IJERMT

International Journal of Engineering Research & Management Technology ISSN: 2348-4039

November-2015 Volume 2, Issue-6

Email: editor@ijermt.org www.ijermt.org

Analysis of Various Parameters on a Novel Custom Captcha Security Method

Manish Kumar
Information Technology
G.B.P.U.A.&T.Pantnagar

Rajesh Shyam Singh
Asstt. Professor
Department of Information Technology
G.B.P.U.A.&T.Pantnagar

Hardwari Lal Mandoria
Professor & Head,
Department of Information Technology
G.B.P.U.A.&T.Pantnagar

ABSTRACT:

Many a time's excessive distortion is applied to prevent recognition which renders the image unreadable for humans too. Earlier CAPTCHA mechanisms lack in security in respect to some or the other aspect of attacks. Some can prevent segmentation based attack but dictionary attack was possible and vice-versa. Most of the CAPTCHA mechanisms uses a single parameter as security to prevent recognition for example most systems use distortion for prevention of attacks but other methods should also be considered. A balance between the difficulty of CAPTCHA for bots and its readability for humans is to be maintained. This paper describes a custom CAPTCHA mechanism which prevents any attacks that were possible in early systems of CAPTCHAs.

KEYWORDS: Captcha, Captcha Security, Recaptcha, Turing Test, Cyber Security

I. INTRODUCTION TO CAPTCHA SECURITY:

The acronym stands for Completely Automated Public Turing-test to tell Computers and Humans Apart (Ahn, L. V. et.al.). In simpler terms a CAPTCHA is a security mechanism that is used to prevent autonomous entries on websites (or prevent computerised bots) from doing unlawful activities on web pages.

The widely used CAPTCHA schemes use combinations of distorted characters and obfuscation techniques that humans are able to recognize but it is difficult enough for automated scripts. CAPTCHAs are also called "Reverse Turing Tests": because they are intended to allow a computer to determine if a remote client is human or not (Bursztein E. et. al.). As time has passed bots have evolved into newer mechanisms to find loopholes in the existing CAPTCHA systems. Therefore, CAPTCHAs must be re-engineered time-to- time to prevent these attacks.

II. THE NEED FOR CAPTCHA SECURITY:

With the massive amounts of spam (unsolicited, junk email) being delivered to inboxes around the world, people have been concerned with publishing their email address online. Spam bots can process thousands of web pages per hour, scanning for email addresses in clear text. Many people have resorted to attempting to fool spam bots by posting their email addresses in a human readable but unconventional form.

III. PROPOSED APPROACH:

This section discusses about the various generated CAPTCHA challenges which were obtained after applying various customizations that are possible in this new approach and also by variation in the parameter values.

A. CHANGING THE DISTORTION LEVEL:

The distortion level refers to the perturbation or the deviation of CAPTCHA characters. Based on its value the level of warping can be adjusted.

a. DISTORTION LEVEL = 0

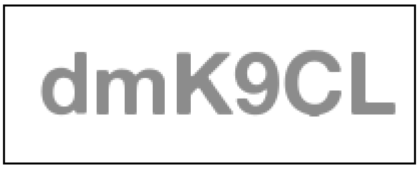

Figure 4.1: CAPTCHA obtained after setting the distortion level to 0

As it can be observed in the image, setting the distortion level to 0 renders the CAPTCHA characters unchanged and no distortion is applied.

b. DISTORTION LEVEL = 1

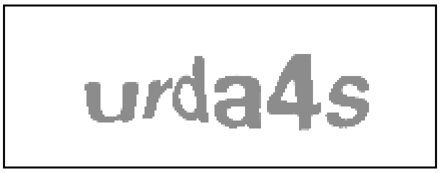

Figure 4.2: CAPTCHA obtained after setting the distortion level to 1

Setting the distortion level to 1 distorts the image to some extent but it may still be read by some bots.

c. DISTORTION LEVEL = 1.5

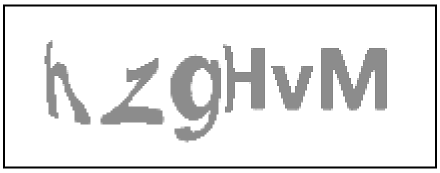

Figure 4.3: CAPTCHA obtained after setting the distortion level to 1.5

After setting the distortion value to 1.5 an optimal level of distortion is achieved rendering the image easy enough to be read by humans but tougher for bots.

