

Running head: CAPTCHA AND WEB SECURITY

CAPTCHA as a Web Security Control

Richard V. Hall, Jr.

University of Houston-Victoria

Original Version Date: 2005-08-24

Current Revision Date: 2005-12-17

Copyright © 2005 Richard Van Hall, Jr.

All Rights Reserved.

### Abstract

This paper covers how Web professionals can combat certain automated attacks with CAPTCHA. Web sites are increasingly under attack from automated scripts. In many cases, sites could deter automated attacks if they could distinguish human users from machine users. Artificial Intelligence experts identify a category of security controls called “CAPTCHAs” which can help Web sites distinguish between human and machine users by posing a problem that is easy for humans to solve, but difficult for machines to solve. The paper discusses CAPTCHA’s variety of modes, implementations, methods of deployment, relative advantages and disadvantages, effectiveness against various attacks, role in security strategies, accessibility considerations, known weaknesses, pitfalls to avoid, and practicality in Web applications.

## CAPTCHA as a Web Security Control

CAPTCHA can control many automated attacks against Web sites, but without careful planning, it can also cause problems. Usually presented as a distorted image of a password to enter, CAPTCHA also comes in a variety of other forms, each with its own advantages and disadvantages. Automated attacks can exploit public Web services for several purposes: placing thousands of unsolicited messages on blogs, forums, guest books, and Wikis; linking to forged Web sites for identity theft (phishing); promoting a product, service or agenda; conducting traditional scams and fraud; vandalizing sites; flooding a site with useless comments; raising a link's rating in search engines; ballot-stuffing of online polls; theft of data available only by paid subscription; copyright infringement by "scraping" data from one site to display on another; and unwanted indexing by spiders that ignore robots.txt files and nofollow attributes. Automated attacks work because the Web treats machine users (called bots) and human users the same. However, there are also occasions when blocking bots would reduce many security threats. Authentication, IP tracking and banning, and message moderation can control many malicious human and machine users, but not sophisticated attacks.

Artificial Intelligence (AI) experts propose CAPTCHA as a category of security controls to help Web sites distinguish between human and machine users. Turing (1950) proposes a test for AI in which a computer must fool a panel of humans into believing the machine is human. Naor (1997) discusses an Automated Turing Test (ATT) in which computers, rather than humans, must determine whether a subject is human or machine. Although Coates, Baird, and Fateman (2001) call Naor's ATT a Reverse Turing Test (RTT), Ahn, Blum, and Langford (2004) point out that RTT refers to an unrelated kind of test. Blum, Ahn, and Langford (2000) propose a class of ATTs called a Human Interactive Proof (HIP), which Hopper (2001) describes as a

protocol "that allows a human to prove something to a computer" (p. 1). Hopper and Blum (2001) propose a HIP called a Secure Human Identification Protocol, or HUMANOID, in which a computer must verify a human's membership in a group without requiring a password, biometric data, electronic key, or any other physical evidence. A HUMANOID test must also remain effective even when others know and witness the authentication process. Blum, Ahn, and Langford (2000) further propose a more specific form of HIP called a Completely Automated Public Turing Test to Tell Humans and Computers Apart (CAPTCHA) in which the computer must be able to "generate and grade" a test "that most humans can pass", but "current computer programs cannot pass" (p. 1). Broder (2002) describes how he and Martin Abadi, Krishna Bharat, and Mark Lillibridge of Compaq patented the use of text images to deter bots, which Compaq licensed to AltaVista in 1998 for its first implementation by Laurent Chavez. According to Luk (2003), AltaVista used the system to deter automated submission of URLs to its search database. Spice (2001) reports how Yahoo's chief scientist Uri Manber challenged Blum at Carnegie-Mellon University to keep bots out of Yahoo's chat rooms. Blum, Ahn, and Langford (2000) assert that Compaq's patent is not a full CAPTCHA because custom OCR software could break it. Blum, Ahn, Hopper, and Langford implemented a CAPTCHA called EZ-Gimpy, which Manber deployed to deter spam attacks launched from Yahoo e-mail addresses registered in bulk (Spice, 2001). Computer vision experts Mori and Malik (2002) now defeat EZ-Gimpy "with a success rate of 94%" (p. 1), but according to Ahn, Blum, and Langford (2004), such cracks are proof that "CAPTCHA implies a win-win situation: either the CAPTCHA is not broken and there is a way to differentiate humans from computers, or the CAPTCHA is broken and a useful AI problem is solved" (p. 60).

