

CAPTCHA - Reliability and Security

Erik Zhao, Jørgen Flægstad, and Krar Zayadi

University of Oslo

{erikdz, jorgengf, kraraz}@student.matnat.uio.no

May 2017

Abstract—This is a home exam for the subject INF3510 - Information Security. This paper will investigate the security of CAPTCHAs as a measure to stop bots. We will also look at how CAPTCHAs work, their popularity, and some popular alternatives.

I. INTRODUCTION

A. What is CAPTCHA?

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a program that focuses mainly on protecting websites against bots. CAPTCHAs generate various tests or challenges that the user must complete before gaining access to a certain feature. The challenges should be designed such that a human can easily solve them, but a computer can not. CAPTCHAs are fairly easily installed and requires little to no maintenance.

CAPTCHA was first invented in 1997 by M.D. Lillibridge, M. Abadi, K. Bharat, and A.Z. Broder, as they added a test to distinguish human users from bots on the web search engine AltaVista[26]. The term CAPTCHA however was first coined in 2003 by L. von Ahn, M. Blum, N.J. Hopper, and J. Langford[25].

B. The Turing Test

CAPTCHA technology has its foundation in an experiment called *the Turing test*. The Turing test was developed by Alan Turing, often thought of as the 'father of modern computing', in 1950 to test a computers ability to exhibit intelligent behaviour that mimics that of a human being. The classic test is a game of imitation; an interrogator asks two participants a series of questions, one of the participants is human, the other a computer. The interrogator can not see or hear either participant and has no way of knowing which of them is the human. If the interrogator is unable to figure out which of the participants is human based on the responses, the computer passes the Turing test. In the case of CAPTCHAs, it is actually a Reverse Turing test as the interrogator is not human.

C. Aim of Paper

In this paper we aim to look at the security behind CAPTCHAs, more specifically the techniques spammers use to bypass or break them (Section III). We will also look at how CAPTCHA schemes work and why they are important (Section II), and finally we will look at some alternatives to CAPTCHAs to combat spam (Section IV).

Websites using reCAPTCHA

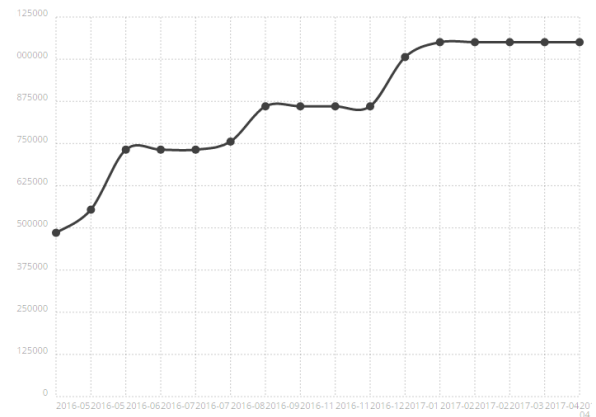


Figure 1. The popularity of reCaptcha by Google[23]

II. CAPTCHA

A. Uses and popularity

Since its origin CAPTCHA has seen use all over the internet. According to Scrapesentry around 3.5 million websites use it in 2017[24]. builtwith.com reports that around 11% of the top 10k websites in the world use CAPTCHAs, where reCaptcha from Google is by far the most popular, being used by about 90% of the sites that use CAPTCHAs[22]. About one million websites on the internet use reCaptcha, which covers around 0.3% of all websites. In Fig. 1 one can see the growth of reCaptcha from May 2016 to April 2017, doubling in usage over just a year. The second most popular CAPTCHA, "Are You a Human?", sees use by about 32,000 sites. What are the rest of the millions of sites using to combat spam? Many websites are finding alternatives to CAPTCHAs, we will discuss these in section IV.

B. Why is CAPTCHA important?

The main task for the CAPTCHA is to stop spam. CAPTCHAs are typically used by sites as an automated method of stopping interactions with computer programs commonly known as bots. The bots are used by spammers to automatically perform certain tasks such as: create a large amount of accounts (for example email accounts to be used for malicious purposes like spamming), manipulating online polls, or creating a massive amount of traffic to a site to cause a

Denial of Service (DoS) attack. CAPTCHAs aim to stop these bots by requiring the user to verify that he or she is human, and not a malicious program, by completing a challenge.