B. CHANGING NOISE LEVELS:

The Noise Level parameter adds some random noise in the form of dots to confuse the bots from reading the CAPTCHA image but the noise doesn’t prove as a hindrance to the user.

a. NOISE LEVEL = 1

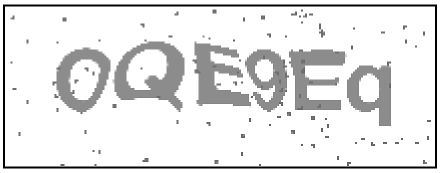

Figure 4.4: CAPTCHA obtained after setting the Noise level to 1

After setting the Noise Level to 1 it can be seen that some random black dots are visible over the CAPTCHA image but they are not enough to prevent bots.

b. NOISE LEVEL = 5

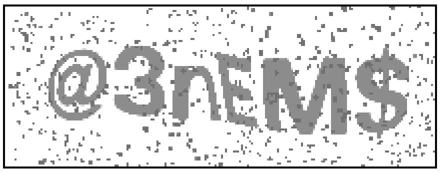

Figure 4.5: CAPTCHA obtained after setting the Noise level to 5

After setting the Noise Level to 5 it can be seen that some more random black dots are visible over the CAPTCHA image.

c. NOISE LEVEL = 10

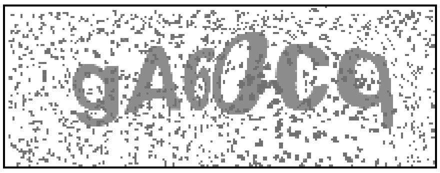

Figure 4.6: CAPTCHA obtained after setting the Noise level to 10

After setting the Noise Level to 10 it can be seen that enough noise is added to prevent bots by confusing the characters of the CAPTCHA image.

C. CHANGING TRANSPARENCY LEVELS:

The transparency level helps to control the transparency of the CAPTCHA characters. The values that can be specified range from 0 to 100 where 0 means that the characters are completely opaque and 100 means they will be completely transparent i.e. not visible. Hence, transparency level must be kept somewhere in between.

a. TEXT TRANSPARENCY LEVEL = 0

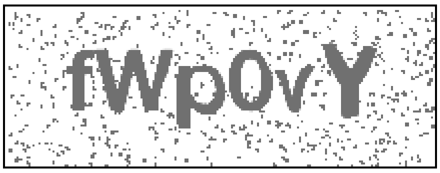

Figure 4.7: CAPTCHA obtained after setting the Transparency level to 0

Keeping the transparency level to 0 renders the characters opaque.

b. TEXT TRANSPARENCY LEVEL = 30

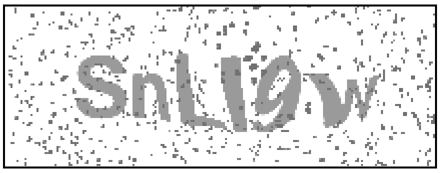

Figure 4.8: CAPTCHA obtained after setting the Transparency level to 30

Keeping the transparency level to 30 applies slight transparency to the image and it may be judged as an optimal value.

c. TEXT TRANSPARENCY LEVEL = 50

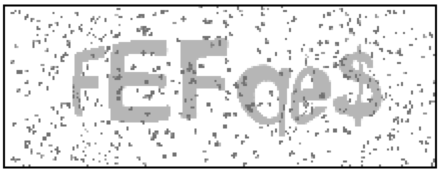

Figure 4.9: CAPTCHA obtained after setting the Transparency level to 50

Keeping the transparency level to 50 applies more transparency to the image and may appear blurry to some users hence it is not recommended.

