International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 02 | Feb 2019 www.irjet.net p-ISSN: 2395-0072

Different Implemented Captchas and Breaking Methods

Shadi Khawandi[1], Anis Ismail[2], Firas Abdallah[3]

[1,2,3]*Faculty of Technology, Lebanese University, Lebanon*

Abstract - *CAPTCHA is almost a standard security technology, and has found widespread application in commercial websites. There are two types: labeling and image based CAPTCHAs. To date, almost all CAPTCHA designs are labeling based. Labeling based CAPTCHAs refer to those that make judgment based on whether the question "what is it?" has been correctly answered. Essentially in Artificial Intelligence (AI), this means judgment depends on whether the new label provided by the user side matches the label already known to the server. Labeling based CAPTCHA designs have some common weaknesses that can be taken advantage of attackers. First, the label set, i.e., the number of classes, is small and fixed. Due to deformation and noise in CAPTCHAs, the classes have to be further reduced to avoid confusion. Second, clean segmentation in current design, in particular character labeling based CAPTCHAs, is feasible. The state of the art of CAPTCHA design suggests that the robustness of character labeling schemes should rely on the difficulty of finding where the character is (segmentation), rather than which character it is (recognition). However, the shapes of alphabet letters and numbers have very limited geometry characteristics that can be used by humans to tell them yet are also easy to be indistinct. Image recognition CAPTCHAs faces many potential problems which have not been fully studied. It is difficult for a small site to acquire a large dictionary of images which an attacker does not have access to and without a means of automatically acquiring new labeled images, an image based challenge does not usually meet the definition of a CAPTCHA. They are either unusable or prone to attacks. In this paper, we present the different types of CAPTCHAs trying to defeat advanced computer programs or bots, discussing the limitations and drawbacks of each, then we will mention some methods of breaking CAPTCHA, OCR and non-OCR.*

Key Words: *CAPTCHAs, Labeling, Segmentation, Image recognition*

1. INTRODUCTION

With the development of the computer applications in different fields, internet has made a tremendous progress and become a special need in human life. It has applications in a wide range of daily affairs including trade, education, daily purchases and dialogues take place with the use of Internet. One of the common actions in the Internet web sites, especially commercial and administrative ones, is to fill out registration forms for certain purposes. Unfortunately, there are some programs which automatically fill out these forms with incorrect information to abuse the site, or automated programs which are usually written to generate spam.

Thus, differentiating between a user and machine over the internet has significant importance in the fields of internet security, artificial intelligence, and machine learning. Currently, HIPs takes the role of preventing robots from signing up for free online services (such as email accounts), abusing online polls, providing biased feedback, and spamming innocent users.

Completely Automated Public Turing test to tell Computers and Humans Apart is class of automated challenges used to differentiate between legitimate human users and computer programs or bots on the internet. Thus, it plays the same role of HIP.

Early internet hackers of the 1980s are often credited with the idea of obscuring text to foil content filtering devices. When trying to obscure sensitive data online, these hackers would use the simple technique of substituting numbers for text (e.g., I = 1, A = 4, etc.). The algorithms they used were extremely simple but would fool content filters and web crawlers. The technique was originally used to get around profanity filters which were installed on online commenting systems and forums. These filters were extremely simple and banned a predefined set of inappropriate words. The end result was something that a fellow human could read with a minor degree of difficulty but a computer could not. This concept eventually evolved into what is currently known as "l33tspeak" (elite speak).

With the recent massive increase in the amounts of spam (unsolicited, junk email) being delivered to inboxes around the world, people have been concerned with publishing their email address online. Spam bots can process thousands of web pages per hour, scanning for email addresses in clear text. Many people have resorted to attempting to fool spam bots by posting their email addresses in a human readable but unconventional form. For example, the text string "Hussein AT Hotmail DOT com" is not easily parsed or understood by spam bots which crawl for email addresses in conventional forms. However, humans can easily identify the intended email address. This solution is not perfect, as any text-based data that humans can easily read can also be easily read by a machine.

A simple set of regular expressions could easily match many predefined text masquerading patterns and successfully harvest the obscured email address. Thus, a need to protect data against automated machine processing arose.

In 1997 Andrei Broder, Chief Scientist of AltaVista, and his colleagues prevented automated machine from adding of URLs to their search engine. They developed a program that permitted human's entrance but not machine's entrance. In 2000, Bots were annoying genius chatter by advertising sites and elicit personal information. CMU researchers: Manual Blum, Luis A. von Ahn and John Langford coined the term "CAPTCHA" that was pointed to "capture", and used CAPTCHA in order to solve Yahoo's chat room problem. In 2001 Allison Coates, Henry S. Baird and Richard Fateman of UC Berkeley developed Pessimal Print: that is low–quality of printed text images used certain rate of distortion [1].