C. How CAPTCHA works

Most commonly, a CAPTCHA consists of a visual challenge in the form of distorted characters or images, although audio-based challenges also exist. N.M Al-Fannah suggests that a successful CAPTCHA needs the following four properties[27],

- *Ease of Computers to Evaluate*: the correctness of the solution to the puzzle should be simple for the computer to verify.
- *Ease for Humans*: the puzzle should be easy for humans to solve.
- *Ease of Generation*: the generation of puzzles in software should be straightforward.
- *Challenging for Computers*: solving a puzzle should be difficult for a computer.

CAPTCHA usually test the user's human capabilities by testing invariant recognition, segmentation and context, properties that are very difficult for a computer to reproduce.

- Invariant recognition refers to the ability to recognize an object regardless of irrelevant image variations.
- Segmentation refers to the human ability to segment and recognize connected characters and objects.
- Context is important, the CAPTCHA must be understood holistically to correctly identify each character.

All of these properties pose a significant challenge for computers, which is what makes CAPTCHAs difficult to solve. Even so, people find exploits to break CAPTCHAs all the time, which we will take a closer look at in section III.

D. Different types of CAPTCHA

There are generally *three* different CAPTCHA challenges:

- Text-based CAPTCHA
- Audio-based CAPTCHA
- Image-based CAPTCHA

Text-based CAPTCHA is the original CAPTCHA, the type most people probably think of when they hear the word "CAPTCHA". The user is presented with an image of a few characters, either letters or digits, and has to write down the characters presented. The characters are usually distorted and the image can have a lot of background noise. Fig. 2 shows an example of a text-based CAPTCHA taken from Facebook.

Audio-based CAPTCHA is similar to the text-based version, except instead of providing the user with a visual image of characters, the user will instead be played an audio file containing a few characters that will have to be written down. This is by far the least popular form of CAPTCHA and also has the lowest completion rate out of all of them. Because of this it is seldom used today. Fig. 3 shows an example of an audio-based CAPTCHA.

Finally we have Image-based CAPTCHA. In image-based challenges, the user is presented with several images and will usually have to identify those photos that are relevant to a certain hint, although there are other variations with different



Figure 2. Text-based CAPTCHA from Facebook.com

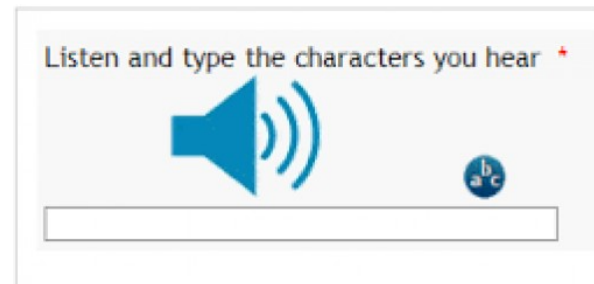


Figure 3. Audio-based CAPTCHA from Wikipedia.org

schemes. These CAPTCHAs are fairly popular for high-traffic sites, but have a few drawbacks. Pictures take up more space on the screen than a text-based CAPTCHA, which could be distracting to the user, as well as being inconsistent with the website. Picking out cats from dogs (Like in Microsoft's *Asirra*) would be inappropriate on the site of a medical institution or a bank. The latter issue could be fixed by having relevant images depending on the content on the site, but that would require much more work and effort. Fig. 4 on the next page shows an example of an image-based CAPTCHA where the user has to identify images containing soup.

E. User experience

Even though CAPTCHAs are made to stop bots, they sometimes tend to stop legitimate users as well. A team from Stanford University tested humans' ability to solve CAPTCHAs and found that when three different people were presented a visual CAPTCHA, they only agreed 71% of the time. For audio CAPTCHAs they only agreed 31% of the time[16]. The team also found that only 1% of users prefer audio-based CAPTCHAs over text-based ones. Non-native speakers of English are slower and less accurate than native speakers, as well as age and education being a factor to how quickly and accurately users solved the challenge. CAPTCHAs are especially difficult for disabled users, such as blind or vision impaired humans, so much so that a blind Australian started a petition to remove CAPTCHAs as he felt they were prohibiting him from using the internet[17].