CAPTCHAs can take a variety of forms. Reading CAPTCHAs show a cluttered image of a distorted password for users to type (Blum, Ahn & Langford, 2000). Shape CAPTCHAs show complex shapes for the user to identify (Malik, 2002). A spatial CAPTCHA's text image is rendered from a three-dimensional model (OCR Research, 2004). Speech CAPTCHAs play a distorted sound file over cluttered background noise (Blum, Ahn & Langford, 2000). Quiz CAPTCHAs show a visual or audio puzzle or trivia question that a computer can generate and display, but not solve (Blum, Ahn & Langford, 2000). Match CAPTCHAs show a set of related images or sounds and ask the user to identify their common theme (Blum, Ahn & Langford, 2000). A text-only CAPTCHA shows a reading, quiz, or match CAPTCHA using only plain text (Godfrey, 2001). A Virtual Reality (VR) CAPTCHA shows a three-dimensional (3D) world for the user to navigate (Perrig & Song, 2002). A natural CAPTCHA uses media files that record the real world rather than synthesizing them from scratch (Lopresti, 2005). An implicit CAPTCHA blends so well into the flow of a Web site that users may not even know it is testing them (Baird & Bentley, 2005).

Each form of CAPTCHA can have multiple implementations, each with its own advantages and disadvantages. Reading CAPTCHAs are the most common, and among the most reliable. Blum, Ahn, and Langford (2000) propose EZ-Gimpy and Gimpy, which shows five pairs of overlapping words, three of which a user must identify. Coates, Baird, and Fateman (2001) propose Pessimial Print, which simulates dirty scans of printed text proven extremely difficult over 40 years of Optical Character Recognition (OCR) research. Mori and Malik (2002) demonstrate an image filtering and dictionary attack (Mori-Malik) with 94% success against EZ-Gimpy and 33% success against Gimpy. Spitz (2002) describes "character shape coding" that uses lowercase type to help break CAPTCHAs simply by noting ascenders or descenders.

Although Chew and Baird (2003) describe multi-font Pessimal Print success against Mori-Malik, they report a 40% penetration rate when Mori-Malik is trained for the font. In their BaffleText proposal, Chew and Baird (2003) attempt to overcome Mori-Malik with "non-English 'pronounceable words'" and "Gestalt-motivated image-masking degradations" using larger shapes (p. 22). BaffleText tests 89% human-readable and 89% resistant to Mori-Malik when adjusted to match Mori-Malik's expectations. However, El-Nasan and Nagy (2002) propose a form of handwriting recognition using statistical analysis of the likelihood of letters to appear together in a language (n-grams), which could diminish any advantage of using pronounceable non-English words. Baird, Riopka, Bently, Moll, and Wang (2005) mention that Patrice Simard of Microsoft Research says OCR can break many popular CAPTCHAs by segmenting the image into letters, recognizing the letters, and then using a dictionary to resolve any ambiguity. To defeat the segmentation attack, Baird and Riopka (2005) propose ScatterType, a method of visually shattering each letter into pieces and overlapping them randomly. Baird, Moll, and Wang (2005) study ScatterType's legibility to improve its human recognition rate. Rusu and Govindaraju (2005) propose H-CAPTCHA as a way to use degraded handwritten names of places. H-CAPTCHA is designed to defeat Automated Handwriting Recognition (AHR) similar to the way that Pessimal Print defeats OCR.

