



Position Based CAPTCHA: Changing Place Restriction Minimize the Automatic Access

Er. Vivek Kumar¹*M.Tech, Computer Science and Engg.
Kanpur Institute of Technology, Kanpur, India*Er. Prem Shanker Yadava²*Assitt. Prof., Information Technology
Kanpur Institute of Technology, Kanpur, India*

Abstract— At present day, most people use internet sites to get any type of information and resources and approximately all internet users have login accounts for internet sites to use their resources. For login accounts user must be registered on internet sites and these internet sites requires only registration by human users but unfortunately some automated computer programs enter these sites and use their resources through false registration. This paper introduces RANDOM CAPTCHA. The proposed RANDOM CAPTCHA scheme increases the strength of CAPTCHA and provides strong security to websites from automated computer programs. In this paper we target to change the place randomly of displaying CAPTCHA and CAPTCHA automatically after a predefined fixed time (few seconds) again and again without refreshing whole webpage. CAPTCHA refreshes code and changes place randomly itself until the final CAPTCHA is filled by user. Now, automated computer programs have more difficulties to break the CAPTCHA. The proposed RANDOM CAPTCHA scheme restricts automated computer programs to illegal access to any resource.

Keywords: CAPTCHA, Time-Variant CAPTCHA, Random Number Generator, System Clock, ASP.NET

I. INTRODUCTION

CAPTCHA stands for “Completely Automatic Public Turing test to tell Computers and Humans Apart”, as in [1]. CAPTCHA is a program which can generate and grade the tests that it itself and other automated computer program cannot pass and most human can pass. The security aspect of CAPTCHA should be such that no computer program should be able to pass the tests generated by it even if the knowledge of the exact working of the CAPTCHA is known, as in [3]. In the recent past several companies have suffered from the attacks of BOTS (Computer Automated Software Programs) wherein thousands of dummy email accounts were signed-up every minute. To overcome this situation the use of CAPTCHA was introduced wherein the test requires the users to prove that they are human beings before they can get a free email account. CAPTCHAs are introduced as effective tools to restrict the attack from BOTS.

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 scheme, as in [1,2]. In this paper we are presenting a RANDOM CAPTCHA. We implemented methods that change place of displaying CAPTCHA and its code at various places over webpage after predefined fixed time again and again without loading or refreshing whole webpage. RANDOM CAPTCHA changes code and place automatically many times until user submits the final CAPTCHA. It restricts BOTS to illegal access to any resources.

II. LITERATURE SURVEY

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, as in [4].



Fig. 1: Pessimal print CAPTCHA

CAPTCHAs are classified based on their content, such as:

Text Based CAPTCHA, it consists of an image with 5 to 6 or more characters in it. The image will have some noise applied to it like horizontal or vertical or cross lines is added in background or characters are surrounded by dots. Examples of Text based CAPTCHA are Gimpy as in [5], EZ-Gimpy and Baffle Text. Image Based CAPTCHA, a database is maintained for various images containing pictures of people, animals, cars etc. Randomly five or more images are picked up so that there is common object in them and this common object is to be identified by the users. Examples of Image based CAPTCHA are 3D-based, pix, Bongo and Ironclad etc.

Audio Based CAPTCHA, user has to listen the distorted audio offered by CAPTCHA system and then user has to enter the text of audio in the space provided. Example of Audio based CAPTCHA is ECO.

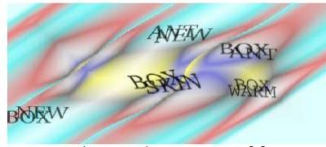


Fig. 2: Gimpy



Fig. 3: EZ-Gimpy



Fig. 4: Baffle Text



What are these Images of?

Fig. 5: Image Based

Captcha2 is a whole new method of human verification. One you won't hate. In fact, it's kind of like a game. Instead of decoding impossible-to-read distorted text and having to type it in, need to just click two times. First click on a letter given in CAPTCHA image proceeds the new CAPTCHA image, click again on the correct letter complete the process, as in [6]. Time-Variant CAPTCHA display various CAPTCHA for a predefined time interval i.e. CAPTCHA will be refresh after a predefined time. CAPTCHA replaces itself until the final CAPTCHA is filled by user, as in [7].