The notion of a machine imitating human intelligence was first addressed as early as 1950 by English mathematician and logician Alan Turing [2]. Acknowledged as the father of modern computing, Turing recognized that computers might eventually be able to imitate human thought in very convincing ways. Therefore, he suggested what is now known as the Turing test, where a human converses with a computer without seeing it. If the human is convinced by the computer's answers that it is human, then the machine passes the test and is deemed to have some level of human-like intelligence.

The idea of a reverse Turing test, where a computer attempts to differentiate between a human and a computer, arose during the late 1990s when computer programs began to imitate humans in order to misuse the resources of internet-based systems.

HIPs [3] are a slight modification of a reverse Turing test, where the challenge is administered by a machine and taken by a human. The burden is on the human participant to convince the machine that he is human. Furthermore, the challenge should not be solvable by any machine. Notice the paradox that this creates: the machine can automatically create, administer, and grade a test that it itself cannot pass. Tests developed to differentiate these programs from real humans took the form of what would come to be known as CAPTCHAs.

CAPTCHAs generate and grade tests that most humans can pass but current computer programs can't. Such tests, often called CAPTCHA challenges are based on hard, open artificial intelligence problems. To date, the most commonly used CAPTCHAs are text-based, in which the challenge appears as an image of distorted text that the user must decipher and retype. These schemes typically exploit the difficulty for state-of-the-art computer programs to recognize distorted text.

A good CAPTCHA must not only be human friendly but also robust enough to resist computer programs that attackers write to automatically pass CAPTCHA tests. However, designing CAPTCHAs that exhibit both good robustness and usability is much harder than it might seem. The current collective understanding of this topic is very limited, as suggested by the fact that many well-known schemes break.

Labeling based CAPTCHA designs have some common weaknesses that can be taken advantage of attackers. First, the label set, i.e., the number of classes, is small and fixed. (Note that labeling is essentially a classification procedure.) All labeling based CAPTCHA designs require that the label set should be known to the user. Users have to tell exactly "what is it?" since label matching is strict. That is the reason why the set of alphabet letters and numbers is a popular choice. Note that the number of classes for alphabet letters and numbers are very limited.

2. Introducing CAPTCHA

The term CAPTCHA, which stands for Completely Automated Turing Test to Tell Computers and Humans Apart, was coined in 2000 by a group of professors at Carnegie Mellon University. They were working on research for Yahoo developing the first of these programs when they first used the term.

It is a type of challenge-response test used in computing as an attempt to ensure that the response is not generated by a computer. The process usually involves one computer (a server) asking a user to complete a simple test which the computer is able to generate and grade. Because other computers are supposedly unable to solve the CAPTCHA, any user entering a correct solution is presumed to be human.

2.1 - The CAPTCHA

You have probably seen them, colorful images with distorted text in them at the bottom of Web registration forms. CAPTCHAs are used by Yahoo, Hotmail, PayPal and many other popular Web sites to prevent automated registrations, and they work because no computer program can currently read distorted text as well as humans can. What you probably do not know is that a CAPTCHA is something more than just an image with distorted text. It is a test, any test, that can be automatically generated, which most humans can pass, but that current computer programs cannot pass. Notice the paradox, a CAPTCHA is a program that can generate and grade tests that it cannot pass (much like some professors).

CAPTCHA stands for "Completely Automated Public Turing Test to Tell Computers and Humans Apart. [28]" It is also known as human interaction proofs (HIPs). The P for Public means that the code and the data used by a CAPTCHA should be publicly available. This is not an open source requirement, but a security guarantee, it should be difficult for someone to write a computer program that can pass the tests generated by a CAPTCHA even if they know exactly how the CAPTCHA works (the only hidden information is a small amount of randomness utilized to generate the tests).

The T for "Turing Test to Tell" is because CAPTCHAs are like Turing Tests. In the original Turing Test, a human judge was allowed to ask a series of questions to two players, one of which was a computer and the other a human. Both players pretended to be the human, and the judge had to distinguish between them. CAPTCHAs are similar to the Turing Test in that they distinguish humans from computers, but they differ in that the judge is now a computer. A CAPTCHA is an Automated Turing Test. We avoid using the term Reverse Turing Test (or even worse, RTT) because it can be misleading. Reverse Turing Test has been used to refer to a form of the Turing Test in which both players pretend to be a computer.

