ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (314-319), Month: July - September 2015, Available at: www.researchpublish.com

Reusable Captcha Security Engine

Mr. Ujwal Namdeo Abhonkar[1], Mr. Swapnil Mohan Phalak[2],
Mrs. Pooja Ujwal Abhonkar[3]

[1,3] Lecturer in Computer Engineering Department
[2]Lecturer in Information Technology Department
[1,2]Sandip Polytechnic, Nashik, India, [3]LGG Polytechnic, Nashik, India

Abstract: **A CAPTCHA is a type of challenge-response test used in computing to determine whether the user is human. "CAPTCHA" is a contrived acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart", trademarked by Carnegie Mellon University. A CAPTCHA involves one computer (a server) which asks a user to complete a test. While the computer is able to generate and grade the test, it is not able to solve the test on its own. Because computers are unable to solve the CAPTCHA, any user entering a correct solution is presumed to be human. A CAPTCHA is sometimes described as a reverse Turing test, because it is administered by a machine and targeted to a human, in contrast to the standard Turing test that is typically administered by a human and targeted to a machine. For example, human can generally read degraded images, but OCR machines cannot. CAPTCHAs are designed to prevent bots – programs that pose as humans on the Internet – from abusing internet services. Bots, driven not to dominate but to sell, sign up for thousands of free email accounts every minute, sending millions of spam messages from them. They infiltrate chat rooms, collecting personal information and posting links to promotional sites. They generate worms, break password systems, invade privacy, and drain resources. To defend e-commerce systems from bots, an increasing number of companies are arming themselves with CAPTCHAs. For example, users registering on Yahoo must first correctly recognize a distorted word displayed against a cluttered background and type it into a box to prove they are human. Such reading-based CAPTCHAs exploit the large gap between humans and machines in their ability to read images of text.**

Keywords: **Captcha, Images, OCR machines, Turing test.**

I. INTRODUCTION

CAPTCHA can be deployed to protect systems vulnerable to e-mail spam, such as the webmail services of Gmail, Hotmail, and Yahoo!. CAPTCHA have also found active use in stopping automated posting to blogs or forums, whether as a result of commercial promotion, or harassment and vandalism. CAPTCHA also serve an important function in rate limiting, as automated usage of a service might be desirable until such usage is done in excess, and to the detriment of human users. In such a case, a CAPTCHA can enforce automated usage policies as set by the administrator when certain usage metrics exceed a given threshold. An example of a system in which vulnerabilities exist, which could easily be prevented using CAPTCHA.

A CAPTCHA system is a means of automatically generating new challenges which:

- Current computers are unable to accurately solve.
- Most humans can solve.
- Does not rely on the type of CAPTCHA being new to the attacker. Although a checkbox "check here if you are not a bot" might serve to distinguish between humans and computers, it is not a CAPTCHA because it relies on the fact that an attacker has not spent effort to break that specific form.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (314-319), Month: July - September 2015, Available at: www.researchpublish.com

II. LITERATURE SURVEY

A way to tell apart a human from a computer by a test is known as a Turing Test. When a computer program is able to generate such tests and evaluate the result, it is known as a CAPTCHA (Completely Automated Public test to Tell Computers and Humans Apart). In the past, Websites have often been attacked by malicious programs that register for service on massive scale. Programs can be written to automatically consume large amount of Web resources or bias results in on-line voting. This has driven researchers to the idea of CAPTCHA-based security, to ensure that such attacks are not possible without human intervention, which in turn makes them ineffective. CAPTCHA-based security protocols have also been proposed for related issues, e.g., countering Distributed Denial-of-Service (DDoS) attacks on Web servers. A CAPTCHA acts as a security mechanism by requiring a correct answer to a question which only a human can answer any better than a random guess. Humans have speed limitation and hence cannot replicate the impact of an automated program. Thus the basic requirement of a CAPTCHA is that computer programs must be slower than humans in responding correctly. To that purpose, the semantic gap between human understanding and the current level of machine intelligence can be exploited. Most current CAPTCHAs are text-based.