III. PROPOSED WORK

The security of CAPTCHA should be very strong and very difficult to break. CAPTCHA will need fast, efficient and intelligent algorithms. BOTS have enough time to scan and apply other processes to break the CAPTCHA between the duration of form load to form submit. So here we will implement a new methodology for CAPTCHA. In this paper we introduce RANDOM CAPTCHA. The proposed RANDOM CAPTCHA will be increase strength of CAPTCHA and provides strong security to websites from BOTS. In this paper we are not focusing on the effective development of CAPTCHA but we implement methods that will change place of displaying CAPTCHA and its code automatically at various places over webpage after a fixed time (few seconds, predefined by developer) again and again without loading or refreshing whole webpage. CAPTCHA refresh and changing its random place process just work with the CAPTCHA and it don't affect webpage. In registration process user fills all required fields before submit CAPTCHA, during this period RANDOM CAPTCHA changes its code and places randomly many times until user fill the final CAPTCHA and submit form.

A. Concept Preparation

Our work is to implement the methods that change the place of CAPTCHA automatically and regularly without loading or refreshing whole webpage. CAPTCHA refreshes and changes its place randomly. In this paper we are not dealing the creation aspects to generate strong CAPTCHA. Normally in registration process user fills CAPTCHA and press submit button. So CAPTCHA is programmed with button in webpage. User fills all required entries before submit form. Hacker or BOTS have enough time between page loading and page submission to break CAPTCHA. We can write the following action:

$$(CSC)_B \leftarrow T_{time} ((WPF)_F) \quad \text{---- (1)}$$

$(CSC)_B$ = CAPTCHA Security Checking on submit button

$(WPF)_F$ = Web Page Form is filled completely

T_{time} = Total time to complete fill form

T_{time} is enough for attacker to know the pattern of CAPTCHA because CAPTCHA has fixed position and code. So we implement a new methodology in programming part of CAPTCHA to change its places randomly within a time spam.

B. Algorithm to change place and code of CAPTCHA with Time-Variant

Here we have some algorithm steps against equation (1), to change place randomly and code of CAPTCHA with time variant system.

Algorithm to change place and code of CAPTCHA with Time-Variant

1. Count $\leftarrow (S_T)_{page\ load}$
 2. Temp \leftarrow Count + T_P
 3. Change $((Random_Captcha)_{place}) \leftarrow$ Temp
 4. Refresh $((Captcha)_{code}) \leftarrow$ Temp
 5. Change $((Random_Captcha)_{place}) \leftarrow$ Refresh $((Captcha)_{code})$
 6. $T_{time} \leftarrow$ Loop (Temp + T_P)
 7. $(WPF)_F \leftarrow T_{time}$
 8. $(CSC)_B \leftarrow T_{time}$
 9. $(CSC)_B \leftarrow (WPF)_F$
-

$(S_T)_{page\ load}$ = System Time on page load

Temp = Temporary variable for time-variant system

T_P = Predefined Time period

Change $((Random_Captcha)_{place})$ = Captcha change place randomly at each repetition

Refresh $((Captcha)_{code})$ = Captcha is refreshed at each repetition

Loop = Repetition till form filled

$(CSC)_B$ = CAPTCHA Security Checking on submit button

$(WPF)_F$ = Web Page Form is filled completely

T_{time} = Total time to complete fill form

In equation (1), attackers or hackers have T_{time} to break CAPTCHA because it has fixed place and code where as in new concept there is (Temp + T_P) which is very small time for displaying CAPTCHA and its place because after (Temp + T_P) time CAPTCHA will change its place. Now, this concept provides more security to CAPTCHA.

C. Algorithms

The following algorithms are used to change place of CAPTCHA randomly after predefined time without loading or refreshing webpage:

1) Algorithms for Time-Variant (System Clock)

It describes relation between CAPTCHA and system time.