D. ADDING SIGNATURE IMAGE

A signature is yet another customization that can be applied in the proposed CAPTCHA system. A signature image helps in verifying the original author of the image. It can also be used to resolve copyright issues.

a. USING SIGNATURE = MANISH

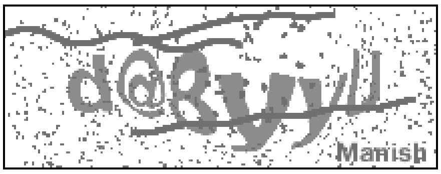

Figure 4.10: CAPTCHA obtained after setting the Signature Value= Manish

Using the text 'Manish' as a signature renders the above displayed CAPTCHA.

b. USING SIGNATURE = MNSH1403@GMAIL.COM

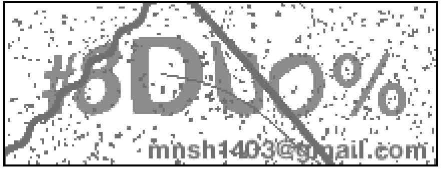

Figure 4.11: CAPTCHA obtained after setting the Signature Value = mnsh1403@gmail.com

Using the text 'mnsh1403@gmail.com' as a signature renders the above displayed CAPTCHA.

E. ADDING LINES FOR GREATER SECURITY

This is an important parameter for the security of CAPTCHA image. It helps to confuse the bots by drawing random lines over the generated image. Number of lines can be set according to the need.

a. USING NUMBER OF LINES = 1

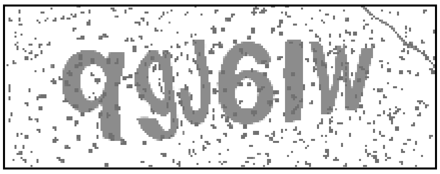

Figure 4.12: CAPTCHA obtained after setting the Number of Lines to 1

Setting the number of lines to 1 will render an image with one random line drawn over the CAPTCHA text.

b. Using Number of Lines = 3

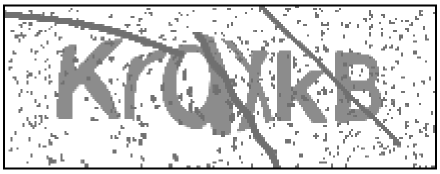

Figure 4.13: CAPTCHA obtained after setting the Number of Lines to 3

Setting the number of lines to 3 renders an image with three random lines drawn over the CAPTCHA text which is enough to confuse the bots.

c. Using Number of Lines=5

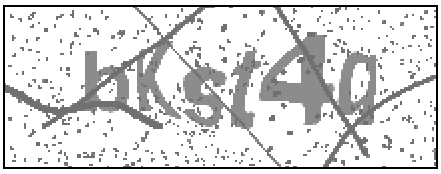

Figure 4.14: CAPTCHA obtained after setting the Number of Lines to 5

Setting the number of lines to 1 will render an image with five random lines drawn over the CAPTCHA text. Notice that more lines means more security from character recognition but it also blocks the visibility of CAPTCHA text for human users.

F. CHANGING THE CHARACTER SET:

Changing the character set is also an important feature of the proposed system. It helps in choosing a customized character set to be used in the CAPTCHA text. Either a number based, alphabet based, special character based or a combination of all these characters can be used as the character set to generate CAPTCHA challenges.

a. USING ONLY NUMBERS:

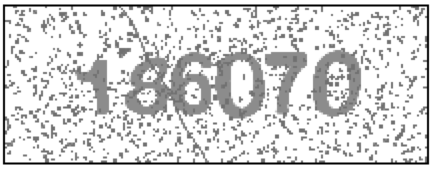

Figure 4.15: CAPTCHA obtained after using character set as Numbers only

Using the numbers as character set i.e. 0123456789 will generate challenges based on these numbers only.

b. USING ONLY LOWER CASE AND UPPER CASE ALPHABETS

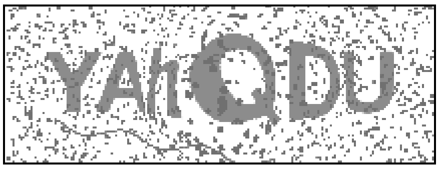

Figure 4.16: CAPTCHA obtained after using character set as Lower Case and Upper Case Alphabets