CAPTCHAs generate and grade tests that most humans can pass but current computer programs cannot. Such tests are often called CAPTCHA challenges that are based on hard, open artificial intelligence problems.

2.2 - Applications of CAPTCHA (Scenarios)

CAPTCHAs have several applications for practical security. Preventing Comment Spam in Blogs. Most bloggers are familiar with programs that submit bogus comments, usually for the purpose of raising search engine ranks of some website (e.g., "buy penny stocks here"). This is called comment spam. By using a CAPTCHA, only humans can enter comments on a blog. There is no need to make users sign up before they enter a comment, and no legitimate comments are ever lost.

Protecting free email services. Up until a few years ago, most of these services suffered from a specific type of attack: "bots" that would sign up for thousands of email accounts every minute. The solution to this problem was to use CAPTCHAs to ensure that only humans obtain free accounts. In general, free services should be protected with a CAPTCHA in order to prevent abuse by automated scripts.

Protecting Email Addresses from Scrapers. Spammers crawl the Web searching for email addresses posted in clear text. CAPTCHAs provide an effective mechanism to hide your email address from Web scrapers. The idea is to require users to solve a CAPTCHA before showing your email address.

Online Polls. In November 1999, http://www.slashdot.org released an online poll asking which was the best graduate school in computer science (a dangerous question to ask over the web). As is the case with most online polls, IP addresses of voters were recorded in order to prevent single users from voting more than once. However, students at Carnegie Mellon found a way to stuff the ballots using programs that voted for CMU thousands of times. CMU's score started growing rapidly. The next day, students at MIT wrote their own program and the poll became a contest between voting "bots." MIT finished with 21,156 votes, Carnegie Mellon with 21,032 and every other school with less than 1,000. Can the result of any online poll be trusted? Not unless the poll ensures that only humans can vote.

Preventing Dictionary Attacks [29]. CAPTCHAs can also be used to prevent dictionary attacks in password systems. The idea is simple, prevent a computer from being able to iterate 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.

Search Engine Bots. It is sometimes desirable to keep web pages un-indexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, does not guarantee that bots will not read a web page; it only serves to say "no bots, please." Search engine bots, since they usually belong to large companies, respect web pages that do not want to allow them in. However, in order to truly guarantee that bots will not enter a web site, CAPTCHAs are needed.

Worms and Spam. CAPTCHAs also offer a plausible solution against email worms and spam, "I will only accept an email if I know there is a human behind the other computer." A few companies are already marketing this idea.

3. Existing Solutions

Many CAPTCHA implementations were designed by different companies (Microsoft, Yahoo, AltaVista) in order to offer a more secure online environment. An environment that distinguishes internet communications originating from humans from those originating from software robots. This section is going to present the different types of CAPTCHAs trying to defeat advanced computer programs or bots, discussing the limitations and drawbacks of each.

3.1 - Text-Based CAPTCHAs and Their Limitations

In character labeling based CAPTCHA designs, the computer renders a sequence of letters after distorting them and adding noise. The user is asked to tell what characters they are in order, and will pass the test if the characters typed (new labels) match exactly those known to the server (known labels). Character labeling CAPTCHAs are the most widely used CAPTCHAs. The popularity of such schemes is due to the fact that they have many advantages [4], for example, being intuitive to users world-wide (the user task performed being just character recognition), having little localization issues (people in different countries all recognize Roman characters), and of good potential to provide strong security (e.g. the space a brute force attack has to search can be huge, if the scheme is properly designed).

In 1997, AltaVista developed the first concrete implementation of a CAPTCHA. AltaVista had been receiving

automated URL submissions to their search engine database by spam bots. A group of researchers from the Digital Equipment Systems Research Center were contracted to develop a solution to prevent such an attack [5]. To combat this, the team of developers created a verification system that makes suggestion of recognizing handwritten images. However, they soon realized that although an image containing text was a step in the right direction, it could easily be foiled by use of OCR software. Optical Character Recognition (OCR) software is designed to translate images of text into a machine editable form. The team researched the limitations of scanners with OCR capabilities, and exploited the weaknesses of the OCR systems when rendering their CAPTCHAs. In order to improve OCR results, the manual suggested using similar typefaces, plain backgrounds, and no skew or rotation. To create an image that was resilient to OCR, they did the exact opposite of the suggestions.