1. $S_T \leftarrow$ fetches System Time
 2. Temp $\leftarrow S_T$
 3. $T =$ Temp + T_P
 4. Condition: change place randomly and code of CAPTCHA after each time T_P
 5. Repeat step 2 and 3
-

S_T = System Time

Temp = Temporary variable

T_P = Predefine Time Period

T = Temporary variable for Time-Variant system

2) Algorithm for Changing Place Randomly

To search and capture any CAPTCHA is very tuff, if CAPTCHA changes its place. Here we are giving algorithm to present and to answer the question, "how CAPTCHA changes its place". We are displaying CAPTCHA in different places in a defined area rather than anywhere in the display screen. It is only to prevent the hesitating situation of users.

1. Define places for CAPTCHA
i.e. $(\text{Captcha})_{\text{place}} = [1,2,3\dots n]$
(for e.g. we take $n=5$ in our project, so $(\text{Captcha})_{\text{place}} = [1,2,3,4,5]$)
2. Use random number generator and set number limit as $(\text{Captcha})_{\text{place}}$
i.e. $(\text{Random})_{\text{number}} \leftarrow (\text{Captcha})_{\text{place}}$
(for e.g. $(\text{Random})_{\text{number}} = [1,2,3,4,5]$)
3. Generate random number from $(\text{Random})_{\text{number}}$
i.e. $(\text{Generated})_{\text{number}} \leftarrow (\text{Random})_{\text{number}}$
(for e.g. Generated number = 3 from $[1,2,3,4,5]$)
4. If $(\text{Generated})_{\text{number}} \in (\text{Captcha})_{\text{place}}$
(means $3 \in [1,2,3,4,5]$)
5. Then $(\text{Random_Captcha})_{\text{place}} \leftarrow (\text{Generated})_{\text{number}}$
(i.e. new random place of captcha = 3)
6. Repeat step 3,4 and 5 until form submit

$(\text{Captcha})_{\text{place}}$ = Places for CAPTCHA to be display

$(\text{Random})_{\text{number}}$ = Random Number Generator

$(\text{Generated})_{\text{number}}$ = Generated Random Number

$(\text{Random_Captcha})_{\text{place}}$ = Random Place of CAPTCHA to be display

3) Algorithm for Submit Button

1. Click on Submit Button
2. If any required field is empty then show a warning message box
3. Else if
4. CAPTCHA matched then go to step 8
5. Else
6. new CAPTCHA generate and matched then go to step 8
7. Else repeat step 6
8. End ; Registration Complete

IV. RESULT

We implemented proposed work in ASP.NET and get effective results. We see the form with CAPTCHA shown in Fig. 6. We explored webpage to registration. Fig. 6 shows snapshot at page load. Since after predefined time (15 seconds) CAPTCHA had changed its place and its code while we did not completely fill the registration form which shown in Fig. 7.