The fact that CAPTCHAs are difficult to solve quickly makes them very annoying to users. Webnographer conducted

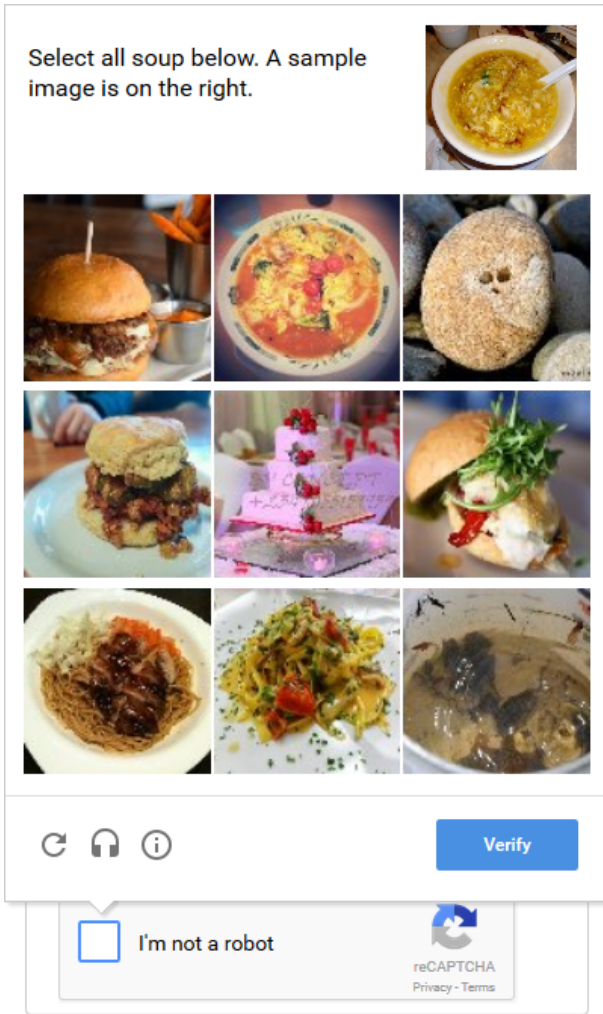


Figure 4. Image-based CAPTCHA from Google's reCaptcha

an online test where they found that only 62% of users completed the CAPTCHA on their first try, 23% succeeded after multiple attempts, and 15% gave up completely[18]. They found that if a user fails the first time, there is a 70% to 90% chance they will give up. 36% of the participants said they had trouble with solving the CAPTCHA. As algorithms have gotten better and better at segmenting, the CAPTCHAs have had to become more and more complex. This does make them harder to break with bots, but at the same time they become more difficult for humans to solve.

Many sites have reported an increase in conversion rates when they have removed the CAPTCHA from their forms. reddit.com reported an 8% increase in account registration after they removed the CAPTCHA from their registration[19]. A study conducted by Animoto resulted in a form without a CAPTCHA having a conversion rate of 64%, while the form with the CAPTCHA had only a 48% conversion rate. Not having the CAPTCHA increased conversion rate by almost one third[20].

The way CAPTCHAs work is that rather than the business solving the issue of spam, the problem is pushed onto the customer to solve, creating more work and effort for the users,

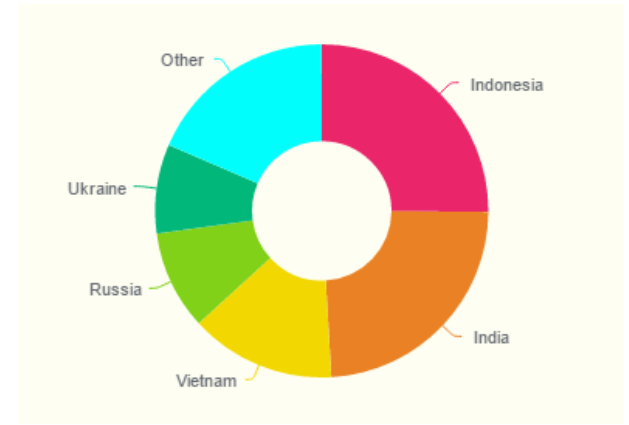


Figure 5. Origin of human operators that solve CAPTCHAs for Anti-Captcha[2]

which is what makes them so annoying. As Harry Brignull said[21],

Using a CAPTCHA is a way of announcing to the world that you've got a spam problem, that you don't know how to deal with it, and that you've decided to offload the frustration of the problem onto your user-base.

III. ATTACKS AGAINST CAPTCHA

There are generally two different ways to bypass CAPTCHAs, by using human labor or by using automated solvers that use algorithms to correctly solve the CAPTCHA.

A. Human labor

The first instance of using human labor to solve CAPTCHAs seem to be around 2006, when someone posted an ad on www.getafreelancer.com, wanting to hire someone to solve CAPTCHAs. He received 58 bids, ranging from \$30 to \$100, with the average being \$57. The ad was eventually closed down by an administrator[1]. Since then, getting other humans to solve CAPTCHAs for you has grown into a considerable market.

