INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 3 ISSUE 5 (2017) PAGES 2351 - 2356
RECEIVED : 25.09.2017 PUBLISHED : 30.10.2017
Oct 30, 2017

COMPLETELY AUTOMATED PUBLIC TURING TEST TO TELL COMPUTERS AND HUMANS APART

[1]Varjith Anchuri, [2]K.Hanuman Krishna, [3]M.Gopi Chand, [4]S.Rishi,
UG Scholar Dept Of CSE, SRM University,Chennai.

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 "**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part", 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: Yahoo, Infiltrade, Captcha, Worms.

1. INTRODUCTION

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.

2. OBJECTIVE

The project Entitled **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.

3. APPLICATIONS OF CAPTCHAS:

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.

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 3 ISSUE 5 (2017) PAGES 2351 - 2356
RECEIVED : 25.09.2017 PUBLISHED : 30.10.2017
Oct 30, 2017

- **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.

4. 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.

5. PROPOSED SYSTEM

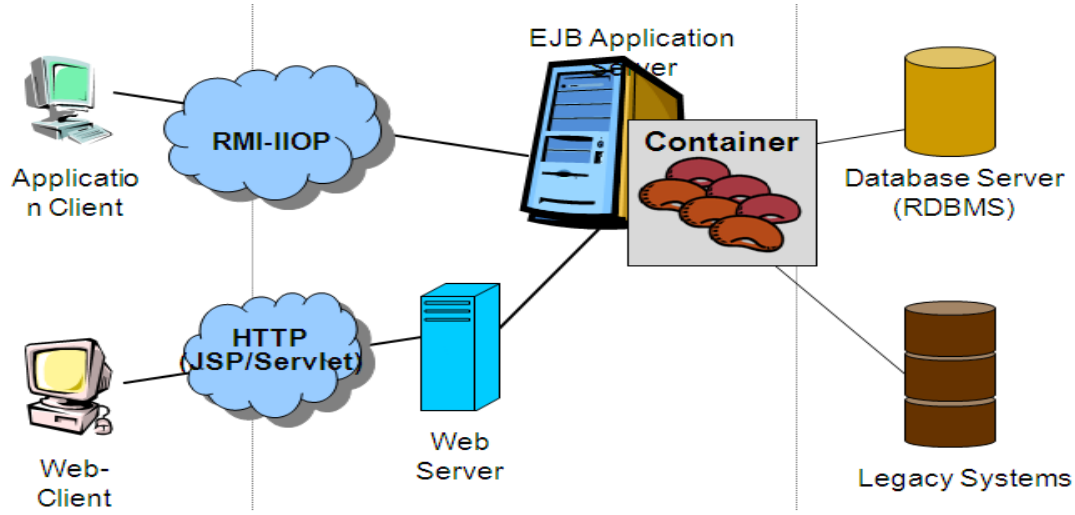

Fig.1. Steps in the execution of a JSP Application

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.

INTERNATIONAL RESEARCH JOURNAL IN ADVANCED ENGINEERING AND TECHNOLOGY (IRJAET) E - ISSN: 2454-4752 P - ISSN : 2454-4744
VOL 3 ISSUE 5 (2017) PAGES 2351 - 2356
RECEIVED : 25.09.2017 PUBLISHED : 30.10.2017
Oct 30, 2017

6. SAMPLE CODING

MyMaingui

```
function varargout = MyMainGUI(varargin) gui_Singleton = 1;

gui_State = struct( 'gui_Name', mfilename, ...

gui_Singleton', gui_Singleton, ...

gui_OpeningFcn', @MyMainGUI_Openingcn, ...

gui_OutputFcn', @MyMainGUI_OutputFcn, ...

gui_LayoutFcn', [] , ...

gui_Callback', []);

if nargin && ischar(varargin1)

gui_State.gui_Callback = str2func(varargin1);

end

if nargout

    varargout1:nargout

= gui_mainfcn(gui_State, varargin:);
else

gui_mainfcn(gui_State, varargin:);
end

function MyMainGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

guidata(hObject, handles);

function varargout = MyMainGUI_OutputFcn(hObject, eventdata, handles)
varargout1 = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
global file I ir ic d thvec;
```



```
thvec=[0.5,0.5,0.5]; [filename,pathname]=uigetfile('*.png','Pick the wav file');
file=strcat([pathname filename]);

I=imread(file);
    ir ic d

=size(I);

figure, image(I);

title('Original Image');

function pushbutton2_Callback(hObject, eventdata, handles)

global I A B;

A=rgb2gray(I);

figure,imshow(A);

B=medfilt2(A);

figure,imshow(B);

function pushbutton3_Callback(hObject, eventdata, handles)

global thvec ic ic I A img1D fid imgHex binary S valRGB Iback IbackTrim;

imgTrans = A';

img1D = A(:);

imgHex = dec2hex(img1D);

fid = fopen('im16_1.txt', 'wt');

fprintf(fid, 'fclose(fid)

binary=dec2bin(imgHex)

    S,valRGB

=Huffman_Encode(I,thvec);

Iback=Huffman_Decode(S,valRGB);
```

```
IbackTrim=Iback(1:ir,1:ic,:);

function pushbutton4_Callback(hObject, eventdata, handles)

global decimal fid1 img1 IbackTrim binary ;

decimal = bin2dec(binary);

fid1 = fopen('out_1.txt', 'wt');

fprintf(fid1 )

img1 = fscanf(fid1, 'fclose(fid1)

figure, image(IbackTrim);

title('Output Image).
```

CONCLUSION

As a contribution toward improving the web security in the field of an automated challenge and response against attacks issued by automated programs, we proposed a simple text_based CAPTCHA. Two main goals have been considered to be achieved that is: simplicity of solving the technique for a human as well as the time that a human actually needs to find the solution.. To increase the difficulty for segmentation and recognition attacks on Captchas, we varied these significant features at each challenge in ranges potentially acceptable to human users. Our mechanism provides a solution to maximize the robustness and usability of text-based CAPTCHAs simultaneously.

REFERENCES

[1] Mumtaz M. Ali AL-Mukhtar and Rana Riad K. AL-Taie : A More robust text based captcha for the security in web applications.
[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. Computer Vision–ECCV 2006, pages 404–417, 2006.
[3] E. Bursztein and S. Bethard.Decaptcha: breaking 75% of eBay audio CAPTCHAs. In Proceedings of the 3rd USENIX conference on Offensive technologies, page 8. USENIX Association, 2009.
[4] E. Bursztein, S. Bethard, Fabry C., Dan Jurafsky, and John C.Mitchell. Design parameters and human-solvability oftext-based captchas. To appears.
[5] ElieBursztein, Steven Bethard, John C. Mitchell, DanJurafsky, and Celine Fabry. How good are humans at solvingcaptchas? a large scale evaluation. In Security and Privacy,2010.