasing the value of CAPTCHA as an application. An application called reCAPTCHA harnesses users responses in CAPTCHA fields to verify the contents of a scanned piece of paper. Because computers aren’t always able to identify words from a digital scan, humans have to verify what a printed page says. Then it’s possible for search engines to search and index the contents of a scanned document. This is how it works: The application already recognizes one of the words. If the visitor types that word into a field correctly, the application assumes the second word the user types is also correct. That second word goes into a pool of words that the application will present to other users. As each user types in a word, the application compares the word to the original answer. Eventually, the application receives enough responses to verify the word with a high degree of certainty. That word can then go into the verified pool.

BREAKING A CAPTCHA

Breaking a CAPTCHA isn’t about solving it, rather making the computer process it like how human perceives it. For each type of CAPTCHA there exist a way to break it. Like for example, The **Gimpy** CAPTCHA displays 10 English words with warped fonts across an irregular background. The CAPTCHA arranges the words in pairs and the words of each pair overlap one another and in EZ-Gimpy, the CATPCHA used by Yahoo! (shown in the figure above), the user is presented with an image of a single word. This image has been distorted, and a cluttered, textured background has been added. The distortion and clutter is sufficient to confuse current OCR (optical character recognition) software. Greg Mori and Jitendra Malik published a paper detailing their approach to cracking the Gimpy version of CAPTCHA. [9] One thing that helped them was that the Gimpy approach uses actual words rather than random strings of letters and numbers. They designed first an algorithm to find words in images works from the bottom-up, starting with visual cues and incorporate lexical information later on. Second algorithm they designed was a holistic one that attempts to find entire words at once, instead of looking for letters. They also used the Gimpy's 500-word dictionary. [3] Most CAPTCHAs don't destroy the session when the correct phrase is entered. So by reusing the session id of a known CAPTCHA image, it is possible to automate requests to a CAPTCHA-protected page. So to crack CAPTCHA without using OCR the following manual steps should be carried out. Connect to CAPTCHA page and Record session ID and CAPTCHA plain text. Then Resend session ID and CAPTCHA plaintext any number of times, changing the user data. The other user data can change on each request. Then hundreds, if not thousands of requests can be automated, until the session expires, at which point we just repeat the manual steps and then reconnect with a new session ID and CAPTCHA text. Spammers often use social engineering to outwit gullible Web users to serve their purpose. Security firm, Trend Micro warns of a Trojan called TROJ_CAPTCHAR**,** which masquerades as a strip tease game. At each stage of the game, the user is asked to solve a CAPTCHA. [8] The result is relayed to a remote server where a malicious user is waiting for them. The strip-tease game is a ploy by spammers to identify and match solutions for ambiguous CAPTCHAs from legitimate sites, using the unsuspecting user as the decoder of the said images.

CONCLUSION

Internet has become a necessity in this modern world. And with increasing demand of internet services web security has become main area of concern for website developers and companies. Most problems are posed by bots, these computer programs are not identifiable because they can easily fill in online forms like humans and can misuse the services provided by the site. So, in order to combat these bots, concept of CAPTCHAs was introduced. Its success rate is very high because bots cannot solve them and hence the website remains protected. CAPTCHA should serve a dual purpose, it should be easy to solve for human users and at the same time must be a problem for bots.

In our study we came across types of CAPTCHAs, their applications and how an ideal CAPTCHA should be like. There is a need of more study and development of this tool as breaks for some of the CAPTCHAs have already been found out.

ACKNOWLEDGMENT

We express our sincere thanks to Mrs. Saruchi Gupta, lecturer, Department of CSE, KIIT College of Engineering, for guiding us right from the inception till the successful completion of our review paper.

REFERENCES

[1] http://www.captcha.net/

[2] http://www.slideshare.net/preetamkajalrout/captcha-7880840

[3] Greg Mori and Jitender Malik, Computer Science Division, "Breaking a visual CAPTCHA", University of California, Berkeley, CA 94720

[4] Luis von Ahn, Manuel Blum and John Langford. "Telling Humans and Computers Apart Automatically".

[5] Elie Bursztein, Matthieu Martin, John C. Mitchell. "Text-based CAPTCHA Strengths and Weaknesses".

[6] http://www.seminarpaper.com/2011/09/captcha-full-report.html

[7] Arvind Kumar, "A Seminar Report ON CAPTCHA".

[8] E. Bursztein and S. Bethard. "DeCaptcha: Breaking 75% of eBay audio CAPTCHAs". In Proceedings of the 3rd USENIX conference on Offensive technologies, page 8. USENIX Association, 2009.

[9] K Chellapilla and P Simard. "Using Machine Learning to Break Visual Human Interaction Proofs".

[10] K. Chellapilla, K. Larson, P.Y. Simard, and M. Czerwinski. "Computers beat humans at single character recognition in reading based human interaction proofs".

[11] Chew, Baird. Baffle text: "A Human Interactive".