# Do re-CAPTCHA Works as Intended?

**Myrna E. Santiago**[1]

[1]**Old Dominion University**

## ABSTRACT

The purpose of this review is to analyze the article, I am Robot (Deep) Learning to Break Semantic Image CAPTCHAS. Sivakron, Polakis, and Keromytis used a series of experiments to test how effective and reliable using Semantic Images Captchas was. They trained a model that was good online and offline breaking Captcha images and alpha-numeric distorted strings. Their analyses and suggestions were met with positive views by Google, and the company used the experiment's findings to modify and fortify Google's re-Captcha service.

Keywords: Captcha, re-Captcha, Images, Breaking

## INTRODUCTION

In recent years deep machine learning has grown exponentially. Many models are created daily to solve complex problems. With the boom of deep learning also comes cybercrime. The need for authentication and more complex security mechanisms is rising rapidly. An aspect that took a huge leap was solving puzzles to authenticate and prove that a human was requesting entry to resources. This idea was adopted by many, but Google implemented it into a program to help their users and clients be safer online and in other areas of service. Captchas have been around for a while and Google continues to evolve their program. Then the article's authors decided to test how effective the new Captcha program was and their findings were surprising. This paper will discuss their findings and conclusions.

## METHODS AND MATERIALS

Sivakron, Polakis, and Keromytis wanted to determine how secure reCaptcha was if confronted with a deep learning model. They first collected information on the different layers of solving captcha puzzles. They also learn about each step in the new captcha and how the system works.

Google reCaptcha works the following way: When you request access to resources or a protected web page you will see a widget like Figure no.3, the widget needs to be checked mark. The code for the widget is obfuscated to prevent external analysis. The widget collects browser information upon loading. This step also allows for checks in the browser and to discard any attempts from known automatization kits. After clicking the checkbox Google receives a request containing the referrer, the website's site key, google.com's cookie, and the information obtained in the widget step. If the system accepts all the information provided a puzzle will be presented to the user for a limited amount of time to be solved.

The authors also studied the available captchas from complex text to image puzzles. Their goal was to design a method that bypasses security measures influencing the analysis risk assessment. The Web-Driver Selenium was used to create the system, because of its rich functionality and handling of all aspects of the webpages required to bypass the browser checks.

The system created work in two steps. The first one created tracking cookies to influence reCaptcha in the type of challenge that it provided. The second part solved the puzzle. The cookie manager created cookies that were trained to be almost identical to real ones. These cookies training were done in a clean virtual machine that was used to follow human patterns.

The reCaptcha breaker was created using a functionality similar to Google Reserve Image Search. Each image was searched into this functionality and the results were captured and added to the training material. The web pages and words were also taken, and translated to English if needed. This allowed robust training and was added to the model, which provided versatility in terms of possible solutions.

Their method also used Image annotations, taken from free services and libraries. These services help with assigning tags and providing descriptions of images. Because sometimes the tags did not match the images, they added to the machine learning a best guess based on subsets of tags.

## ATTACK

The experiment's goals were to train the model on human behavior and to receive captchas that only required check boxes as authentication. During the start of the attack, the model quantified the browsing history needed before a checkbox challenge was offered. They concluded that not using any type of account to verify users responded better than accounts used without verification or even verified ones to alternate emails or cellphone numbers. The interesting part was that the model was presented with the checkbox challenge 9 days after requested access on systems with no accounts verified. The model used to verify the accounts has to wait 60 days to receive the same challenge. The experiment also showed that geolocation did not impact the results.

The live attack demonstrated that the models were a little over 70 percent successful in breaking in. These results were not so surprising. Previous observations proved that image repetition is a problem for the captchas. Also, hint repetition was used and very useful to break the captcha thanks to 5 type counting for 54 percent and 10 types counting for 91 percent of the challenges. The models were also quite effective online making them cost effective.

## COUNTERMESURES

One of the proposed countermeasures is token auctioning, this allows the comparison of the IP address that solved the captcha with the one that submitted the token. If made mandatory it will increase the cost of multiple attacks serving as a deterrent for threat actors.

Another countermeasure is requests through verified Google accounts. These accounts are hard to maintain because of phone verification. A person can maintain a couple of accounts under one number but multiple ones become a problem for the actor

and higher security for Google. Cookie repudiation can be a solution, especially if a cookie is just created. There should be a specific amount of time that passes before the browser is deemed safe. A browser check should be performed between the one in use and the one in the User-Agent slot.

Another possible deterrent is not allowing flexibility in captcha selection. It will force the user to only select the correct images. Not using hints on images and avoiding repetitions of challenges also contribute to raising security measures. Another method for images is making them high-resolution which increases attack cost and raises the security measures.

## Figures and Tables

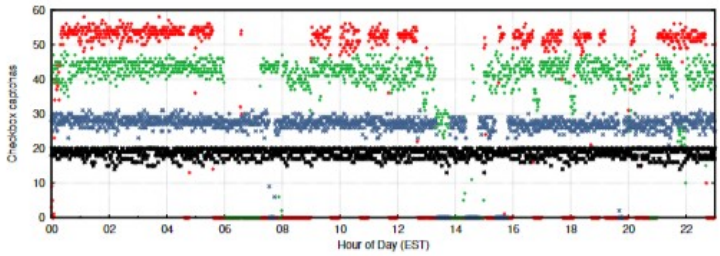


**Figure 1.** Checkbox challenges created per minute. The lower-frequency attacks produce more checkbox challenges than the more aggressive ones.

| Image Selection | Constraint | Pass |
|---|---|---|
| $n$ Correct + $k$ Wrong | $k \leq 1$ | ✓ |
| $(n-1)$ Correct | $n > 2$ | ✓ |
| $(n-1)$ Correct + $k$ Wrong | $k > 0$ | ✗ |

**Figure 2.** It shows the combinations made to pass or fail the image selection challenge from a mathematical point of view.


(a) Before user clicks checkbox. (b) User considered human.

**Figure 3.** The checkbox which is the main goal of the experiment, since is the easiest captcha to break.

## Citations

Table and figures belong to the article of Sivakorn et al. (2016).

## CONCLUSION

These experiments and articles were essential to cybersecurity because they allowed the discovery of grave flaws in the recaptcha program, which is part of the security measures in many websites. By applying the recommendations, Google demonstrated a high value in these types of research. This article not only showed the steps taken but also the level of depth and ethical knowledge demonstrated by the researchers. They were careful in each step and shared their findings with the proper authorities instead of cashing in the exploits. Multiple research and experiments have been done on this theme in the last decade alone. The purpose was to create a secure method of authentication. A great lesson given is to not take for granted the security measures provided through services, even if they come from reputed sources. We need to test each security measure and keep up with the cybersecurity news to see the latest research and to make sure we create effective ways as countermeasures against cybercrime.

## REFERENCES

Sivakorn, S., Polakis, I., and Keromitis, A. D. (2016). I am robot (deep) learning to break semantic images captchas. *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, (15987970).