Using the lower case and upper case alphabets as character set i.e. ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz will generate challenges based on these alphabets only.

c. USING NUMBERS AND ALPHABETS:

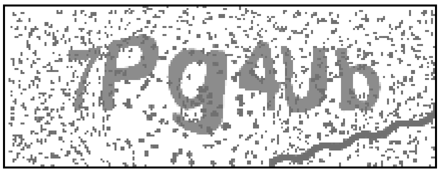

Figure 4.17: CAPTCHA obtained after using character set as Numbers and Alphabets

Using the numbers and alphabets as the character set that is ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 will generate challenges based on alphabets and numbers.

d. USING NUMBERS, ALPHABETS AND SPECIAL CHARACTERS:

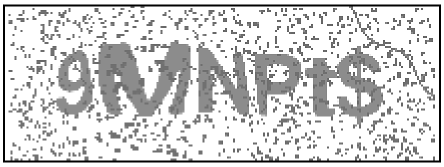

Figure 4.18: CAPTCHA obtained after using character set as Numbers, Alphabets and Special Characters

Using the numbers and alphabets as the character set that is ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789@#$%&* will generate challenges based on alphabets, special characters and numbers. This will be even more secure.

G. CHANGING THE CAPTCHA CODE LENGTH:

CAPTCHA Code Length refers to the length of the string that is produced as a challenge. The longer the string is the more secure it will be for bots to discover.

a. CODE LENGTH 4

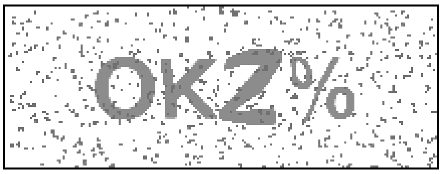

Figure 4.19: CAPTCHA obtained after setting the Code Length to 4

Keeping the code length value to 4 generates challenges with string length of 4 characters. But this should be avoided because shorter CAPTCHAs are easier to break.

b. CODE LENGTH 6

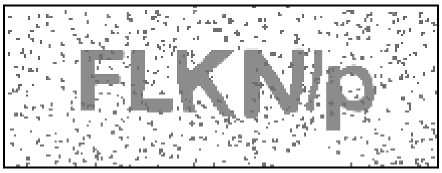

Figure 4.20: CAPTCHA obtained after setting the Code Length to 6

Keeping the code length value to 6 generates challenges with string length of 6 characters. CAPTCHAs of this length are generally secure.

c. CODE LENGTH 7

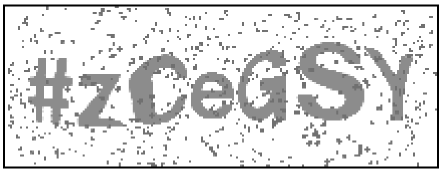

Figure 4.21: CAPTCHA obtained after setting the Code Length to 7

Keeping the code length value to 7 generates challenges with string length of 7 characters.

H. USING THE MATHEMATICAL CAPTCHA:

If there is a need to change the type of the challenge to a mathematical CAPTCHA challenge this can be also be done. A mathematical challenge provides the user with a simple arithmetic expression to be solved. If the users provide correct answer the CAPTCHA is successfully validated. This method is also much secure against bots.