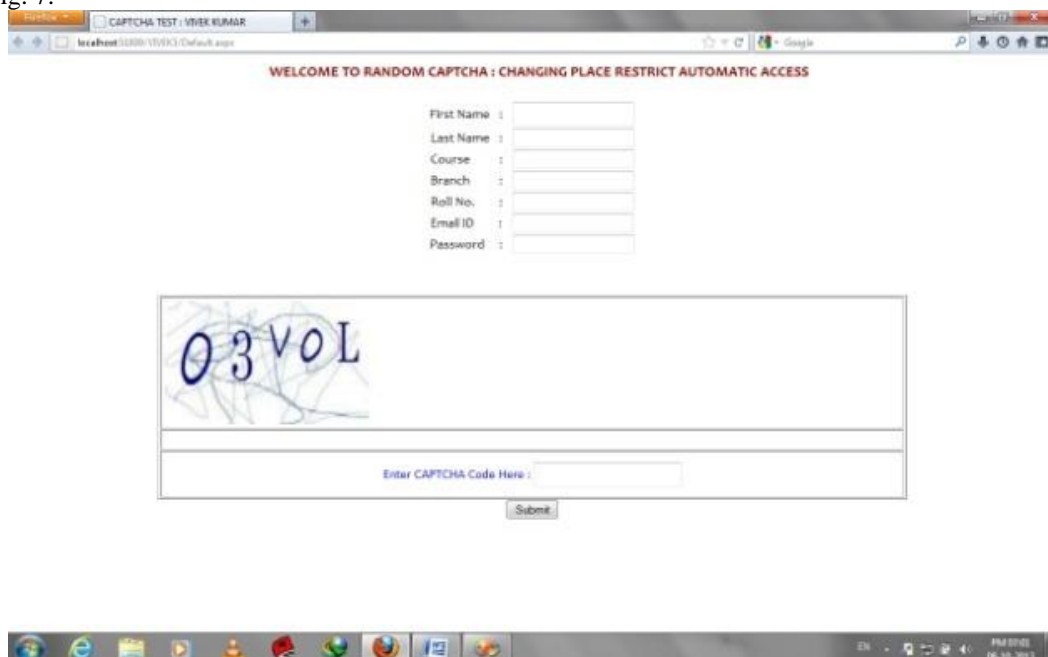


Fig. 6: CAPTCHA on page load

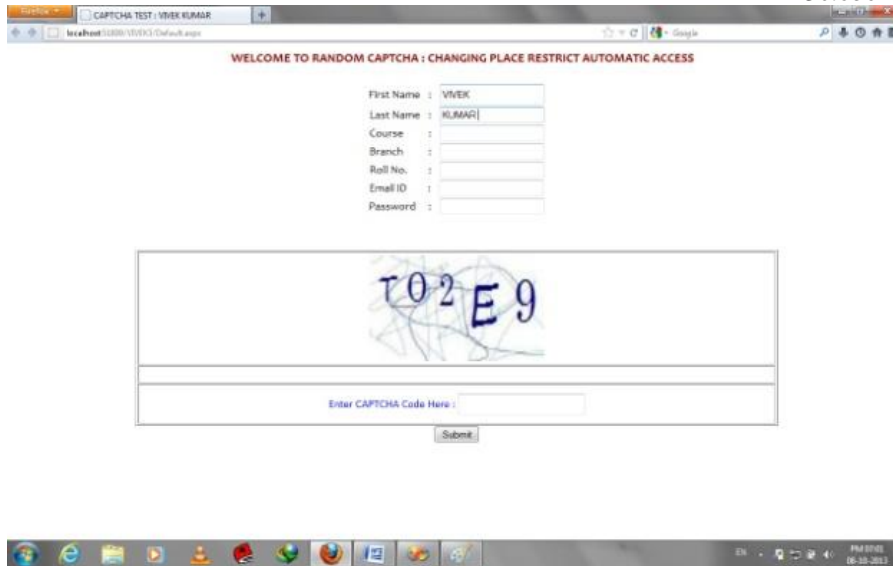


Fig. 7: CAPTCHA changed its place and code after 15 second

Fig. 8 shows that form is not completed and CAPTCHA changed place randomly and its code without loading or refreshing webpage after predefined time and this process don't affect webpage. If CAPTCHA is scanned and calculated by BOTS, new CAPTCHA wastes all calculation. Fig. 9 shows that registration process complete after submit correct CAPTCHA by human user.

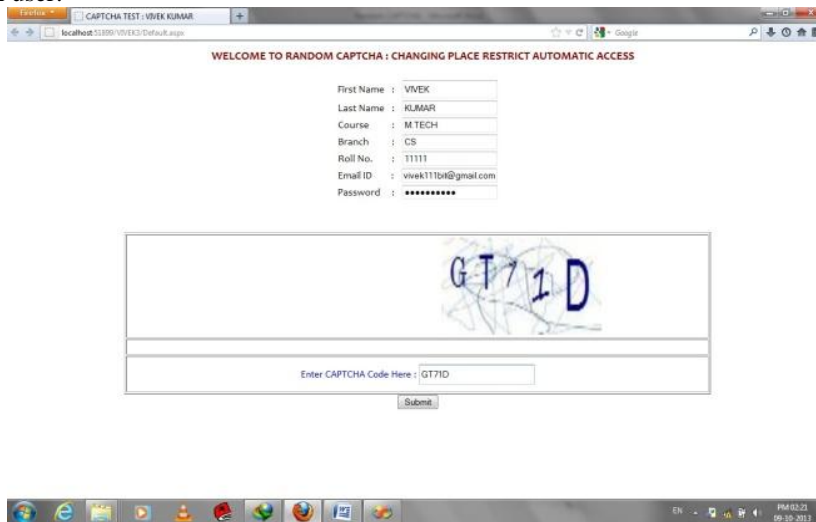


Fig. 8: CAPTCHA changed its place and code while form is not complete and don't affect