In the summer of 2000, Yahoo also began to experience a similar problem where their chat rooms were being spammed by chat bots. This gave birth to the CAPTCHA project. The researchers provided yahoo with three options (see Fig. 1): EZ-Gimpy renders a single, distorted English word on a noisy background, Gimpy-r renders a random string of distorted characters on a noisy background, and Gimpy renders 5 pairs of overlapping distorted words (of which you must type 3).

(a) EZ-Gimpy CAPTCHA (b) Gimpy-r CAPTCHA

(c) Gimpy CAPTCHA

Fig- 1: Examples of EZ-Gimpy, Gimpy-r, and Gimpy CAPTCHAs

In June 2003, shape context matching was used to solve Gimpy with 33% accuracy and EZ-Gimpy with 93.2% accuracy [6]. In June 2004, distortion estimation techniques were used to solve EZ-Gimpy with 99% accuracy and Gimpy-r with 78% accuracy [7]. Due to the limited and fixed size of EZ-Gimpy's dictionary, every challenge image was easily compared against a template database. The distorted template image with the best correlation was returned as the result. However, Gimpy-r does not rely on a dictionary, and therefore requires local distortions to be removed via distortion estimation techniques.

In 2001, researchers at the Xerox Palo Alto Research Center and the University of California at Berkeley synthesized low quality images of machine printed text using a range of words, fonts, and image degradations .Following Baird's quantitative stochastic model of document image quality [8] and a list of problematic OCR examples, noise was introduced into the rendered strings by using two image-degradation parameters, blurring and thresholding (see Fig. 2). A couple of years later, a reading based CAPTCHA known as BaffleText [9, 10] was developed (Fig. 3). BaffleText exercised the Gestalt perception abilities of humans, humans are extremely good at recognizing and understanding pictures despite incomplete, sparse, or fragmented information, where as machines are not.

Fig- 2: PessimalPrint CAPTCHAs

Fig-3: Baffletext CAPTCHAs

OCR systems separate recognition into two sub tasks, segmentation and classification. In 2004, researchers at Microsoft Research exploited the fact that segmentation is much more difficult than classification for OCR systems. So, they developed a CAPTCHA based on hard segmentation problems, as opposed to hard classification problems. Although character classification was still required, the main challenge was correctly segmenting the string. Another contribution was the observation that website owners with CAPTCHAs have the advantage in the battle against CAPTCHA attackers. This is because CAPTCHA generation is a synthesis task while attacking a CAPTCHA is an analysis task. Analysis is orders of magnitude more difficult than synthesis. In the synthesis task, the creator has the ability to use randomness and creativity, while in the analysis task, the

attackers are tightly constrained by the decisions made by the creator.

Fig- 4: Microsoft's Segmentation-Based CAPTCHAs

A formal study of user friendliness for transcription tasks was conducted at Microsoft Research. They studied the effects of varying the distortion parameters and attempted to determine the optimal parameters where the CAPTCHAs prove hard for machines but easy for humans. As researchers found in the past, the most effective CAPTCHAs are segmentation based challenges, which continues to be a computationally difficult task (see Fig. 4). In 2004, researchers at Microsoft Research attacked several commercial CAPTCHA implementations and achieved high accuracy (80%-95%) [11]. Neural networks were used to perform character recognition. Their attacks had the most difficulty with the segmentation task, not the recognition task. Therefore, they suggested that researchers focus their efforts on building CAPTCHAs which rely on the segmentation task instead of the recognition task. It was later confirmed in July 2005 that computers are as good as, or better than humans at classifying single characters under common distortion and clutter techniques. However, other researchers have developed an attack that recognizes the "hard-to-segment" Microsoft CAPTCHA more than 60% of the time.

Figure 5 presents some character based CAPTCHAs that can be sampled from the web while signing up for free e-mail accounts with Mailblocks (www.mailblocks.com), MSN/Hotmail (www.hotmail.com), Yahoo (www.yahoo.com), Google (gmail.google.com), running a whois query at Register.com (www.register.com) or searching for tickets at Ticketmaster (www.ticketmaster.com).