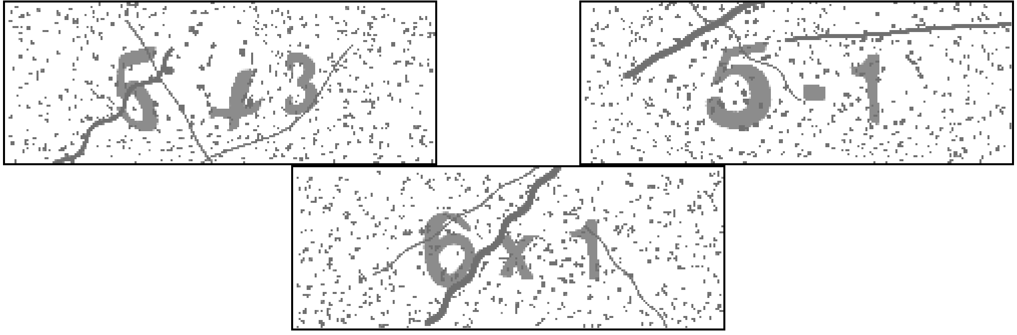

Figure 4.22: CAPTCHA obtained after using Mathematical CAPTCHA Type

I. USING WORD CAPTCHA:

The word CAPTCHA is yet another customization that can be used with the proposed system. It generates a CAPTCHA challenge based on picking random words placed beforehand in a predefined text file.

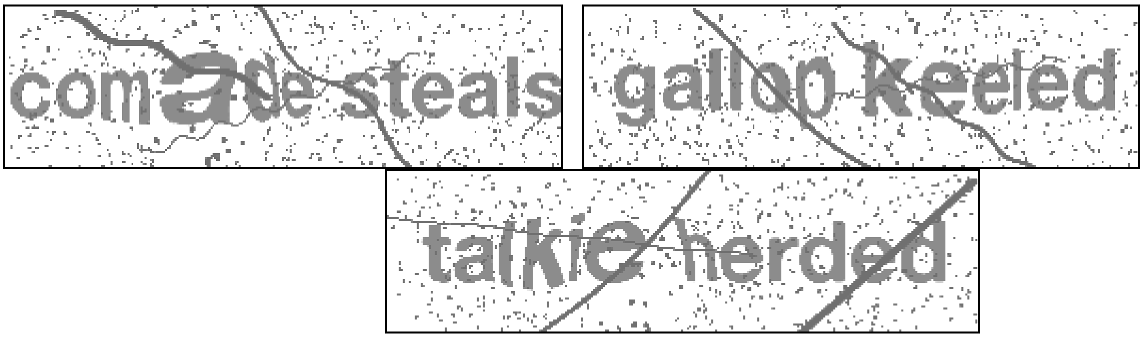

Figure 4.23: CAPTCHA obtained after using Word CAPTCHA Type

IV. CONCLUSIONS:

it is possible to enhance the security of an existing text captcha by systematically adding noise and distortion, and arranging characters more tightly. these measures, however, would also make the characters harder for humans to recognize, resulting in a higher error rate. there is a limit to the distortion and noise that humans can tolerate in a challenge of a text captcha. usability is always an important issue in designing a captcha. with advances of segmentation and optical character recognition (ocr) technologies, the capability gap between humans and bots in recognizing distorted and connected characters becomes increasingly smaller.this trend would likely render text captchas eventually ineffective. the proposed method with optimal level of parameters is able to prevent any type of bot activity. however it was observed that if more than optimal values of parameters were used it rendered the challenges unreadable which in turn is of no use.

REFERENCES:

1. Luis von Ahn, Manuel Blum, Nicholas J. Hopper and John Langford. The official CAPTCHA website, www.captcha.net, 2000
2. Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford, "Using Hard AI Problems for Security", 2003
3. Mark D. Lillibridge, Martin Abadi, Krishna Bharat, Andrei Z. Broder, "Patent US 6195698 - Method for selectively restricting access to computer systems" - Google Patents, 1998
4. Greg Mori and Jitendra Malik, "Recognizing Objects in Adversial Clutter: Breaking a Visual CAPTCHA", 2000
5. Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, Manuel Blum, "reCAPTCHA: Human-Based Character Recognition via Web Security Measures, 2009
6. Ashish Jain, Abhimanyu, "Sequenced Tagged CAPTCHA: Genaration and its Analysis", Advance Computing Conference, 2009. IACC 2009. IEEE International
7. Elie Bursztein, Matthieu Martin, John C. Mitchell, "Text-based CAPTCHA: Strengths and Weaknesses", ACM Computer and Communication Security, 2011 (CSS'2011)