Quiz CAPTCHAs are rarely both strong enough to deter machines or easy enough for humans to solve reliably. Blum, Ahn, and Langford (2000) propose Bongo as the prototypical quiz CAPTCHA, which presents a multiple-choice visual puzzle reminiscent of Mensa tests. A quiz CAPTCHA might also pose a trivia question. One problem with a quiz CAPTCHA is that it must assume a common base of factual knowledge that most humans already know, and that most computers cannot learn. Burnett and Foster (2004) point out that the strength of a quiz

CAPTCHA is often compromised by how often a random guess can be right. In their example, they describe a system that shows a photo of one to three zebras and asks how many zebras are in the photo. Because a random guess will be right one out of three times, then a spammer who writes a script to register 1000 e-mail addresses can expect to get 333, by the law of averages. In the official Bongo example, a random guess would be successful 50% of the time.

Match CAPTCHAs usually require huge stores of tagged media files and present users with far too many choices, but hash visualization may alleviate the need to select from a list. Blum, Ahn, and Langford (2000) proposed Pix as the prototypical matching CAPTCHA, which presents a set of four related photographs and asks the user to select, from a list of hundreds, the keyword that most accurately describes what the photos have in common. For Pix, the user must choose from a list because one person's "eagle" could be another person's "bird." Perrig and Song (2002) discuss Déjà Vu as a matching CAPTCHA based on "hash visualization," requiring humans to recognize an object they have seen before. Whereas Pix suffers from humans identifying multiple words with an image, Perrig and Song (2002) describe Deja Vu as a way to associate images with a specific word, such as the name of a person or place. Poorly implemented Pix could be simple to crack if the image database is small and the images are easily distinguished from one another by such factors as their pixel dimensions or the color of their center pixels. The need to present a long list of possible solutions could be overcome by superimposing the distorted and scrambled letters of a unique solution on the photo of an object, so that a human who sees a photo of a bird and the letters GELEA would know that EAGLE, not BIRD, is the right solution.

Speech and audio CAPTCHAs are growing in popularity as companions to reading CAPTCHAs for the visually impaired, but they do not provide full accessibility, and could

potentially weaken protection. Blum, Ahn, and Langford (2000) proposed the Sounds test as a speech, matching, or quiz CAPTCHA using sound files instead of images. Lopresti, Shih, and Kochanski (2001) apply the Pessimist Print approach to the Sounds CAPTCHA such that their speech CAPTCHAs use known limitations of Automatic Speech Recognition (ASR) to generate sounds that humans can understand, but are too distorted or cluttered (with background music or noises) for ASR to detect. Chan (2002) discusses two approaches to cluttering Sounds called BYAN-I and BYAN-II. BYAN-I overlays six digits spoken in the subject's native language with the same six digits spoken in a foreign language, but custom ASR might break it. In BYAN-II, the background noise is a random set of words in a foreign language, which may be more difficult to break, but require more tests. Blum (2002) proposes a blend of speech CAPTCHA with HUMANOIDs called PhonOIDs to authenticate users over phones. Speech CAPTCHAs can extend access to the visually impaired who are otherwise blocked by visual CAPTCHAs. The dual audio-visual implementation at [www.captchas.net](http://www.captchas.net) can be problematic because it uses the same password for both the image file and the sound file, so sophisticated attack could improve its accuracy by evaluating both files in tandem. A better solution would be to use a different password for each form of the CAPTCHA.

Text-only CAPTCHAs would solve the problems of accessibility, bandwidth, and server load, but unfortunately, none are yet strong enough to withstand a directed attack. Godfrey (2001) and Przydatek (2002) discuss a variety of text-only CAPTCHAs and explain that all text-only CAPTCHAs appear to be easier to solve than image or sound CAPTCHAs. Baird, Riopka, Bently, Moll, and Wang (2005) describe how people began to disguise their e-mail addresses from spammers in the mid-1990s by writing them out as "baird AT cse DOT lehigh DOT edu" (p. 3).