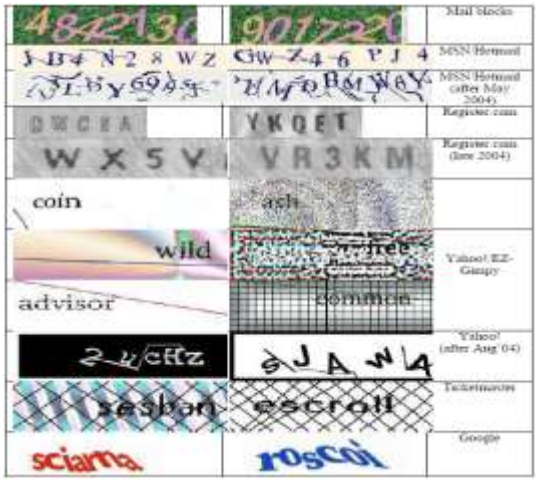

Fig-5: Examples of Various Character Labeling CAPTCHA

Solutions to Yahoo (version 1) CAPTCHAs are common English words, but those for Ticket master and Google do not necessarily belong to the English dictionary. They appear to have been created using a phonetic generator. Examining the changes in MSN, Yahoo, and Register.com HIPs, it can be noted that these CAPTCHAs are becoming progressively more difficult. While MSN introduced more arcs as clutter, Yahoo gave up their language model and replaced simple textures and grids with more random intersecting lines and arcs. Register.com's update was minor as they introduced digits into their character set.

In [12] Chellapilla et al. have discussed the various issues when designing a character labeling based CAPTCHA. They can be summarized as follows: Character set, the character set to be used in the CAPTCHA. Affine transformations, which are Translation, rotation, and scaling of characters. Adversarial clutter represented as Random arcs, lines, or other simple geometric shapes that intersect with the characters and themselves. Image warp such as elastic deformations of the CAPTCHA Image at different scales i.e., those that stretch and bend the character itself (global warp) and those that simply jiggle the character pixels (local warp) and Background and foreground textures which are used to form a colored CAPTCHA image from a bi-level or grayscale CAPTCHA mask. In [13], each character fragment is labelled in order from top to bottom and left to right, and then the components are combined on the idea of jigsaw puzzle to generate candidate characters.

[14] provides a systematic analysis of text-based CAPTCHAs and innovatively improve their earlier attack on hollow CAPTCHAs to expand applicability to attack all the text CAPTCHAs. With this improved attack, they have successfully broken the CAPTCHA schemes adopted by 19 out of the top 20 web sites in Alexa including two versions of the famous ReCAPTCHA. With success rates ranging from 12 to 88.8% (note that the success rate for Yandex CAPTCHA is 0%), they demonstrate the effectiveness of their attack method. It is not only applicable to hollow CAPTCHAs, but also to non-hollow ones.

[15] Presents a novel segmentation and recognition method which uses simple image processing techniques including thresholding, thinning and pixel count methods along with an artificial neural network for text-based CAPTCHAs. We attack the popular CCT (Crowded Characters Together) based CAPTCHAs and compare our results with other schemes. As overall, our system achieves an overall precision of 51.3, 27.1 and 53.2% for Taobao, MSN and eBay datasets with 1000,500 and 1000 CAPTCHAs respectively.

3.2 - Image-Based CAPTCHAs and Their Limitations

While requiring a user to recognize distorted characters is the most common type of CAPTCHA, semantic image understanding tasks have also been proposed. Chew and Tygar from the University of California at Berkeley

investigated a set of three image recognition tasks using a fixed English dictionary of 627 words and Google Images [16, 17]. The Naming images, where the use should determine the common term associated with a set of 6 images (see Fig. 6 left). They used approximate matching to grade the responses. Second, Distinguishing images where the he should determine if two sets of images contain the same subject, and finally identifying anomalies, where he should identify the "odd one out" from a set of 6 images (see Fig. 6 Right)

Fig-6: Examples of Imaged-Based Naming and Anomaly CAPTCHAs

The problems which affected human performance were evaluated and tested during an in-depth user study. Two formal metrics for evaluating CAPTCHAs were also proposed as well as attacks on the three image-based CAPTCHAs. The first metric evaluated CAPTCHA efficacy with respect to the number of rounds of a CAPTCHA and the second metric measured the expected time required for a human to pass the CAPTCHA.

In late 2003, researchers at Microsoft Research argued that the most familiar objects to humans are human faces. They developed a CAPTCHA designed to confuse face recognition algorithms while still being easy to use [18, 19, 20]. Images are automatically synthesized from facial models and the task is to locate and click on the 4 corners of the eyes and 2 corners of the mouth (6 points in total). However, the images looked eerie to many users (see Fig. (7)). For this reason, the system was never adopted.

Fig- 7: Example of an Artifacial CAPTCHA

A similar approach to face recognition based CAPTCHA was developed in 2006 [21]. Photographs of human faces were mined from a public database and distorted. The user is then prompted to match distorted photographs of several different humans. This CAPTCHA has the benefit of being language independent (ignoring textual instructions for completing the task).