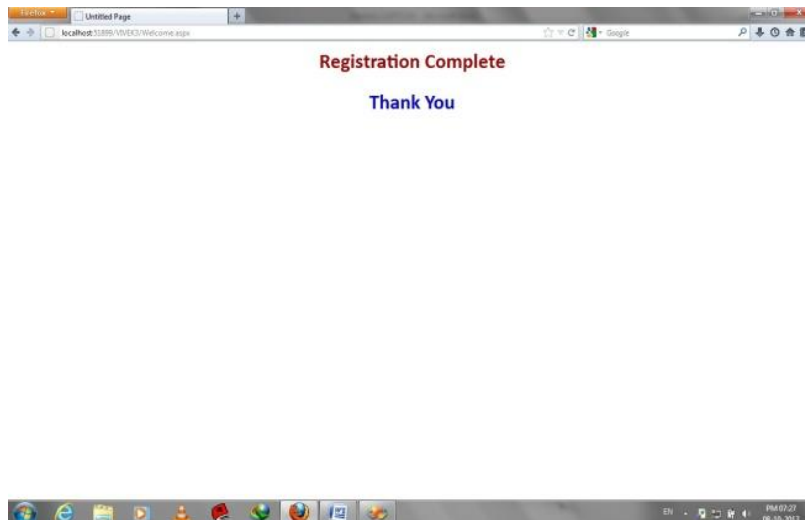


Fig. 9: CAPTCHA submit and registration complete

V. CONCLUSIONS

In this paper we introduced RANDOM CAPTHCA, which changed its place and code automatically after predefined time without loading or refreshing web page. This methodology increased strength of CAPTCHA and provides strong security to websites from automated computer programs. Only human can pass RANDOM CAPTCHA test. This project is very effective, efficient, and practical tool for CAPTCHA. RANDOM CAPTCHA implementation is based on the sharing of system clock, we can change system clock concept by number counting calculation. This method is very useful for websites because it proves strong security to websites from BOTS.

REFERENCES

- [1]. L. von Ahn, M. Blum, N. Hopper, and J. Langford. CAPTCHA: Using Hard AI problems for Security in Advances in Cryptology EUROCRYPT 2003, page 646. DOI: 10.1007/3-540-39200-9_18
- [2]. L. von Ahn, M. Blum, and J. Langford, "Telling Human and Computers Apart Automatically," in Communications of the ACM, vol. 47, February 2004, no. 2, pp. 57-60. DOI: 10.1145/966389.966390
- [3]. A.A.Chandavle , Dr.A.M. Sapkal and Dr.R.M.Jalnekar (2010) A framework to analyze security of Text based CAPTCHA International Journal of Forensics and Computer Application ,1 (27) , 127-132
- [4]. Allison L. Coates, Henry S. Baird, Richard J. Fateman."Pessimal Print: A Reverse Turing Test", Proceedings, IAPR 6th Int'l Conf. On document Analysis and Recognition, Seattle, WA, September 10-13, 2001, pp.1145-1158
- [5]. M. Blum, L. von Ahn, and J. Langford, The CAPTCHA Project (Completely Automated Public Turing Test to Tell Computers and Humans Apart), School of Computer Science, Carnegie- Mellon University, November 2000
- [6]. <http://captcha2.com>
- [7]. Time-Variant Captcha: Generating Strong Captcha Security by Reducing Time to Automated Computer Programs Prem Shanker Yadava , Chandra Prakash Sahu , Sanjeev Kumar Shukla VOL. 2, NO. 12, December 2011. ISSN 2079-8407