Human labor is flexible, and can be used for the entire range of CAPTCHA schemes. Using human labor to verify CAPTCHAs is a classic man-in-the-middle attack. CAPTCHAs are designed to validate *humans*, and can not separate the trusted users from the untrusted ones.

Since solving CAPTCHAs requires no skill or training, the work can be outsourced to countries with the lowest labor cost, like Indonesia and India. In countries like these, sweatshops are set up where human operators solve hundreds of CAPTCHAs an hour, at a very low fee. Fig. 5 shows the origin of the human operators that work for a popular CAPTCHA solver service called Anti-Captcha. At its origin in 2007, the price for solving CAPTCHAs was at around \$10 per 1,000 CAPTCHAs solved, but as the market grew, the price dropped to around \$0.5-\$1 per 1,000 CAPTCHAs solved in 2010. At the time of writing (April 2017), most distributors of human based solving charge on average \$0.7 per 1,000 CAPTCHAs solved.

A different way of using humans to solve your CAPTCHAs, is to get them to do it without them knowing it. Spammers set up pornographic sites or sites for gambling, and get users to complete a CAPTCHA to enter. These CAPTCHAs are ripped directly from the site the spammers are trying to spam, and the users are unknowingly helping the spammers[3].

B. Automated solvers

Automated solvers are programs that use algorithms to solve CAPTCHAs. These algorithms are highly complex, yet often fails to replicate human accuracy. Just like with everything, the motivation behind these solvers is money. The economics of the automated solvers are driven by several factors: the cost to develop new solvers, the accuracy of these solvers and the responsiveness of those that update the CAPTCHAs. Just like with anti-virus and anti-cheating software, there is an "arms race" between the spammers and the CAPTCHA developers. The spammers find a vulnerability they can exploit to bypass the CAPTCHA, the developers finds their mistake and patches it, and then the spammers will have to find a new exploit. Traditionally this favors the adversary, because usually it is the attackers role to generate new instances while the defender must recognize the attack - and the recognition problem is almost always much harder to do. This, however, is not the case for CAPTCHAs, as the roles are reversed, it is the attackers that have the recognition problem. The automated solvers have a shelf life, they can only be used until the solver is detected and the CAPTCHA is changed. The price of the solver must therefore reflect this. In addition, the solver must also be cheaper than the alternative, human labor. The automated solvers also need to be accurate, which is often a very difficult task. The reason they need to be accurate is because several sites have began blacklisting IPs after they fail the CAPTCHA several times.

Due to all of these issues with automated solvers, the fact that they are not always accurate, that they can become obsolete after a patch, and that they are fairly complicated to produce, the spammers realised that a good approach was to have hybrid systems, where both software and human labor is utilised. The automated software detects CAPTCHAs that are vulnerable to its automated solver, and sends the rest of the CAPTCHAs it cannot solve to human based solvers.

The solvers themselves are very different depending on the type of CAPTCHA it solves. It is generally best to focus on one specific CAPTCHA and only solve that particular one, as having a generic algorithm for all, or many, CAPTCHAs will generally result in low accuracy. Let's take a closer look at how the algorithms work for solving the four different CAPTHCAs: text-based, audio-based, image-based and the reCaptcha.

1) *Text-based:* The standard approach to solving text-based CAPTCHAs automatically has been a sequential process where a segmentation algorithm splits the image into segments that contain each individual character, followed by a character recognition step that uses machine learning to identify the character. This is known as a *segment then recognize* approach. When CAPTCHAs were first introduced, the words were of a fixed length or/and there were a limited set of words. This



Figure 6. Example of negative kerning in a CAPTCHA[4]

made it very easy for the early algorithms to guess the correct words. These days it is not as simple, as there are several new tactics to make the words difficult to crack for computers. One of the best tools against segmentation algorithms is to use *negative kerning*. Negative kerning, also known as character collapsing, uses the negative space between characters to resist segmentation by ensuring that each character is occluded by its neighbours. This method is the most used in modern text-based CAPTCHA and is still thought of as one of the most secure methods of preventing segmentation.

As one can see in the Fig. 6, the letters and numbers are merging, using up the negative space between each other, which is making it difficult to segment each character. The strength of a CAPTCHAs ability to withstand an automated solver lies in the difficulty of segmenting the image into the individual characters, rather than recognising the characters themselves.