In January 2005, some researchers thought that current CAPTCHAs were too demanding of legitimate human users. Instead, they proposed Implicit CAPTCHAs which require as little as a single click [22]. The challenges were so elementary that a failed challenge indicates an attempted bot attack. The authors suggest disguising necessary browsing links in images and claim that bots would not be able to find these hidden links (see Fig. 8). While the usability of the system is attractive, the system could easily be attacked on a case-by-case basis. For example, if the user is told to click on a specific, static place on an image, an attacker would only have to solve this once (challenges are static and therefore are reused). This type of CAPTCHA may work for low traffic or low value services, but it would never survive in a large scale application, as it is impossible to automate the generation of challenges.

Fig- 8: CAPTCHA - User is Instructed to Click on Top of Mountain

One of the more interesting CAPTCHA ideas appeared in January 2011 as a result of an effort by social-networking giant Facebook. The company is currently experimenting with social authentication in an effort to verify account authenticity (see Figure 9).

Fig- 9: Facebook's Friend Recognition Test

What makes Facebook's project slightly different than the normal CAPTCHA is that the authentication is supposed to filter out human hackers rather than machines.

There is potential for Facebook to roll this out across the Web. With 600 million users and millions of websites that integrate with it, Facebook has the ability to use this social recognition CAPTCHA in a big way, and it could prove to be easier than text recognition.

There is one problem. People does not actually know there friends. The reality is that friend requests are exchanged between even the barest of acquaintances, remembering names to go with all those faces could be challenging. As intuitive and intelligent as Facebook's idea might be, it is ultimately flawed because, as humans, we do not follow the rules.

Significant amounts of research have gone into the development of CAPTCHAs over the past 12 years. The first CAPTCHAs required users to transcribe strings of distorted text. Later, more advanced CAPTCHAs which relied on image understanding emerged. Text based CAPTCHAs were usable but easily defeated, while image based ones were affecting human performance. The challenge in designing an effective CAPTCHA is making a compromise, CAPTCHA must not only be human friendly but also robust enough to resist computer programs that attackers write to automatically pass CAPTCHA tests.

Bongo CAPTCHA is named after Mikhail M Bongard who published pattern recognition problems book. In Bongo [23] visual based pattern recognition is provided for the user to solve. The Figure 10 shows an example of Bongo CAPTCHA. It contains 2 block series namely the right block and the left block series. The series of the right block differs from the left blocks, and the user should identify the characteristic which set them apart.

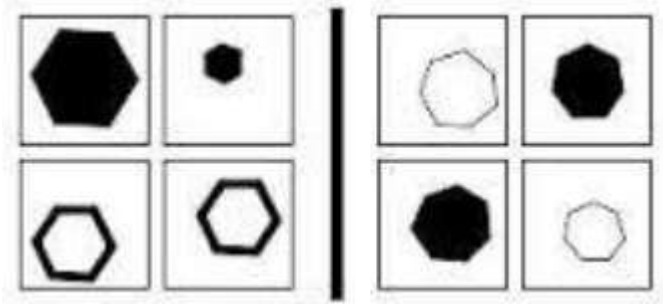

Fig- 10: Example of Bongo catpcha

3.3- Video based CAPTCHA

Video based CAPTCHA system [24] uses a technique in which the video contains few random words. The Figure 11 shows an example of Video CAPTCHA. When the video is played the user has to submit those displayed words. The users need not to wait until the video finishes for submitting the displayed words. The user passes the test only when the ground truth tags which are produced automatically matches with the user entered tags.

Figure 11- Example of Video based CAPTCHA

3.4- Puzzle based CAPTCHA

Puzzle based CAPTCHA can either be a picture based puzzle or a mathematical puzzle. The Figure 12 shows an example of Puzzle based CAPTCHA. In a picture based puzzle, the picture is divided into segments and is shuffled. Each segment will have a segment number followed by the next segment. The user has to combine these segments properly to form a correct complete picture [25]. The mathematical puzzle is 100% effective and can be integrated into login, registration forms in the website for secured access. The user has to solve the math puzzle provided in order to gain the access to secured services.

Fig- 12: Example of Puzzle based Captcha

4. Breaking CAPTCHAs

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. There are two methods in order to break CAPTCHA, OCR and non-OCR.

