Securing Websites through Multi-CAPTCHA

[1] Mr. Shaik Muhammed Manzoor, [2] Mrs. Kumari R Soumya

[1] Department of Computer Science and Engineering
Jyothi Engineering College, Cheruthuruthy, Thrissur, India.

[2] Asst. Professor, Department of Computer Science and Engineering
Jyothi Engineering College, Cheruthuruthy, Thrissur, India.

Abstract - CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) is a simple test that is easy for humans but extremely difficult for computers to solve. CAPTCHA has been widely used in commercial websites such as web-based email providers, TicketMaster, GoDaddy, and Facebook to protect their resources from attacks initiated by automatic scripts. By design, CAPTCHA is unable to distinguish between a human attacker and a legitimate human user. This leaves websites using CAPTCHA vulnerable to 3rd party human CAPTCHA attacks. In order to demonstrate the vulnerabilities in existing CAPTCHA technologies we develop a new streamlined human-based CAPTCHA attack that uses Instant Messenger infrastructure. Facing this serious human-based attack threat, we then present a new defense system called Multi-CAPTCHA , which is the next generation of CAPTCHA technology providing the first steps toward defending against 3rd party human CAPTCHA attacks. Multi-CAPTCHA requires a user to solve a CAPTCHA test via a series of user interactions. The multi-step back-and-forth traffic between client and server amplifies the statistical timing difference between a legitimate user and a human solver, which enables better attack detection performance

***Keywords* - CAPTCHA, Website Securing, Human Factors, Network Security.**

1. Introduction

The mechanism for using randomly generated images containing words or characters for human-user validation was developed by Alta Vista in the late 1990's [1]. The term CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) was coined in 2000 by Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford [1]. CAPTCHAs were designed to generate tests that distinguish humans from malicious computer programs. Today, CAPTCHA technology is widely used to defend against scripted registrations in web-based services such as web-based email accounts. Despite their widespread use, CAPTCHAs are not foolproof. CATPCHAs can and have been broken consistently. Besides the primary attack method using image processing to decode CAPTCHA tests, recently techniques have been developed for utilizing a 3rd party human user to break given CAPCHAs. This type of attack is particularly difficult to prevent, and our research shows that no effective solutions currently exist. One of the primary aims of this paper is to present a novel and effective method for coping with this security challenge.

A good CAPTCHA must not only be human friendly but also robust enough to resist computer programs that attackers/hackers write to automatically pass CAPTCHA tests. CAPTCHA resist the automatic registration over the internet and provides the smooth registration process. In order to verify that registration request is submitted by individual user from online rather than malicious software the academia proposed CAPTCHA technology

2. Related work

The first CAPTCHAs were based on images of distorted characters, which is still the dominant approach adopted. Users have to read the characters and enter them into a text field. An increasing amount of distortion has been added to these CAPTCHAs to combat the rapid evolution of image processing and more specifically optical character recognition (OCR) algorithms. Approaches discussed in literature demonstrate that this category of CAPTCHAs is nonetheless legible to computers. Thus, alternatives to this method have been proposed.

Text based CAPTCHAs provide puzzles as plain text. This class of CAPTCHAs includes solving of simple mathematical expressions or completing sentences. A problem with the latter method is that clever algorithms can predict the likelihood of a given word occurring in a sentence, given the two previous words as context. Also, no OCR processing is required to analyze this class of CAPTCHAs.

Another category of CAPTCHAs displays a set of images together with a task to recognize and point to a certain image or subset of images. This class of CAPTCHA has a number of issues. If a user has to recognize a single image only, the success probability of random guessing becomes too high, while, the selection of multiple images places a high burden on the user. Finally, assembling a sufficiently large image database is not trivial, considering copyright issues.

The major concern with CAPTCHAs from a usability perspective is that most of the present schemes are not accessible for humans who are blind, visually impaired or have a learning disability. It is a well-known fact that under distortion some characters such as ' 1' and 'l', 'O' and '0', and '5' and 'S' have a high potential for confusion. This effect is further amplified by the font face. To increase the effectiveness of CAPTCHAs against OCR-like attacks, Google and Yahoo! CAPTCHAs have created new confusing characters, such as 'vv' and 'w', 'cl' and 'd', and 'rn' and 'm'.