Spatial CAPTCHAs have the potential to be the most machine-resistant form of reading CAPTCHA, but the amount of processing power required to render them could be prohibitive. Although an unrendered 3D model or rendered animation might suffer from an excess of data available to be cracked by pattern matching algorithms, the OCR Research Group's tEABAG\_3D prototype has shown that rendering 3D text to an image can add the difficulty of spatial recognition to the difficulty of OCR (<http://ocr-research.org.ua/>). However, rendering a large number of 3D images on high-traffic Web servers could be impractical, because Web servers are not typically equipped with powerful graphics cards. One solution could be to render one image every few minutes or hours, so that all users during that time period would use the same password, but a few minutes or hours might be all the time a cracker needs to manually discover the password and launch an attack. Another solution might be to render 3D images offline as a matching, quiz, natural, or implicit CAPTCHA. For example, a specialized 3D CAPTCHA server might grab a tagged photo from Flickr.com, reduce the image to a standard size, scramble the letters in a tag, render 3D letters over a slightly distorted photo, and present the result as a puzzle. Such a server might generate thousands of new challenges each day.

Virtual Reality CAPTCHAs are not currently practical for high-traffic Web sites, but with advances in streaming video and Web3D, they might make niche appearances on the Web in the future. Perrig and Song (2002) discuss Map as a VR CAPTCHA that requires the subject to navigate between two random points in a three-dimensional world or maze. The subject, called a driver, knows the Map, but onlookers, called passengers, have difficulty remembering a path between points, even if they see it. Machines that do not have access to the Map and cannot form a spatial model of the scene will inevitably take a wrong turn and fail the test.

Natural CAPTCHAs are among the most promising new developments in CAPTCHA technology. Lopresti (2005) describes a way to strengthen visual CAPTCHAs, both against machines and in favor of humans, by using “natural CAPTCHAs” taken from scans or photos of real documents, rather than "synthetic CAPTCHAs" generated entirely by machine. He asserts that with a large enough external supply of tests and graders, the machine need not actually generate and grade the test for it to be practical. By "harvesting" natural images and reliable human graders from the Internet, a CAPTCHA may strengthen beyond current limitations. Graders can be harvested in a process called "Collaborative Filtering" attributed to Chew and Tygar (2005) by asking subjects to solve more than one CAPTCHA. The computer knows the answer to the first CAPTCHA, and uses it to validate the user. The computer does not know the answer to the remaining CAPTCHAs, but uses the subject's input to discover the answer. Lopresti (2005) also promotes the revival of the Open Mind Initiative ([openmind.org](http://openmind.org)) as a source of material for CAPTCHAs.

Implicit CAPTCHAs are among the easiest to implement, and among the few that attempt to overcome social engineering attacks, which recruit unwitting human users of another Web site to provide solutions to a CAPTCHA. Baird and Bentley (2005) propose "Implicit CAPTCHAs" specifically as a way to make CAPTCHAs less offensive to humans and less susceptible to social engineering attacks. Implicit CAPTCHAs masquerade as normal links, provide reasonably high accuracy in one click, require the user to have experienced the Web site, and provide human success so easily that failure is near certainty of a machine attack. They give an example of a photograph with several meaningful lines of text placed on it almost randomly, one line of which is a link to the next page. If only a small portion of the pixels in the image will result in success, a random guess should have a high failure rate. In the case of a poll, each option is

presented as an image of degraded text, and the order of the options is randomized. A user can easily distinguish among the options, while a machine will have much more difficulty selecting the right option. To resist against a flood of meaningless votes, the poll could present the submit button as an image with one positive word hidden among many negative words, each randomly placed and degraded.