Automated solving of text-based CAPTCHAs generally fall into two categories: using side-channel information unrelated to the distortion itself, for example dictionary attacks, and finding weaknesses in the distortion algorithms of a particular CAPTCHA. The first one treats CAPTCHAs in a generic manner, and is therefore easily patched by the defenders, the second approach is by far more successful. An example of these specific attacks was a precisely tuned segmentation algorithm that was used against a popular CAPTCHA, reCaptcha 2011, where the authors of the algorithm used a complex image preprocessing phase that relies on character alignment, morphological segmentation with three-color bar character alignment and heuristic recognition[5]. While the algorithm was very effective against reCaptcha 2011, it could not recognise or solve any other CAPTCHA schemes.

2) *Audio-based:* Breaking audio-based CAPTCHAs is done in a similar fashion to text-based CAPTCHAs, using the segment then recognise approach. Generally, the audio file is a series of spoken letters and/or digits, and so the algorithm will attempt to separate and segment each "noise" and then use machine learning to recognise the segments. One team from Carnegie Mellon University collected thousands of audio CAPTCHAs and used these to train their algorithm[6]. Each audio file is transformed into a set of fixed-size segments labelled either as noise, a digit or a letter. These segments were then used for training. The researchers used three different methods to recognise the characters; AdaBoost, Support vector machine (SVM), and k-nearest neighbor (k-NN). The SVM approach was the most successful by far, against Google, Digg and reCaptcha. The research team noted that the most difficult CAPTCHAs to break were those that used human voices for background noise instead of running water or static, which was


| | GRIS | Alchemy |
|---|----------------|-------------|
|  | wine and blood | wine, glass |

Figure 7. Output from each image annotation module[9]

much easier for the program to filter out. It was also noted that in Google's case, the human voices in the background were on a loop, which also made it easier to filter out.

From their experiments, it was noted that CAPTCHAs containing longer solutions and multiple speakers tend to be more difficult to solve. Also, since the methods depend on the amount of training data that is available, having a large vocabulary would make it more difficult to collect enough training data. The team managed to get a success rate of 45% against reCaptcha, which is quite impressive when the human pass rate is only 70%[7].

3) *Image-based*: As automated solvers got better and better at cracking text-based and audio-based CAPTCHAs, the new reCaptcha from Google was released. reCaptcha uses browser characteristics and tracking cookies to give the user a confidence score. The score reflects the confidence of the system that the user is a human user and not a bot. For high-confidence scores, the user is only required to click a check box. For low-confidence scores, the user is presented with an image-based challenge, as the system finds the user suspicious.

To break the image-based CAPTCHA, a team from Columbia University built a system consisting of two main components[8]. The first is responsible for creating tracking cookies that can influence the risk analysis process. The second component processes the challenge and solves them. The first component is used to avoid the image-based CAPTCHA, so more about that in the next section.

Once the system is confronted with the image-based challenge, the module will first extract the images as well as the *hint* that is given (for example, "wine"). Next, all the images are passed on to an image annotation module. This module uses several different image annotation services to find the "best guess" to what the image contains. The Google Reverse Image Search (GRIS) is used to reverse search google for the image, if the description is not in English, it will use Google Translate. Other services used were Alchemy, TDL, NeuralTalk and Caffe. These image annotation services all use deep learning to offer tags of what the image could portray. The module will then pick the tag that the services most agrees on and compare it to the hint given in the challenge. A table showing the results of the services can be seen in Fig. 7.

4) *Checkbox CAPTCHA*: Finally, we have the checkbox CAPTCHA, Google's new "no captcha reCaptcha". As explained in the previous section, reCaptcha will use tracking cookies and browser characteristics to determine whether or

not the user is a human or a bot. If the system is fairly confident the user is human, all the user has to do is click a checkbox confirming that they are.

The system created by the team from Columbia does not only solve the image-based CAPTCHAs, it also tries to solve the checkbox-CAPTCHA. This is what the first component of the system does. Since the Google tracking cookie plays an important role it is important that the program is able to create cookies which are subsequently "trained" to appear as originating from legitimate users, and not bots. They first create a cookie in a clean virtual machine, where the browser automation system imitates a human browsing the internet. They configured the system to perform actions specific to the website being visited. For example, the bot will conduct Google searches and follow links from the results, open videos in Youtube, and perform searches in Google Maps. The bot will also visit popular websites that contain social plugins associated to Google. As such, they are able to create Google tracking cookies that are similar to those created by human users.