Major usability aspects which have not been investigated so far include the amount of time a user requires to solve a CAPTCHA tasks as well as the complexity of the task. These aspects are key when evaluating the applicability, cost and acceptance of a novel technology from a user perspective.

3. Multi-Captcha

Facing with the growing threat of 3rd party human attacks on existing CAPTCHA technologies, the industry is in need of a reliable defense technique. This section describes our proposed multi-CAPTCHA as the first and initial step towards defending against this type of attack.

Multi-CAPTCHA utilizes a sequence of mouse clicks to allow a user to interactively solve a CAPTCHA challenge. First, a normal CAPTCHA image is dynamically generated and displayed. The user clicks on the CAPTCHA image to begin the multi-CAPTCHA input sequence. Upon clicking on the CAPTCHA image, several buttons with obfuscated characters appear below the CAPTCHA image. This update is performed via an asynchronous JavaScript (Ajax) [11] request to the server that is rendered back to the user's web browser without refreshing the whole web page. Once the set of character buttons is displayed, the user must click on the button corresponding to the first character in the CAPTCHA image. Upon each click, a new set of buttons is rendered. This input sequence continues until one click has been performed for each character of the CAPTCHA image.

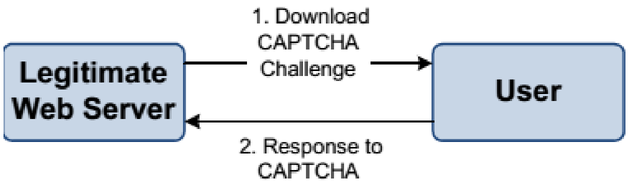


Fig 1: User-Interaction with multi-CAPTCH

On the server side, session information is stored about the indices of the correct responses and the indices of the user clicks. When the input sequence is complete, the correct index sequence is then compared with the user clicked index sequence. If there is a match, the CAPTCHA has been correctly decoded by the user.

Multi-CAPTCHA measures the time it takes for a user to respond on a per-character basis. This allows the timeout value for multi-CAPTCHA to be enforced for each character input. Therefore, the per-character timeout for multi-CAPTCHA can be set much lower than the timeout value for a standard CAPTCHA. This provides a much greater resolution in determining human attacks because the relative time between each input and the time it takes to send the CAPTCHA to a human solver is minuscule. Additionally, multi-CAPTCHA allows users to take as much time as needed to decode the image first before starting the multi-step challenge/response sequence. Due to this separation of decoding and actions, subsequent interactive steps are expected to be swift.

4. Implementation

We implemented this project in Java. We use Apache WEB Server to host the website and develop JSP pages with the help Netbeans 7.4 IDE. We have taken few fields in the form just to show the impression of the registration form. The CAPTCHA is shown in Fig3.

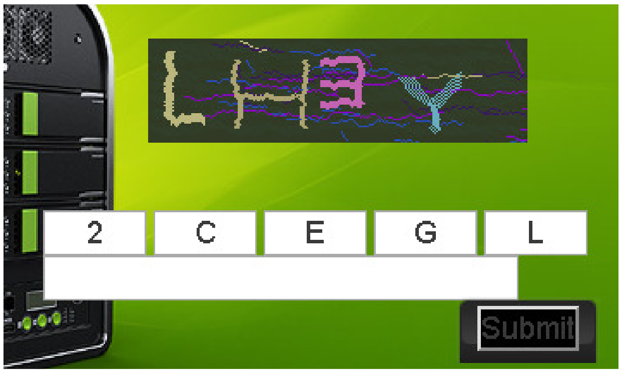
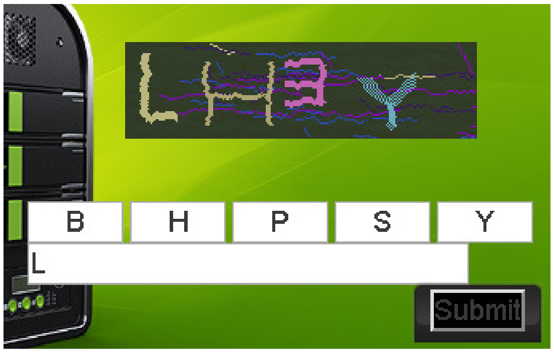