Web sites can deploy CAPTCHAs either by installing turnkey CAPTCHA software, by programming custom CAPTCHA software, or by subscribing to a remote CAPTCHA service. Installing existing CAPTCHA software comes with the same considerations as installing any packaged security control, like a firewall or antivirus. To be effective, the software makers have to update the software frequently to patch vulnerabilities in previous versions and to combat cracks. There are dozens of strong turnkey options among thousands of weak options. The popular solutions suffer from being big targets, while the less popular solutions are usually less effective. Oftentimes, price, server capabilities and accessibility standards will limit the available alternatives to only a few. Some of the weaknesses of using turnkey CAPTCHA can be alleviated by customizing an Open Source CAPTCHA or by building a new CAPTCHA from scratch. By introducing unique variations or completely different approaches to CAPTCHA, your own installation will not be threatened whenever a turnkey CAPTCHA is cracked. However, a higher degree of expertise may be required, followed by a commitment to maintain the code. Inattention to detail in either phase can result in opening more security holes than are closed. There are several advantages of subscribing to a service: the code that generates the CAPTCHA images and sounds will not bog down the subscriber's server; the bandwidth for transferring the image and sound files is assumed by the remote service; the remote service keeps its own software up-to-date, requiring less attention from the subscriber. The two primary

disadvantages of subscribing to a CAPTCHA service are that another potential avenue of attack results from passwords or keys exchanged over the network, and that all remote CAPTCHA services require one organization to trust another with the power to abuse the site. With the [www.captchas.net](http://www.captchas.net) service, secret key encryption and MD5 digests allow the subscriber to securely send a random string to the service, which decrypts the string and returns the corresponding image or sound files. Although the password is relatively secure, the need to transfer it to another server exposes yet another option for cracks. There is a serious issue with trust whenever one company out-sources its security to another. A security provider's employees may experience strong temptation to abuse their power over the subscribers. In the case of [www.captchas.net](http://www.captchas.net), the service provider has the secret key, which can be used symmetrically to gain access to any site that subscribes to the service. In the case of a public Web form, however, the risk would not seem to be as great as for a bank outsourcing its money transfer codes to another firm. Assuming other security measures are in place and the CAPTCHA is only being used to keep scripts out of an area already available to the human public, the potential damage from a rogue employee would not seem to be greater than the potential damage from any other member of the public.

In conclusion, care must be taken when installing a CAPTCHA to avoid causing new problems, particularly with regard to accessibility. Federal regulations and professional ethics require Web sites to remain accessible to users with disabilities, but many CAPTCHA implementations result in denial of service to people with visual impairments, auditory impairments, or both; so the W3C has issued a set of guidelines for CAPTCHA accessibility at <http://www.w3.org/TR/turingtest/> which all Web professionals should review before installing a CAPTCHA. In addition, some CAPTCHA software opens new security holes. Overall,

CAPTCHAs that rely on virtual reality, spatial recognition, matching, plain text, or quizzes may not be practical enough or secure enough for the Web. Natural, implicit, reading, and speech CAPTCHAs are currently more promising. Web professionals should carefully test a CAPTCHA's strengths and weaknesses before integrating it into their sites, but if properly implemented and deployed, many CAPTCHAs can be effective controls against the most common forms of automated attacks.

## References

- Aboutafadel, E., Olsen, J., & Windle, J. (2005). Breaking the Holiday Inn Priority Club: CAPTCHA. (Completely Automated Turing tests to tell Computers and Humans Apart). *The College Mathematics Journal*, 36(2), 101-108.
- Ahn, L. von, Blum, M., Hopper, N. J., & Langford, J. (2003). *CAPTCHA: Using hard AI problems for security*. Retrieved October 10, 2005 from <http://www.captcha.net>
- Ahn, L. von, Blum, M., & Langford, J. (2004). Telling computers and humans apart automatically. *Communications of the ACM*, 47(2), 57-60.
- Baird, H.S., (2002). The ability gap between human and machine reading systems. *Proceedings of the First HIP Conference, 2002*.
- Baird, H.S., & Bentley, J.L. (2005). Implicit CAPTCHAs. *Proceedings, IS&T/SPIE Document Recognition & Retrieval XII Conference*. San Jose, CA. January 16-20, 2005.
- Baird, H.S., Moll, M.A., & Wang, S.Y. (2005). ScatterType: a legible but hard-to-segment CAPTCHA. *Proceedings, IAPR 8<sup>th</sup> International Conference on Document Analysis and Recognition*. Seoul, Korea. August 29-September 1, 2005.
- Baird, H.S., & Riopka, T. (2005). ScatterType: a reading CAPTCHA resistant to segmentation attack. *Proceedings, IS&T/SPIE Document Recognition & Retrieval XII Conferences*. San Jose, CA. January 16-20, 2005.
- Baird, H.S., Riopka, T., Bentley, J., Moll, M.A., & Wang, S.Y. (2005). Protecting e-commerce from robots impersonating human users. Retrieved November 11, 2005 from <http://www.cse.lehigh.edu/pr/HIP05/Presentations/>
- Barnett, M.M., & Foster, J.C. (2004). *Hacking the Code: ASP.NET Web Application Security*. Syngress Publishing, Inc. Rockland, MA.

