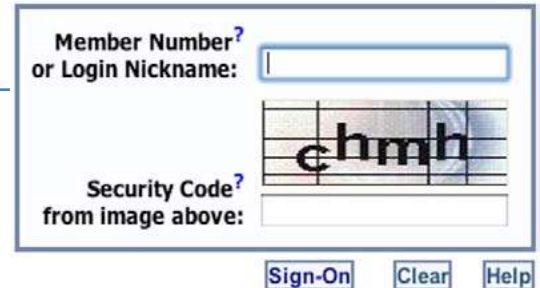


Reverse Turing Testing with CAPTCHA

Using machines to ensure that users are human

By Jason Address – ISSA member, Colorado Springs, USA chapter

This paper discusses the CAPTCHA, which attempts to protect Web-based, automated applications against non-human attackers.



The task of protecting Web-based, automated applications against attacks intended to abuse the functionality of these systems is an ongoing challenge. This paper discusses one possible solution to these issues, the CAPTCHA. The paper discusses the history and functionality of the CAPTCHA, modern methods used to defeat it, and countermeasures to such attacks. Also discussed are advances in artificial intelligence research brought about by the ongoing conflict.

CAPTCHA is an acronym for Completely Automated Turing Test To Tell Computers and Humans Apart. A CAPTCHA is a program that provides protection for Web-based applications by generating problems that are solvable by humans, but not presently easily solvable by machines. The figure below depicts an early example of a CAPTCHA in the form of distorted text.¹

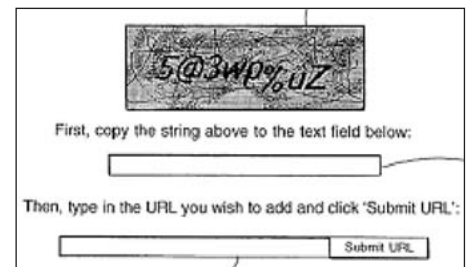


While the earliest work in discerning humans from computers took place in the 1950s with Alan Turing's now-famous Turing test (a test not quite yet successfully passed), the first sign of the technology that would eventually come to be known as the CAPTCHA was discussed in 1996 in *Verification of a Human in the Loop or Identification via the Turing Test* by Naor.² Naor proposed an automated Turing test in which the computer would be a "player trying to establish whether the entity on the other end is a machine or human."

Shortly thereafter in 1997, the first known implementation of such a scheme was developed at AltaVista by Andrei Broder and his colleagues. It was implemented in order to prevent automated programs or "bots" from inserting spurious entries into the listings of their search engine. In order to prevent the images from being easily resolved using software OCR (optical character recognition), Broder's team deliberately

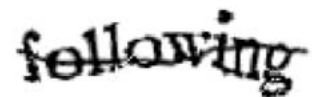
included image artifacts that were problematic for various OCR implementations.³

In April of 1998, a patent was filed by Compaq (who, at that time, owned Alta Vista) for a method for selectively restricting access to computer systems. The figure below shows an illustration⁴ from the patent filing, showing a dialog very similar to that used for filtering submissions to the Alta Vista search engine.



By the year 2000, Luis von Ahn and Manuel Blum of Carnegie Mel-

lon University coined the acronym CAPTCHA. von Ahn and his colleagues developed several varieties of CAPTCHA that improved on Broder's technique by increasing the difficulty of solving the challenge with machine solutions. This was largely accomplished by the addition of features which make segmentation of the image into individual characters more difficult, such as the angled line across the image.⁵



Why are CAPTCHAs useful?

The category of problems solvable by the use of CAPTCHAs generally focuses on the abuse of Web services by automated software. CAPTCHAs can be used in a variety of cases where it is desirable to exclude non-human participants, including user registration, user login, and content viewing, just to name a few.

1 Wikipedia.org, <http://en.wikipedia.org/wiki/Image:Captcha.jpg>.

2 M. Naor, *Verification of a Human in the Loop or Identification via the Turing Test*, 1996 – <http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.pdf>.

3 A. Coates, H. Baird, and R. Fateman, *Pessimist print: A Reverse Turing Test*, 2001 – http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=953966.