Fig 3: Initial CAPTCHA

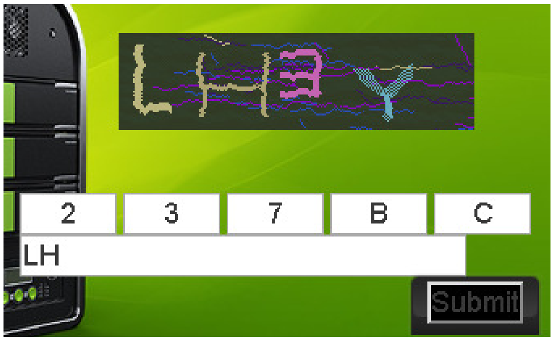
The multi-CAPTCHA is loaded and waiting for user response. Here we use mouse clicks to register user responses. The variables that to be clicked for CAPTCHA verification is generated randomly. User need to click the correct variable by looking to the CAPTCHA image. After clicking the correct alphanumeric, the listed variables randomly changes as shown in Figure 2.



L

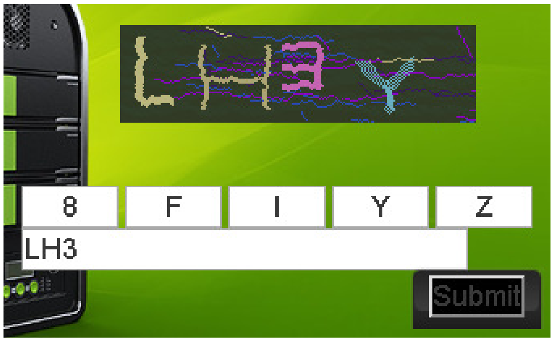
Fig 4: Characters changed after clicking L

This iteration repeats until user types all the four letters. The sequence is shown in Figure 3 to 5.



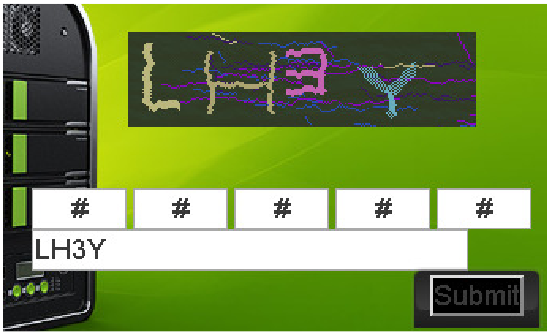
LH

Fig 5: Characters changed after clicking H



LH3

Fig 6: Characters changed after clicking 3



LH3Y

Fig 7: Characters changed after clicking Y

This will improve security since the characters are changing instantaneously after each click. Per-character-response time for a legitimate user interaction with multi-CAPTCHA is calculated as shown below:

$$\mathbf{R_u = t_1 + t_2 + U \quad (1)}$$

Where:

- R_u: the total response time for a single character in the legitimate user scenario
- t_1: network time to download and view the CAPTCHA and obfuscated characters
- t_2: network time to submit HTML post to the web server

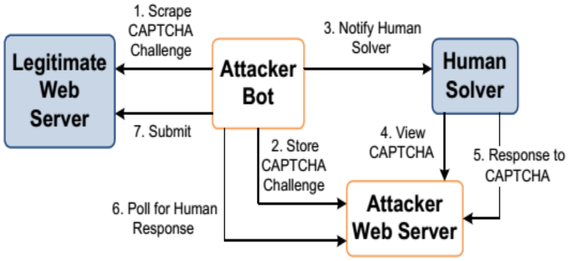


Fig 8: Attack-Bot's interaction with multi-CAPTCHA

Figure 8 depicts the complicated process of forwarding multi-CAPTCHA to a 3rd party human solver. In the first iteration, the attack script notifies the human solver of the multi-CAPTCHA location, which is served on an attacker's web server. For subsequent iterations, the Attacker-Bot proxies the challenge/response inputs back and forth with the target website.The human solver have to respond to the multi-CAPTCHA by clicking on the corresponding character button. At this point, the attacker's web server has the response information, but it cannot immediately deliver it to the attacker bot. The attacking script has to poll for the result. Depending on the polling interval and resource available, step 6 adds additional delay time to the process. It should be noted that each set of buttons is only sent out by the server after the server receives the click response for the previous set of buttons. Therefore, an attacker bot has to relay the sequence of challenges and responses between the target server and the 3rd party human solver. The replay process takes additional times beyond the network delay between attacker bot and the 3rd party human solver. The total per-character-response time for an attacker can be expressed as follow:

$$\mathbf{R_a = t_1 + t_2 + t_4 + t_5 + t_6 + t_7 + U \quad (2)}$$

Where:

- R_a: the total response time for a single character in the 3rd party human attacker user scenario
- t_1: network time to download and scrape the CAPTCHA and obfuscated characters
- t_2: network time to upload or ftp CAPTCHA image and obfuscated characters to attacker's web server
- t_4: delay to download and view the CAPTCHA and obfuscated characters

- t_5: network time to submit HTML post to the web server
- t_6: the polling time for response
- t_7: network time to submit HTML post to the web server
- U: time for human user to decode and click on the corresponding character.

By comparing Equations (1) and (2), it is clear that a 3rd party human solver attack adds 4 additional time delays in solving a single CAPTCHA character in the proposed multi-CAPTCHA implementation. Since a typical R_u (legitimate user per-character response time) is on the order of 1-2 seconds, the added delay time in an attack scenario would be significant. This additional delay time allows multi-CAPTCHA to be effective in detecting and stopping 3rd party human attacks.

5. Conclusion

CAPTCHA plays an important role in protecting Internet resources from attacks by automated scripts. However, CAPTCHA is believed to be vulnerable to 3rd party human attacks due to the nature of its design. We then proposed the novel multi-CAPTCHA system which provides simple yet effective defense against 3rd party human solver attacks. The multi-step back-and-forth traffic between client and server amplifies the statistical timing difference between a legitimate user and a human solver attack, and hence, provides a better attack detection performance. As the first step towards defending against the growing threat of 3rd party human CAPTCHA attacks, we hope that the proposed multi-CAPTCHA system will encourage researchers and the security industry to develop more secure and reliable CAPTCHAs.

References

[1] Luis von Ahn , Manuel Blum , John Langford, “Telling humans and computers apart automatically”, Communications of the ACM, v.47 n.2, p.56-60, February .

[2] G. Mori and J. Malik. “Recognizing objects in adversarial clutter – breaking a visual CAPTCHA.” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, Jun. 18-20, 2003.

[3] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion Estimation Techniques in Solving Visual CAPTCHAs", Proc. IEEE CVPR, 2004 .

[4] Jeff Yan, Ahmad Salah El Ahmad, “A Low-Cost Attack on a Microsoft CAPTCHA” Proceedings of the 15th ACM conference on Computer and communications security, 2008.

[5] Elias Athanasopoulos, Spyros Antonatos: “Enhanced CAPTCHAs: Using Animation to Tell Humans and Computers Apart.” Communications and Multimedia Security 2006: 97-108

[6] Sam Hocevar, “PWNtcha–Captcha Decoder Website”, http://caca.zoy.org/wiki/PWNtcha

[7] Dancho Danchev,“Inside India’s CAPTCHA Solving Economy”, http://blogs.zdnet.com/security/ p=1835, 2008

[8] Albert E. Whale, “ABS computer technology, inc, Why the CAPTCHA Approach is Doomed”, http://www.abscomptech.com/home/headlines/news/why-the-CAPTCHA-approach-isdoomed , 2009

[9] Wumpus1, “A CAPTCHA Server Control for ASP.NET”,http://www.codeproject.com/KB/custom-controls/CaptchaControl.aspx

[10] Byron Acohido, “Cybergangs use cheap labor to break codes on social sites”, http://www.usatoday .com/tech/news/computersecurity/2009-04-22-captcha-code-breakers_N.htm, 20092003

[11] J.J. Garrett, "Ajax: A New Approach to Web Applications”, Adaptive Path Assays, 2005. http://www.adaptivepath.com/ideas/essays/archives/000385.php Y. Wang, Y. Xiang, J. Zhang, and S.-Z. Yu, “A novel semi-supervised approach for network and security. Especially in network traffic classification.