The goal is that when the bot requests a CAPTCHA, it will be presented with a checkbox, not an image-based challenge. The researchers attempted to find the minimum amount of browser activity required to receive a high-confidence score, what they found is that no matter what they did, they were presented with the checkbox on the 9th day after the creation of the cookie. Even without conducting any web surfing at all, on the 9th day the bot was presented with the checkbox. When logged into an account, the threshold actually increased to 60 days. In conclusion, to avoid the image-based challenge, the best approach is to *not* use an account, and to append a 9-day old cookie to the request. The researchers found no negative effect on the mouse behaviour, cookie reputation or screen resolution, although Google claims that newer versions of reCaptcha will analyse mouse behaviour.

In the end, it seems that none of the CAPTCHA versions are safe from automated solving. Google's reCaptcha seem to be the most secure, but still has flaws that can be exploited by capable and motivated attackers. The ever ongoing arms race between the CAPTCHA developers and the exploiters will probably never halt. Research in CAPTCHA solving has always followed the same exploit-patch cycle where the attacker finds a flaw, and then the defender patches it or moves on to a different scheme.

C. Testing anti-CAPTCHA services

1) *Anti-Captcha*: Anti-Captcha uses human workers from all around the world to solve CAPTCHAs. They charge 70 cents per 1,000 CAPTCHAs and can process an "unlimited number of simultaneous uploads". First we will try Anti-Captcha against Wikipedia's standard text-based CAPTCHA. We first mark the image containing the captcha, then we mark the placement of the solution. It took approximately 20 seconds before the CAPTCHA was solved and the solution was written in and we could create our account (Fig. 8). We then attempted to create an account on MMO-champion as it has a standard text-based CAPTCHA as well, this time

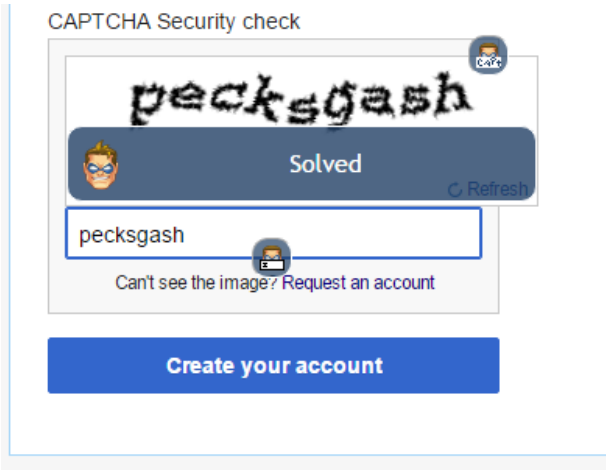


Figure 8. Anti-Captcha: CAPTCHA solved

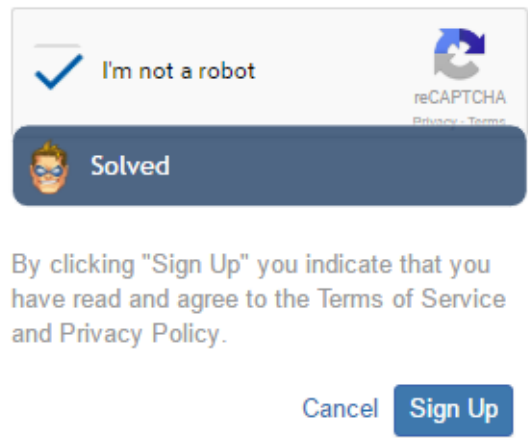


Figure 9. Anti-Captcha: reCaptcha checked

the solution took 35 seconds. In fact, Anti-Captcha worked fairly well for all text-based challenges. We then wanted to test Anti-Captcha against reCaptcha’s checkbox, so we went to make an account of Quora.com. The moment we entered the account creation page, the CAPTCHA was almost immediately ”solved” as the checkbox was checked (Fig. 9). It seems that Anti-Captchas human based service works quite effectively.