Commercial text-based CAPTCHAs have been broken using object-recognition techniques, with accuracies of up to 99% on EZ-Gimpy. This reduces the reliability of security protocols based on text-based CAPTCHAs. There have been attempts to make these systems harder to break by systematically adding noise and distortion, but that often makes them hard for humans to decipher as well. Image-based CAPTCHAs have been proposed as alternatives to the text media. More robust and user-friendly systems can be developed. State-of-theart content-based image retrieval (CBIR) and annotation techniques have shown great promise at automatically finding semantically similar images or naming them, both of which allow means of attacking image-based CAPTCHAs. Generally CAPTCHAs are look like shown as below:

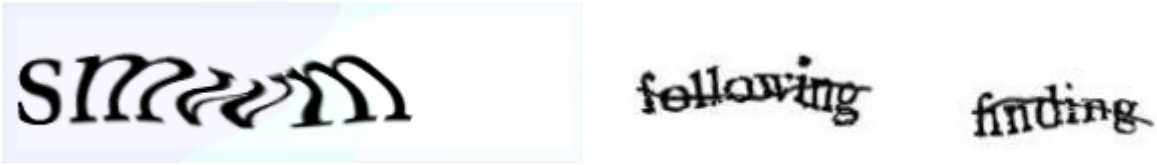

Fig.1 Modern CAPTCHA

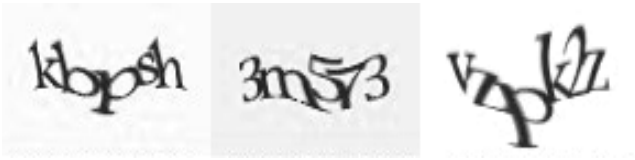

Fig.2 CAPTCHA with crowded symbols

Fig.3. Shadow Filter

A. *Existing System:*

The existing system consists of the typing test given to help determine if the person typing is really a human or some bot. It can be an anti-spam device. You must prove yourself a true human posting some message or recommending URL rather than a computer program spamming the universe. The test typically requires you to type in some warped series of letters.

Limitations in Existing System:

There is no security in those anti-spam devices, because while transferring URL to that anti-spam device there may be spammers to hack the data.

There is a very long process to type required test into the specified text box to prove user self as human in the registration or form filling process in sites.

B. *Proposed System:*

The proposed system consists of a CAPTCHA is a program that protects websites against bots by generating and grading tests that humans can pass but current computer programs cannot.

C. *Problem Definition:*

The main objective of the project is to generate the CAPTCHA images, provide a secure Form filling interface for the internet based Applications. Provide environment for the user to handle manually the form filling task. Provide the interface for the user to identify the image and fill the specified text box.

It is sometimes rumored that spammers are using pornographic sites to solve CAPTCHAs: the CAPTCHA images are sent to a porn site, and the porn site users are asked to solve the CAPTCHA before being able to see a pornographic image. This is not a security concern for CAPTCHAs. While it might be the case that some spammers use porn sites to attack CAPTCHAs, the amount of damage this can inflict is tiny (so tiny that we haven't even noticed a dent!). Whereas it is trivial to write a bot that abuses an unprotected site millions of times a day, redirecting CAPTCHAs to be solved by humans viewing pornography would only allow spammers to abuse systems a few thousand times per day. The economics of this attack just don't add up: every time a porn site shows a CAPTCHA before a porn image, they risk losing a customer to another site that doesn't do this.

Advantages over Existing System:

- background colors
- background gradient fill colors
- fonts (2 default and limited to the amount of font's on your system)
- font color
- random character generator (characters can be configured)
- optional border around the CAPTCHA
- border color (defaults to black)
- border thinness (defaults to one)

III. METHODOLOGY

A. Implementation:

The system would be implemented in a web based and collections environment.

The following guidelines are strongly recommended for any CAPTCHA code:

Accessibility: CAPTCHAs must be accessible. CAPTCHAs based solely on reading text — or other 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. Any implementation of a CAPTCHA should allow blind users to get around the barrier, for example, by permitting users to opt for an audio or sound CAPTCHA.

Image Security: CAPTCHA images of text should be distorted randomly before being presented to the user. Many implementations of CAPTCHAs use undistorted text, or text with only minor distortions. These implementations are vulnerable to simple automated attacks.

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:

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

(2) Systems where a solution to the same CAPTCHA can be used multiple times (this makes the CAPTCHA vulnerable to so-called "replay attacks").

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (314-319), Month: July - September 2015, Available at: www.researchpublish.com

Most CAPTCHA scripts found freely on the Web are vulnerable to these types of attacks.

Security Even After Wide-Spread Adoption. There are various "CAPTCHAs" that would be insecure if a significant number of sites started using them. An example of such a puzzle is asking text-based questions, such as a mathematical question ("what is 1+1"). 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.

Fig.4 CAPTCHA generation

B. The Algorithm:

The algorithm used to create the CAPTCHA does not need to be made public, though it may be covered by a patent. Although publication can help demonstrate that breaking it requires the solution to a difficult problem in the field of artificial intelligence, deliberate withholding of the algorithm can increase the integrity of a limited set of systems, as in the practice of security through obscurity. The most important factor in deciding whether an algorithm should be made open or restricted is the size of the system.

