International Engineering Journal For Research & Development

Vol.5
Issue 5

E-ISSN NO:-2349-0721

Impact factor : 6.549

DYNAMIC CAPTCHA

[1]Abhishek Chaudhary, [2]Prathamesh Deogharkar, [3]Sapna Dhanraj, [4]A. R. Sonule
Computer Engineering A. C. Patil College of Engineering Navi Mumbai, India[1,2,3,4]
cabhishek745@gmail.com[1], deogharkar47@gmail.com[2] , dhanrajsapna9011@gmail.com[3],
arsonule@gmail.com[4]

ABSTRACT

A CAPTCHA, elaborated as a Completely Automated Public Turing Test to Tell Computers & Humans Apart is a popular software used to prevent automated bots from engaging the web applications and hogging resources. All the available CAPTCHA formats remain insufficiently resistant to bots, especially when combined with a relay attack. We propose a new type of dynamic CAPTCHA that is resistant to automated as well as relay attacks due to its dynamic nature. In our CAPTCHA, the user needs to identify a moving object as the target, from among a number of randomly moving decoy objects and trace that target with the mouse cursor. The user passes the test when they are able to trace the target for a certain amount of time. The latency introduced to the remote solver makes it difficult to break the CAPTCHA since the target object moves dynamically. It is also difficult for a bot to track the target using image processing because there are a number of similar looking objects. With the CAPTHCA's parameters set to suitable value, a relay attack cannot be established economically and false acceptance rate with bots are minimized without affecting human success rate.

Keywords—CAPTCHA, Relay attack, Automated bot.

INTRODUCTION

As the Internet technology rises towards the pinnacle of advancements, a number of Web Applications are being developed. With the increase in these web applications, the number of unethical people seeking to exploit this privilege has also risen. One of the major tools used for exploiting these services is an automated bot that hogs resources unnecessarily. A bot is the malicious program, a software application which has the capability to perform repeated tasks automatically over the internet and thus creating problems in the network [2]. CAPTCHA is reverse Turing test that is used to differentiate between humans and automated bots.

The first inspiration for Captcha originated from the online surveys and email spam. Captcha is a sort of challenge response test. It was initially developed by Atla vista in 1997, to prevent bots from "including URL" function of their search engine. The term CAPTCHA was instituted in 2000 by Luis von Ahn, Nicholas J. Hopper, Manuel Blum of Carnegie Mellon University and John Langford of IBM [3].

Generally, a CAPTCHA should have the following properties:

- It should be accessible.
- It should be non-troublesome and straight forward to the end user.
- It cannot stigmatize or redirect from the basic role of the page.
- It should be automated.
- It should not put a huge strain on the server/browser.

Our work intends to develop a new Dynamic CAPTCHA having all the previously mentioned characteristics along with the resistance towards relay and automated attacks while maintaining high usability.

The rest of the paper is organized as follows: section II gives literature review of the work and points drawbacks of existing systems. Section III gives proposed system in detail. In section IV, the Architecture of the proposed system is depicted. The explanation of experiment and its analysis is given in Section V. Section VI presents software and hardware used for proposed work. Finally, section VII concludes the work and gives future direction.

LITERATURE REVIEW

There are various types of CAPTCHA in existence. Among those the most commonly used ones are the Text based, Image based, Audio Based and Re-captcha. Each one of these, have some drawbacks associated with them, most originating from their static nature. Text based CAPTCHA is the most widely used CAPTCHA in web application. It is an image of distorted text/numbers in addition with some background noise or clutter. The content is generated randomly either text or alphanumeric [2]. If the characters are too distorted then users may face problem while identifying the literals correctly and if the CAPTCHA contains letters with less distortion then it can be easily identified by OCR (Optical Character Recognition) techniques.

Image-based CAPTCHAs are challenge-tests in which the collection of several images is given to the users and they have to guess those images that have some similarity as per the associations given in with the CAPTCHA [2]. A drawback is that is requires large number of images in the database. Sound-based CAPTCHAs are based on the auditory perception of human users, and can be divided into two categories. The first one presents users with a sound clip which contains distorted numbers and characters with background noise [5]. A drawback is that it is available only in English therefore end user must have a comprehensive English vocabulary. Character that have similar sound can raise some problems. The other kind offers sounds related with images. Sound-based CAPTHAs have been broken by high-quality voice recognition and noise removal programs.