4.1 - OCR Method

In 2003, Greg Mori and Jitendra Malik have broken successfully Gimpy 33% of time and EZ-Gimpy 92% of time. Their approach is based on three main steps. First, it locates

letters at various location of image by finding bunch of points and then compares candidate letters with each of the 26 letters. Second, it construct graphs between pair of consistent letters which could be formed a complete word. Third, it selects paths through the graph of letters that form real word, and choose the word which has the most matched score for its letters.

J.Yan, A.Ahmad has used pattern recognition approach and debugs some exception that exists in breaking variant of text-based CAPTCHAs [26]. They segment image based on aggregation of foreground pixels and some of conditions on number of pixels. They compare number of foreground pixels in every segment with table of letter-pixel count and recognize the six-letter word with success rate ranging from 92% to 94%. They achieve a success rate of 99% by using snake algorithm to segment random six-letter string. If they use a dictionary for the spelling check, success rate will be 100%. They have achieved success rate ranging from 61% to 63% in recognizing numbers and with snake algorithm increased their success to 100%.

4.2 - Non-OCR Method

In 2005 CAPTCHAs were broken with non-OCR. This method misuse CAPTCHAs which do not kill the session when they are solved correctly. By reusing this session ID, you could request to a CAPTCHA-page automatically. Another misuse of session to break CAPTCHA is when you are on an insecure shared server. Spammer can access to everyone else's session file and abuse its data, such as: its CAPTCHA script. But storing a hash of the CAPTCHA word in the session will rescue this vulnerability.

4.3 - Spammer's Tactics

Spammers use these two tactics to bust CAPTCHA and are successful approximately 30% of time.

4.4.1 - Algorithm-Based

Recently Spammers use a complicated process to bypass Google's CAPTCHA. Since of using variation of CAPTCHA image with Google. They use two hosts on same domain simultaneously in order to achieve high chance of success or check the efficiency and correctness of both host's functionalities. One of them extract an image from a bot-infected machine in format of Bitmap and then break the code. The other one segment the image and send as portable image file then CAPTCHA breaking server responses to each segment request. With success rate of 20%, both of algorithms can break CAPTCHA simultaneously.

4.4.2 - Mechanical Turks

Some companies recruit workers from low-income countries to solve CAPTCHA [27]. There is competition between these CAPTCHA solver companies. E.g., Indian's companies are market leader in this area. Also spammers use Trojan programs to bypass CAPTCHAs. For example, they allow users to view pornographic site by solving some CAPTCHAs or player to play exciting game step-by-step by solving CAPTCHA.

Each type of the CAPTCHAs has its own usability. OCR and non-OCR are two methods to bust the CAPTCHAs that protect many online Web services. Spammers also use expensive Mechanical Turk's method to defeat the CAPTCHA. Designing CAPTCHA that is difficult enough to solve by bots is important, but considering that if the CAPTCHA annoy genuine users is the major challenge.

5. Conclusion

CAPTCHA plays important role in World Wide Web security where it prevents Bot programs and Hackers from abusing online services. In this paper, we have provided a set of techniques that would allow for the system to be secure and less vulnerable to bot attacks. It is a well synthesized CAPTCHA, where the attacker should pass three obstacles in order to bypass it.

REFERENCES

[1] H. S. Baird, A. L. Coates, and R. J. Fateman, "Pessimalprint: A reverse turing test", Int. Journal of Document Analysis and Recognition, 5(2-3):158-163, Seattle, WA, April 2003.

[2] "The Alan Turing Internet Scrapbook, The Turing Test 1950", [Online]. Available:

http://www.turing.org.uk/turing/scrapbook/test.html

[3] K. A. Kluever, "Securely Extending Tag Sets to Improve Usability in a Video-Based Human Interactive Proof", Department of Computer Science Rochester Institute of Technology, Rochester. [Online]. Available: http://www.kloover.com/thesis/proposal.pdf

[4] K. Chellapilla, K. Larson, P. Y. Simard, and M. Czerwinski, "Building segmentation based human-friendly human interaction proofs (hips)", H.S. Baird and D.P. Lopresti (Eds.): HIP 2005, LNCS 3517, pp. 1-26, 2005.

[5] M. D. Lillibridge, M. Abadi, K. Bharat, and A. Z. Broder, "Method for Selectively Restricting Access to Computer Systems," U.S. Patent No. 6,195,698, February 27, 2001

[6] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual Captcha", presented at conf. Computer Vision and Pattern Recognition, vol. 1, pp 134-141, Madison, WI, USA, June 2003.