- Blum, M., Ahn, L. von, & Langford, J. (2000). The CAPTCHA Web site. Retrieved November 11, 2005 from <http://www.captcha.net>
- Blum, M. (2002). PhonOID protocol class #81. *Proceedings of the First HIP Conference, 2002*.
- Broder, A., Preventing bulk URL submissions by robots in AltaVista. *Proceedings of the First HIP Conference, 2002*.
- Chan, N. (2002). Sound oriented CAPTCHA. *Proceedings of the First HIP Conference, 2002*.
- Chew, M., & Baird, H.S. (2003). BaffleText: a Human Interactive Proof. *Proceedings of the SPIE/IS&T Document Recognition and Retrieval Conf. X*. Santa Clara, CA. January 22-23, 2003.
- Coates, A.L., Baird, H.S., Fateman, R.J. (2001). Pessimial Print: a reverse Turing test. *Proc. IAPR 6<sup>th</sup> Intl. Conference on Document Analysis and Recognition*. Seattle, WA. September 10-13, 2001, 1154-1158.
- El-Nasan, A., & Nagy, G. (2002). On-line handwriting recognition based on bigram co-occurrences. *Preprint ICPR-02*.
- Godfrey, P.B. (2001). Text-based CAPTCHA algorithms. *Proceedings of the First HIP Conference, 2002*. December 15, 2001.
- Hopper, N.J., & Blum, M. (2001). Secure human identification protocols. *ASIACRYPT 2001*. LNCS 2248, 52-66.
- Hopper, N.J. (2001). Security and complexity aspects of Human Interactive Proofs. *Proceedings of the First HIP Conference, 2002*. December 3, 2001.
- Juels, A. (2002). At the juncture of cryptography and humanity. *Proceedings of the First HIP Conference, 2002*.

- Lopresti, D., Shih, C., & Kochanski, G. (2001). Human Interactive Proofs for spoken language interfaces. *Proceedings of the First HIP Conference, 2002.*
- Malik, J. (2002). Visual shape recognition and CAPTCHAs. *Proceedings of the First HIP Conference, 2002.*
- Mori, G., & Malik, J. (2003). Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2003. Retrieved October 10, 2005 from <http://www.cs.sfu.ca/~mori/research>
- Perrig, A., & Song, D. (2002). New directions for user authentication: reflex instead of reflection. *Proceedings of the First HIP Conference, 2002.*
- Przydatek, B. (2002). On the (im)possibility of a text-only CAPTCHA. *Proceedings of the First HIP Conference, 2002.*
- Rusu, A., & Govindaraju, V. (2005). Visual CAPTCHA with handwritten image analysis. Retrieved November 11, 2005 from <http://www.cse.lehigh.edu/prr/HIP05/Presentations/>
- Spice, B. (2001). Robot solves Internet robot problem. *Pittsburgh Post-Gazette*. October 21, 2001. Retrieved November 11, 2005 from <http://www.post-gazette.com/healthscience/20022021blumside1021P4.asp>
- Spitz, A.L. (2002). A feeble classifier relying on strong context. *Proceedings of the First HIP Conference, 2002.*
- Turing, A.M. (1950). Computing machinery and intelligence. *Mind* 59, 236, 433-460.
- Lopresti, D. (2005). Leveraging the CAPTCHA problem. *Proceedings of the Second HIP Conference, 2005.*