RELAY ATTACKS

Relay attack is the common vulnerability of all the static CAPTCHAs. Fig.1 depicts the general working of a relay attack. The automated bot receives the CAPTCHA and relays the CAPTCHA test to a remote solver. For simple text- based CAPTCHA, the bot can simply take a screenshot of the CAPTCHA and relay it to the solver. The remote solver is a person that is paid to solve the CAPTCHAs. The solver then easily solves the CAPTCHA and then relays back the answer of the CAPTCHA. The bot receives the CAPTCHA's answer and enters it. The server verifies the answer and deems the bot as a legitimate Human user. Thus, in this way the bot is able to bypass the CAPTCHA using relay attack.

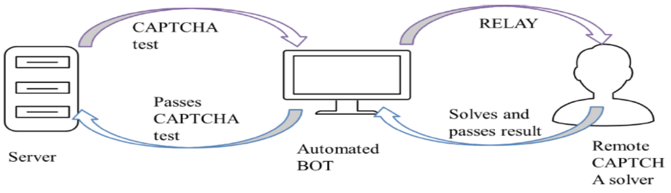


Fig. 1. Working of relay attack

Several Dynamic CAPTCHAs have been introduced to counter this relay attack. DCG-CAPTCHA is an example of such a CAPTCHA, that attempt to identify delays caused by communication relays. In DCG-CAPTCHA, a user drags and drops a moving object with the same contour to an answer area. Since this CAPTCHA has interactive and dynamic features that targets the network latency inherent in relay attacks (such as relaying the CAPTCHA challenge, task solving by remote human), it offers a higher level of resistance to those attacks. However, DCG-CAPTCHA has been shown to be insufficiently robust against automated attacks based on image processing [1,4].

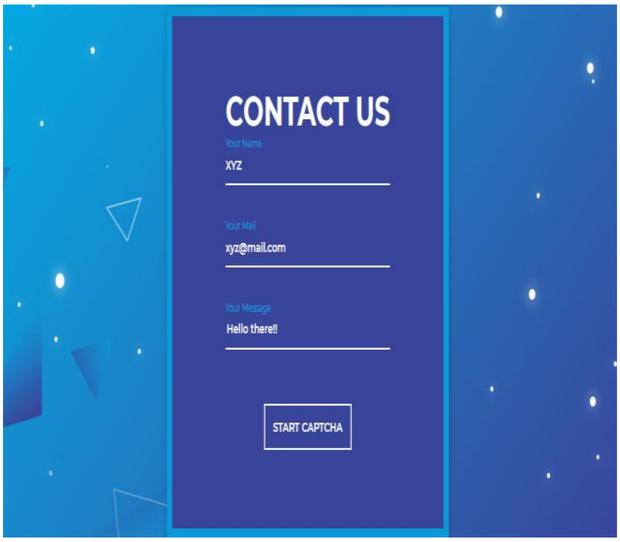
PROPOSED SYSTEM

In the proposed system, we present a new dynamic CAPTCHA that is resistant to automated attacks as well as relay attack. Fig. 2 shows the served web page. The user fills in the data and then clicks on "START CAPTCHA" button. Upon pressing this button, the user is presented with the CAPTCHA consisting of a window (Refer Fig. 3) in which there are a number of randomly moving circles and squares. Each object (circle and square) is of slightly different size and moves with a different velocity. The target object can either be a square or a circle and will have a different color than the rest of the decoy objects. The user will have to differentiate between the target and the decoy objects on the basis of this color difference. After identifying the target object, the user needs to trace (keep the mouse pointer on the object) that target with the mouse cursor.

Starting the CAPTCHA also initializes a counter mechanism and a timeout mechanism. The counter mechanism is used to gather the points obtained by the users. The counter samples the screen at each 100ms and if the mouse pointer is inside the target object's dimensions, a point is rewarded to the user. The CAPTCHA window is present on the screen only till the timeout of 15 sec expires.

$$\textit{Points (max)} = 15\ \textit{sec}/100\ \textit{ms} = 150 \quad \ldots\ldots\ldots\ldots \quad (1)$$

As per (1) the point scale is from 0 to 150. The "RELOAD CAPTCHA" button which appears after starting of the CAPTCHA test is used for reloading the CAPTCHA. The shape of the objects, motion of the objects and the color of the objects are all reloaded on pressing of this button. The points counter and the timeout mechanism are also reinitialized



CONTACT US

Your Name
XYZ

Your Mail
xyz@mail.com

Your Message
Hello there!!