[7] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion estimation techniques in solving visual Captchas", presented at Conf. on Computer Vision and Pattern

Recognition, vol. 02, pp 23-28, Los Alamitos, CA, USA, June 2004.

[8] H. S. Baird, "Document image defect models and their uses", in Proc of the 2nd Int. Conf. on Document Analysis and Recognition, pp 62-67, Tsukuba Science City, Japan, October 1993.

[9] M. Chew and H. S. Baird, "Baffletext: A human interactive proof", in Proc. of the SPIE/IS&T Document Recognition & Retrieval Conf. X, Santa Clara, CA, pp 305-316, January 2003.

[10] H. S. Baird and M. Luk, "Protecting websites with reading-based aptcha", in Proc. of the 2nd Int. Web Document Analysis Workshop, pp 53-56, Edinburgh, Scotland, August 2003.

[11] K. Chellapilla and P. Y. Simard, "Using machine learning to break visual human interaction proofs (HIPs)", In L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, pp 265-272, Cambridge, MA, December 2004.

[12] L. v. Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. Communications of the ACM, 47(2):56-60, February 2004.

[13] H. Gao, J. Yan, F. Cao et al., "A Simple Generic Attack on Text Captchas," in Proceedings of the Network and Distributed System Security Symposium, pp. 1–14, San Diego, Calif, USA, 2016.

[14] H. Gao, X. Wang, F. Cao et al., "Robustness of text-based completely automated public turing test to tell computers and humans apart," IET Information Security, vol. 10, no. 1, pp. 45–52, 2016.

[15] Rafaqat Hussain, Hui Gao, Riaz Ahmed Shaikh, Segmentation of connected characters in text-based CAPTCHAs for intelligent character recognition, Multimedia Tools and Applications, December 2017, Volume 76, Issue 24, pp 25547–25561.

[16] M. Chew and J. D. Tygar, "Image Recognition Captchas", in Proc. of the 7th Int. Information Security Conf. pp 268-279, Palo Alto, CA, September 2004.

[17] M. Chew and J. Doug Tygar, "Image Recognition Captchas", Tech. Rep. UCB/CSD-04-1333, EECS Department, University of California, Berkeley, August 2004.

[18] Y. Rui and Z. Liu, "Artifacial: Automated reverse turing test using facial features", in Proc. of the 11th ACM Int. Conf. on Multimedia, pp 295-298, New York, NY, USA, November 2003.

[19] Y. Rui and Z. Liu, "Excuse me, but are you human?", in Proc. of the 11th ACM Int. Conf. on Multimedia, pp 462-463, New York, NY, USA, November 2003.

[20] Y. Rui and Z. Liu, "ARTiFACIAL: Automated Reverse Turing test using FACIAL features", presented at Multimedia Syst., pp.493-502, June 2004.

[21] D. Misra and K. Gaj, "Face recognition CAPTCHAs", presented in Int. Conf. on Internet and Web Applications and Services/Advanced International Conf. on Telecommunications, pp 122, Washington, DC, USA, February 2006.

[22] H. S. Baird and J. L. Bentley, "Implicit Captchas", in Proc. of the IST SPIE Document Recognition and Retrieval XII Conf., San Jose, CA, USA, January 2005.

[23] Anju Bala and Baljit Singh Saini, "A Review of Bot Protection using CAPTCHA for Web Security,"(IOSR-JCE) IOSR Journal of Computer Engineering, Volume 8, Issue 6 (Jan. - Feb. 2013), 36- 42.

[24] H. Kwak, M. chew, P. Rodriguez, S. Moon and Y.Y. Ahn, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," In Proc. IMC 2007, ACM Press, 1–14.

[25] Preet Pal and Ved Prakash Singh, "Survey of Different Types of CAPTCHA," / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2242- 2245.

[26] J. Yan and A.S. El Ahmad, "A low-cost attack on a Microsoft Captcha", in Proc. of the 15th ACM Conf. on Computer and Communications Security, Alexandria, VA, USA, October 2008.

[27] B.Vikas, "Spammers Pay Others to Answer Security Tests". The New York Times, April, 2010. [Online]. Available: http://www.nytimes.com/2010/04/26/technology/26captcha.html?src=me&ref=technology.

[28] L. v. Ahn, M. Blum, and J. Langford. Telling humans and computers apart automatically. Communications of the ACM, 47(2):56-60, February 2004.

[29] B. Pinkas and T. Sander, "Securing passwords against dictionary attacks", in Proc. of the 9th ACM Conference on Computer and Communications Security, pp 161-170, New York, USA, November 2002.