Although an algorithm which survives scrutiny by security experts may be assumed to be more conceptually secure than an unevaluated algorithm, an unevaluated algorithm specific to a very limited set of systems is always of less interest to those engaging in automated abuse. Breaking a CAPTCHA generally requires some effort specific to that particular CAPTCHA implementation, and an abuser may decide that the benefit granted by automated bypass is negated by the effort required to engage in abuse of that system in the first place.

Functions

- Encryption
- Image Processing

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (314-319), Month: July - September 2015, Available at: www.researchpublish.com

- Base 64 Coder
- Configuration

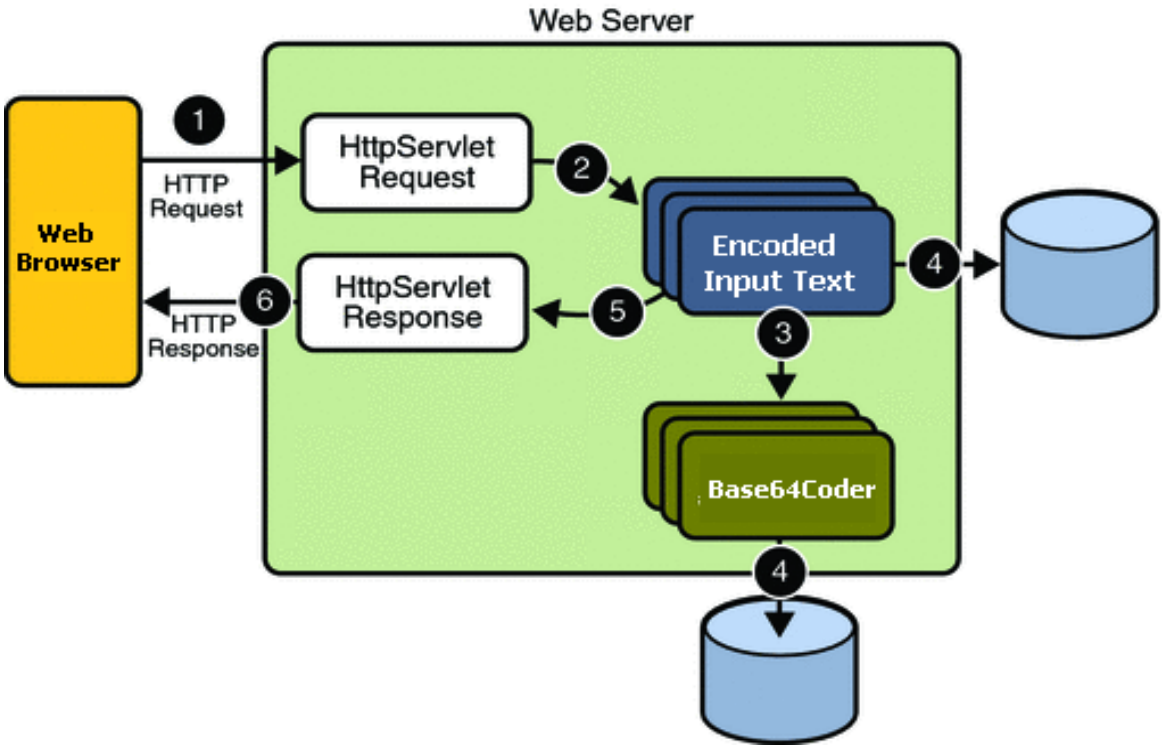

Fig.5 Architecture

IV. APPLICATIONS

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 Website Registration:

Several companies (Yahoo!, Microsoft, etc.) offer 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 in search of 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:

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:

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 unindexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, doesn't guarantee that bots won't 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 don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, CAPTCHAs are needed..

V. CONCLUSION

The Reusable CAPTCHA Security Engine is mainly aimed at developing better CAPTCHAs. The best CAPTCHA would allow all human to pass through, while rejecting all machines. We would like to test these CAPTCHAs and invite both users and bots to attack them.

REFERENCES

[1] Core Java™ 2 Volume I – Fundamentals 7th Edition.

[2] Pearson Education – Sun Microsystems.

[3] Core Java™ 2 Volume II – Advanced .

[4] Pearson Education – Sun Microsystems.

[5] Head First Servlets & JSP.

[6] O'Reilly – SPD.

[7] The Book of JavaScript 2nd Edition

[8] Effective Java – Programming Language Guide.

[9] Pearson Education – Sun Microsystems.

[10] JBoss – A Developers Notebook.

[11] O'Reilly – SPD.