4 M. Lillibridge, et. al., United States Patent 6195698, *Method For Selectively Restricting Access to Computer Systems*, 1998.

5 Wikipedia.org, <http://en.wikipedia.org/wiki/Image:Modern-captcha.jpg>.

User registration

The Internet is populated by a wide variety of free Web-based email services, discussion forums, blogs, and other similar tools. Many of these tools, in particular Web-based email, require user registration in order to access their services. Such registrations, however, are very easily abused (e.g., to send spam). The use of CAPTCHA in this situation can prevent abuse of the registration facility.

User login

In password-based systems, CAPTCHAs can be used as a reasonable alternative to locking accounts that have received multiple bad password attempts. Locking accounts under this sort of circumstance can be exploited by nefarious users to perform a denial-of-service attack on the legitimate account owner – not a desirable response. A better alternative might be to prompt the user with a CAPTCHA when the threshold of password attempts has been reached.

Content viewing

In some cases, it is desirable to prevent the contents of a website or other document from being indexed by search engines or viewed in general. While some mechanisms exist to prevent this sort of viewing by automated agents, many of them depend on the agent voluntarily respecting the notice that the site should not be viewed in this way. CAPTCHAs can be used as a gateway to the viewing of content to disallow non-human viewers. A common implementation of this is to put contact information, such as an email address, behind a CAPTCHA to mitigate the problem of *web spiders* reaping addresses from websites.

Developing CAPTCHAs

In essence, a CAPTCHA is generally a challenge-response system that can automatically generate new challenges with a set of features similar to the following:⁶

- Difficult to solve by machines
- Solvable by most humans
- Robust response to attack
- Independent of cultural or language-based knowledge
- Accessible to disabled users

CAPTCHAs frequently present images consisting of a series of alphanumeric characters ranging from a common word to a string of up to eight random characters. These characters are often distorted and overlapped and are placed on a background containing various items that add noise to the image and make the characters more difficult for OCR systems to distinguish from the background.

In order to assist those with accessibility issues (e.g., visually impaired users), many CAPTCHAs also have the capability to play an audio recording of the characters displayed in the CAPTCHA. These audio versions of CAPTCHAs are often not distorted to the same degree as the visual images.

Issues with CAPTCHAs often fall into the category of readability by humans. The line between producing a CAPTCHA image that is too difficult for a machine to solve and one that is easy enough for a human to solve is a fine one. Machine solving technology has reached the point where producing difficult machine problems has also started to mean producing difficult human problems. This issue is exacerbated by focused research on the decoding of CAPTCHAs themselves.

Defeating and protecting CAPTCHAs

Machine solving of CAPTCHAs has become increasingly common as technology progresses. Problems that were completely beyond the possibility of machine solution when CAPTCHA technology was first introduced over a decade ago are now trivial, and security researchers are hard-pressed to stay ahead of attackers. For systems of sufficient complexity as to avoid machine solutions at present, the threat of relay attacks that enable unintended human solvers to bypass the security of CAPTCHA systems still exists. Adding to both of these potential issues is the threat of poor implementation.

Machine solving

Machine solutions to CAPTCHAs generally consist of the processing of an image or audio segment to remove the (intentional) background noise and artifacts to reveal the characters or words that compose the solution. Given the rate at which solutions can be attempted using automation, even a machine solution that is only successful ten percent of the time will still produce a large number of positive results.

Machine solving visual CAPTCHAs

Visual CAPTCHAs can often be defeated through a process that follows steps similar to these:

1. Extract the image from the webpage
2. Remove the background clutter with color filters and detection of thin lines
3. Segment the image into regions each containing a single letter
4. Identify the letter for each region

Steps 1, 2, and 4 are relatively easy tasks for computers. The only part 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 process becomes very difficult with current software.

Thus, CAPTCHAs continue to try to make segmentation more difficult. This can be accomplished by adding noise, additional lines of the same weight as the characters themselves, overlapping of characters, and other similar methods. Unfor-

⁶ H. Baird, *Research on Human Interactive Proofs and CAPTCHAs*, 2003 – http://www.cse.lehigh.edu/~baird/research_hips.html.

tunately (or fortunately, depending on the viewpoint), better OCR techniques are constantly being developed to overcome these issues.