Fig. 2. Starting Page

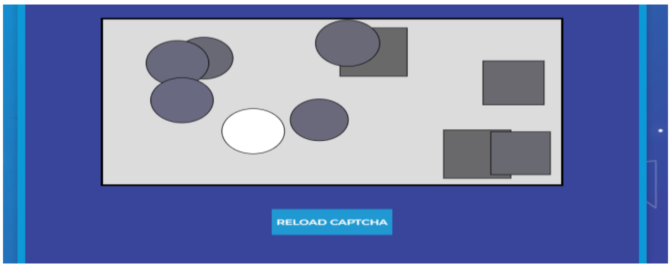


Fig. 3. Proposed Dynamic CAPTCHA

Upon the expiring of the timeout, points gathered by the user along with the data to be submitted/ other request/ services are sent back to the server. The server then evaluates the points. If the points are more than a pre-determined threshold value only then, the user is deemed as a legitimate user and the data is submitted/response is sent. If the points are below the threshold level, the user does not pass the test and the necessary actions are taken.

ARCHITECTURE

The Architecture of the proposed system is depicted in Fig. 4. The user requests for the website or a form which he/she need to access. This website is hosted by the server which creates a session for that user and returns a webpage. The user will then fill the form and then click on the start CAPTCHA button to start the CAPTCHA test. He/ She will try to trace the target object. Once the timeout expires, the server will receive the result value along with the data to be submitted if any. Fig. 5 depicts the flow of this whole process.

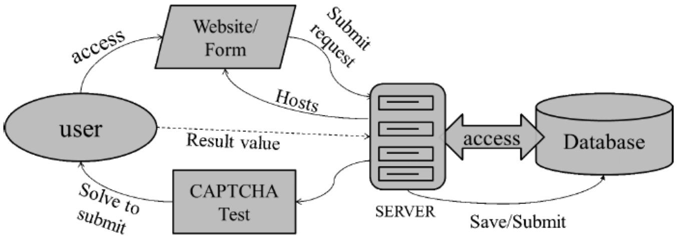


Fig. 4. Proposed Dynamic CAPTCHA Architecture

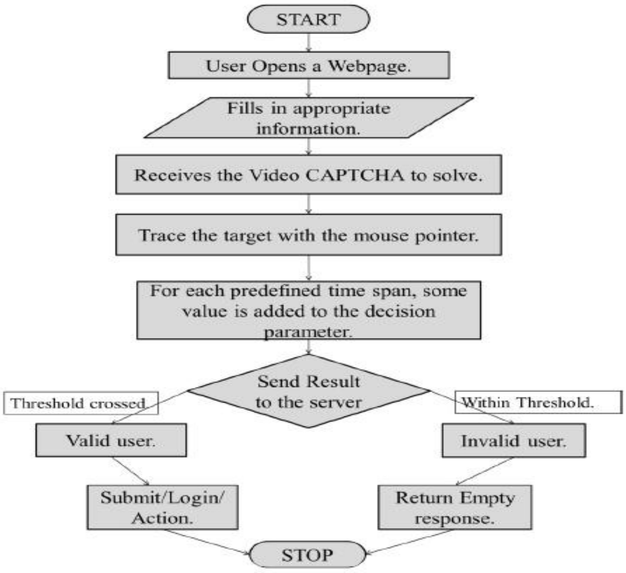

START
User Opens a Webpage.
Fills in appropriate information.
Receives the Video CAPTCHA to solve.
Trace the target with the mouse pointer.
For each predefined time span, some value is added to the decision parameter.
Send Result to the server
Threshold crossed.
Within Threshold.
Valid user.
Invalid user.
Submit/Login/ Action.
Return Empty response.
STOP