2) *2Captcha*: We then tried 2Captcha, another human-based CAPTCHA solver service. Ironically enough, they had a reCaptcha that had to be solved to set up an account, which we used Anti-Captcha to bypass. 2Captcha uses workers around the world that sign up to solve CAPTCHAs for money. We made an account to try and become one of these solvers. First, we had to go through a ”training mode” so that 2Captcha would be certain that we could actually solve the challenges. The ”training mode” took around 10 minutes, and was an introduction to the different CAPTCHAs, as well as an ”exam” to make sure that we were capable. We were finally able to start completing CAPTCHAs for money. We spent 3 minutes solving challenges, solving a total of 22 CAPTCHAs without any errors and made 0.00712 USD, a salary of 14 cents an hour.

| | |
|--|------------|
| | neynt |
| | 867826 |
| | cubedswing |
| | e3tj6jdp |
| | 4cz8jyaz |
| | vnsqjdo |
| | apnodot |
| | advses |
| | dmvhk |

Figure 10. CaptchaSolutions: The results from 9 CAPTCHAs

3) *CaptchaSolutions*: Finally, we wanted to find a solver that uses an automated program to solve the CAPTCHA, not human labor. We ended up with CaptchaSolutions, an automated solver that decodes text-based CAPTCHAs. CaptchaSolutions uses API keys to solve your CAPTCHAs for you. We tried several different CAPTCHAs, with mixed results. Mostly, the software was very accurate, and much faster than human-based solvers, taking around 1-2 seconds to solve each CAPTCHA. The results can be viewed in Fig. 10. It managed to solve 7 out of the 9 CAPTCHAs, and we think that the bad resolution of the CAPTCHAs that it did not manage to solve might have contributed to the difficulty. While the accuracy is not as high as human-based services, it was incredibly quick at solving the challenges. The price is about the same as the human-based service. We believe that the human-based service is superior due to its higher accuracy and adaptability, although the best solution would be a hybrid service where the software tries to solve the CAPTCHA using an automated solver, if it is unable to solve it, it forwards it to a human solver. This solution is the ”best of both worlds”, as you get the speed of the automated solver and the accuracy of the human-based solver.

IV. ALTERNATIVES

If CAPTCHAs are such an annoyance to users, and are actually fairly easily bypassed by either human labor, automated solvers or a combination of both, why do millions of sites still use them? Are there better alternatives to filter out bots and counter spam?

A. Honeypot technique

One solution that does not interfere with the user experience is the so-called Honeypot technique. Essentially, the Honeypot technique tries to fool bots into auto-filling forms that are invisible to human users. The field is hidden, and so unless you either know it is there or a bot, you will leave it empty. The downside is of course that legitimate users might end up filling the field (or the browser might auto-fill it for the user), or that malicious scripts might learn which labels mean that a field should be left alone.

B. Biometric security

Most devices these days have a camera and a smart screen or trackpad, with these tools apps can use your eyes, face or fingerprints to confirm your "humanness". Using data that is tied to your DNA for verification is definitely an exciting development, but with one major concern; when someone steals your password, you can create a new one, but when someone steals your fingerprint pattern then you are in trouble. Biometric security is a form of inheritance-based security, which could become the security of the future.

C. Text Message Verification

Recently, the security of mobile apps is trying to increasingly rely on the user's cell phone number instead of the traditional Username and Password. Text Message Verification (TMV) "solves" the hacker issue by involving personal devices tied to human-only accounts. This development also signals for more personality in the security process.

D. Open Authentication

Open Authentication (OAuth) coordinates with third party accounts (Facebook, Twitter, etc.) that the user is logged on to and verifies the identity of the user through these existing profiles. Creating an account on these third party sites already requires authentication, such as CAPTCHA or alternatives, in the first place, so essentially this method aims to limit the amount of verification users have to do by connecting them to already verified accounts.

V. FUTURE OF CAPTCHA

It is quite clear that no matter what developers do, someone *will* find a way to exploit and attack their solution. There was a time when CAPTCHAs were considered virtually unbreakable, but now there are very few, if any, CAPTCHAs that can withstand a motivated and capable attacker. As Sivakorn, Polakis and Keromytis comments[10],

[T]he capabilities of computer vision and machine learning have finally reached the point where expectations of automatically distinguishing between humans and bots with existing schemes, without excluding a considerable number of users in the process, seem unrealistic.

Does the perfect CAPTCHA exist? A test that cannot be broken by a bot, but a human will pass every time? Many researchers think the answer is no, and is looking for other methods to distinguish between bot and man. A team from Google and Stanford University came up with many new potential directions for CAPTCHAs, including incorporating video or requiring the user to perform a difficult cognitive task, like circling or rotating an object[11]. There are many CAPTCHAs that use these new methods, like Microsoft's *Asirra*, which has users identifying cats out of a set of photos of both cats and dogs[12], *MintEye* CAPTCHA, which has the user "undistorting" an image that is distorted, and *NuCaptcha*, which had the user watch a video. All of these CAPTCHAs were fairly easily broken. *Asirra* was broken by a classifier trained to recognize image textures[13], *MintEye* was broken by a very simple attack containing only 23 lines of Python code[14] and *NuCaptcha* was broken by traditional OCR-based techniques[15].