What seems to be almost the certain path of CAPTCHAs is to abandon what has become the traditional CAPTCHA scheme. One possible solution is to move away from the use of alphanumeric characters entirely. Using problems such as “How many seashells are in this picture?” presents a considerably more difficult machine problem, but does require considerable re-coding of the present CAPTCHA applications and possibly more intense use of resources.

Machine solving audio CAPTCHAs

As an accessibility feature, many CAPTCHA implementations provide a recorded voice version of the text needed for the solution. These recordings tend to follow a fixed script, thus making the identification of the needed characters or word for the solution to the CAPTCHA relatively obvious. To make matters worse, such recordings are often the voice of a single person, both within individual CAPTCHAs and across multiple CAPTCHAs.

Given such a uniform format, the solution to breaking audio CAPTCHAs is similar to the solution to breaking visual ones, i.e., segmentation. Once the key word or characters are segmented out, they are relatively easily identified using standard voice recognition technology.

Since solving voice CAPTCHAs relies on one of the same principal techniques as solving visual CAPTCHAs, the same general solution exists as well. Although the voice recording may not be as clear after this is done, breaking up the clean silences of the recording can go a long way toward solving the problem. This might be accomplished by embedding background music in the recording at the same level as the voice that is speaking the components of the CAPTCHA. Another possible enhancement is to include additional gaps in the recording so that the silences between spoken characters are not so easily detected.

Human solving

CAPTCHAs can be defeated by a form of relay attack utilizing humans to solve the problem. The general approach to this kind of attack is to automatically fill the form fields and get to the point of needing to solve the CAPTCHA, then passing the CAPTCHA to a human to be solved. As an incentive for humans to solve CAPTCHAs for this type of attack, it has been rumored that perpetrators of such attacks provide the solvers access to adult materials in exchange for solutions.

Human solving of CAPTCHAs is considerably more difficult to work around than machine solving. In the case of machine solving, the problem can simply be made more difficult. In the case of human solving, the problem is actually at the correct level of difficulty for the person solving it; it is just not being solved by the intended audience. In this case, the solutions to the problem involve making transportation of the problem itself to a different location more difficult.

One possible solution to human solving is to set a very short limit on the time allowed to solve the problem. If the solver only has five seconds to solve the problem, this may not be enough time for the attacker to pick the image up from the site and display it on the attack site for the proxy solver. This, of course, might also cause problems for legitimate users of the site.

Another possible solution might be to make the CAPTCHA generally less portable. This could be accomplished by embedding the CAPTCHA in a Flash animation or streaming video and only showing the user one character at a time. This would force the attacker to move video rather than single images from the originating site to the attack site – a more expensive proposition. Again, this may cause some issues for legitimate users of the site, but perhaps not as many as the previous solution.

Implementation attacks

As with any system, design flaws in the security system implementation can prevent security features from performing their intended task. Such flaws may cause CAPTCHAs to fall prey to very simple forms of attack and leave systems open to abuse or compromise by attackers.

Session reuse

Some CAPTCHAs can be bypassed entirely, without making any attempt at solving by simply re-using the session ID of a known CAPTCHA solution. This information is sometimes encoded in the URL presented after a successful solution has been entered and, in such cases is trivial for an automated attack to re-use.

Session re-use can be defeated by disallowing the re-use of previous sessions. This is, in general, a good security practice for Web-enabled applications, as session re-use can be a very large security hole and may lead to the compromise of the entire application.

Multiple attempts

A CAPTCHA that allows multiple attempts at a solution provides an avenue for decreased difficulty in machine solving. When an attempt is made to solve the CAPTCHA through OCR, the OCR software has a much greater chance of success if allowed to guess multiple times.

The solution to this problem is simple: do not allow more than one attempt on a given CAPTCHA. This not only makes the problem more difficult for machine solving, but will likely result in less frustration for actual human solvers, as they are not being presented with the same potentially difficult CAPTCHA repeatedly.

Weak hashes

Some implementations of CAPTCHA use hashes, such as an MD5, to pass the solution to the client in order to validate the solution to the CAPTCHA. Given the small number of char-

acters that are generally contained in a CAPTCHA, weaker hashing algorithms may produce hashes of a small enough size that they are easily cracked, thus revealing the solution to the CAPTCHA.