Fig. 5. General Flow of the system

Having a fixed scale for evaluation can introduce some problems. For example, consider the instance where the scale of evaluation is 0 to 150 and the threshold to cross is set at 100. An attacker having the knowledge about this fixed scale and threshold can hardcode the bots to return a value greater than 100 to gain access. This situation can defeat the purpose of the CAPTCHA. Hence, instead of using a fixed scale, the proposed system uses a dynamic range along with a dynamic threshold. Hence the server while returning the CAPTCHA test generates a random number, say '*x*' as the starting of the scale. The end of the scale is calculated as '*x + 150'*. The threshold can be determined by adding a particular value to this '*x*', tentatively '*x + 150*'. Since this '*x*' will be random for each session, there is no question of hardcoding the result value.

Experiment And Analysis

To test the CAPTCHA's robustness against relay attack we used a VNC client to act as a remote solver. Two VNC clients named, Team Viewer and AnyDesk were used for the experiment. The latency that was introduced to these VNC was up to such an extent, that the target was not at all Trackable whatsoever. Even if we consider a VNC with a very low latency period, the round-trip latency to relay the actions will still render the relay attack unsuccessful. We confirm that the success rate of relay attacks is not more than 2-3%.

The false acceptance rate of an automated attack using mean shift algorithm for such dynamic CAPTCHA is equal to 0.01%, which is a value required from security point of view [1].

To test the usability aspect of the CAPTCHA we had a number of people solve the CPATCHA. 95% of the users were able to pass the captcha test. Hence, the False Rejection Rate of this CAPTCHA is less than 5%.

Softare and Hardware Details

The following software and hardware are used for proposed Dynamic CAPTCHA.

A. Hardware Requirements

a) *Server*

- Processor: Minimum Clock speed 2 GHz
- Ethernet connection (LAN) or Wi-Fi
- Hard Drive Space: Minimum 32 GB; Recommended 64GB or more
- Memory (RAM): Minimum 4 GB; Recommended 8GB or more
- Keyboard and Mouse

b) *Client/s*

- Processor: Minimum Clock speed 2 GHz
- Ethernet connection (LAN) or Wi-Fi
- Hard Drive Space: Minimum 16 GB; Recommended 32GB or more
- Memory (RAM): Minimum 2 GB; Recommended 4GB or more
- Keyboard and Mouse
- Monitor

B. Software Requirements

a) *Server*

- Sublime text 3 or any good text Editor
- HTML5, CSS3, JavaScript and its Libraries

- Python, Java or node.js for Backend

b) *Client/s*

- Latest web browser, Firefox or Chrome

CONCLUSION AND FUTURE SCOPE

A number of CAPTCHAs are available but none of them provides total protection against relay attack. We proposed a dynamic CAPTCHA that is resistant to relay attack and automated attacks as well. We implemented this CAPTCHA and performed various viable tests on the same. The tests revealed that the false acceptance rate due to relay attacks on our CAPTCHA was as low as 2-3%. Also, the usability of our CAPTCHA is excellent. The false rejection rate of users was calculated to be less than 5%.

In the future works, this CPATCHA system can be implemented as an API, so that anyone can integrate and use this dynamic CAPTCHA into their websites to enhance their protection against automated and relay attacks.

REFERENCES

[1] S. Usuzaki et al., "Interactive video CAPTCHA for better resistance to automated attack" 2018 Eleventh International Conference on Mobile Computing and Ubiquitous Network (ICMU), Auckland, New Zealand, 2018, pp. 1-2.

[2] K. Aiswarya, K. S. Kuppusamy “A study of audio captcha and their limitations” in International Journal of Science and Research (IJSR), 2015.

[3] “The Official CAPTCHA Site.” [Online]. Available: http://www.captcha.net/. [Accessed: 29-Feb-2020].

[4] M. Mohamed, S. Gao, N. Saxena, and C. Zhang, “Dynamic cognitive game captcha usability and detection of streaming-based farming,” In The Workshop on Usable Security, co-located with NDSS, 2014.

[5] H. Gao, H. Liu, D. Yao, X. Liu and U. Aickelin, "An Audio CAPTCHA to Distinguish Humans from Computers," 2010 Third International Symposium on Electronic Commerce and Security, Guangzhou, 2010, pp. 265-269.