A more secure scheme, if such hashes were to be used, would be to utilize a keyed-hash message authentication code, such as HMAC-MD5.⁷ This would provide a considerably more secure solution than the use of a very short MD5 hash.

Small image pools

Some CAPTCHA implementations do not use dynamic images, but instead use a relatively small fixed pool of images. This allows the attacker to record the CAPTCHAs and solutions for easy future indexing. When a given CAPTCHA appears, it can be checked against the index first in order to see if a solution already exists. If so, there is no need to crack the CAPTCHA again.

The solution to fixed image pools is simply to not use them at all, as this method is inherently weak from a security perspective. Dynamically generated images provide an immediate solution to this shortcoming, as they almost entirely preclude such indexing. Many commercial CAPTCHA generating programs exist, including quite a few open source solutions. One popular solution, developed by the team that coined the term CAPTCHA is reCaptcha.⁸ As a basis for CAPTCHA images, reCAPTCHA uses text that has failed OCR attempts from book and newspaper digitization projects and sends the results back to these projects.

The future

Advances in artificial intelligence

CAPTCHAs are based on difficult AI problems, such as resolving distorted characters from background noise. AI research sits directly in between the two sides battling over the problem. On one side are security researchers attempting to keep attackers out with more difficult problems. On the other side are attackers attempting to solve increasingly more difficult problems with machine solutions. One positive benefit of this ongoing conflict is that research in AI is continuously advanced by both sides.

Additionally, CAPTCHAs present legitimate researchers in AI with a simple and fixed problem to solve. Where normally many problems that would be solved by a human need to be stripped down to the bare essentials in order to be worked as an AI problem, the CAPTCHA is already as simple as it needs to be for such work.

The next generation of CAPTCHA

The next generation of CAPTCHA will likely lay in the direction of more logically complex tests. When a test is used

that depends on being able to sort out alphanumeric characters from surrounding background noise, this test is only good for as long as it takes for image recognition technology to progress. The use of this sort of CAPTCHA is finite in nature as these problems can only be increased in complexity to a certain point, otherwise they will become too difficult for humans to solve reasonably.

When using tests that are more logically complex, the test changes from sorting out alphanumeric characters from background noise to tests like deciding which of a group of people is the most attractive or tests involving optical illusions. These sorts of problems are almost infinitely more difficult for machines to solve and should be considerably easier for people to solve.

Other approaches

It may be that the CAPTCHA is not the best long-term method for protecting Web-enabled services from abuse. In the long run, as the types of problems on which CAPTCHA depends become more easily solved through advances in AI and better machine solving techniques, CAPTCHA will need to be replaced or used in combination with other methods.

One promising such method that could be used in conjunction with CAPTCHA is particularly appealing for pitting machine against machine. The use of heuristic analysis on patterns of usage for Web-enabled services has the potential to be very useful. One of the flaws of CAPTCHA is in the single-layered nature of its protection mechanism. The CAPTCHA is adequate until it is broken, and then there are no further mechanisms in place to prevent the service from being abused. Given the nature of the abuses of such systems, when an attack along these lines occurs, it should be easily detectable to a heuristic system. For example, a webmail account under normal use might send ten email messages over the course of a day. This could be evaluated by performing a baseline analysis of the mail usage across the system. When an account deviates from the baseline by a certain percentage, say by sending twenty messages in an hour, the web interface could present the user with a more difficult class of CAPTCHA in order to verify that the account has not been compromised.

Many other such similar solutions exist, such as the use of OpenID, PKI, biometric systems, and scores of others. The only thing that is certain is that the present solutions will eventually be overcome, and new problems will need to be posed to supplant them.

About the Author

Jason Andress is a tinkerer, rascalion, and all around geek. He works for a major software company, teaches graduate and undergraduate security courses, and enjoys a good game of Scrabble. He can be reached at jason.andress@gmail.com.



7 H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication* (RFC2104), 1997.

8 Carnegie Mellon University, reCAPTCHA: Stop Spam, Read Books, 2009, <http://recaptcha.net>.