It seems implausible that CAPTCHAs will ever reach a standard such that it is unbreakable, which is why many researchers have begun looking in other directions, such as biometric security and Text Message Verification. This direction shifts the anti-spam problem from "Prove you are human" to "Prove you are you". Instead of verifying your humanness, you instead verify your identity. In a sense, we would all end up with a personal unique electronic identity that mirrors our physical identity.

VI. CONCLUSION

CAPTCHAs have been used for two decades to detect bots and block spam. Combating spam is very important as there is such a massive amount of it on the internet. CAPTCHA developers and spammers are in an ever ongoing arms race where the spammers find a new vulnerability to exploit, and then the developers patch it. In the past, CAPTCHAs were fairly effective, but as OCR and segmentation technology as evolved it has become increasingly difficult to create a challenge that bots can not complete, while humans can. CAPTCHAs are also fairly unpopular among users, as they are annoying, time consuming and difficult. Due to this, there has been a shift in CAPTCHAs and anti-spam away from challenging the user, to using 'invisible' measures to verify whether or not the user is human. We believe there is still a place for CAPTCHAs on the internet for sites without a large amount of traffic that might not require state-of-the-art sophisticated anti-spam techniques, but for massive websites that need strong anti-spam schemes it does seem like we are shifting towards personal verification like TMV or biometric security.

REFERENCES

- [1] Zulfikar Ramzan: www.symantec.com/connect/blogs/captcha-solving-service, Sep 2006
- [2] Image taken from their website, anti-captcha.com, 2017
- [3] Cory Doctorow: web.archive.org/web/20060209040456/www.boingboing.net/2004/01/27/solving_and_creating.html, Jan 2004
- [4] Jim Thatcher: <http://jimthatcher.com/captchas.htm>, Mar 2009 (Although the CAPTCHA is taken from Yahoo)
- [5] C. Cruz-Perez, O. Starostenko, F. Uceda-Ponga, V. Alarcon-Aquino, and L. Reyes-Cabrera. Breaking recaptchas with unpredictable collapse: heuristic character segmentation and recognition. In *Pattern Recognition*, p. 155–165. Springer, 2012.
- [6] J. Tam, J. Simsa, S. Hyde, and L.V. Ahn. Breaking Audio CAPTCHAs, p. 2-3
- [7] J. Tam, J. Simsa, S. Hyde, and L.V. Ahn. Breaking Audio CAPTCHAs, p. 8
- [8] S. Sivakorn, I. Polakis, and A.D. Keromytis. I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs.
- [9] S. Sivakorn, I. Polakis, and A.D. Keromytis. I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs. p. 6
- [10] S. Sivakorn, I. Polakis, and A.D. Keromytis. I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs. p. 14
- [11] E. Bursztein, J. Aigrain, A. Moscicki, and J.C. Mitchell. The End is Nigh: Generic Solving of Text-based CAPTCHAs, p. 11
- [12] J. Elson, J.R. Douceur, J. Howell, and J. Saul. Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization, 2007
- [13] P. Golle. Machine Learning Attacks Against the Asirra CAPTCHA, 2008
- [14] Breaking the minteye image captcha in 23 lines of python. <http://www.jwandrews.co.uk/2013/01/breakingthe-minteye-image-captcha-in-23-lines-of-python/>
- [15] Y. Xu, G. Reynaga, S. Chiasson, J.M. Frahm, F. Monrose, and P. van Oorschot. Security and Usability Challenges of Moving-Object CAPTCHAs: Decoding Codewords in Motion, 2012
- [16] E. Bursztein, S. Bethard, C. Fabry, J.C. Mitchell, and D. Jurafsky. How Good are Humans at Solving CAPTCHAs? A Large Scale Evaluation, 2010
- [17] Wayne Hawkins, <https://www.change.org/p/it-s-time-to-finally-kill-captcha-0da5aea6-50d9-4b9a-b54d-f30d4f1e9478>, 2013
- [18] web.archive.org/web/20160207073026/webnographer.com/blog/2015/04/captchas-how-bad-are-they-really-for-user-experience/, Apr 2015
- [19] <http://www.donotlick.com/2015/01/05/8-increase-in-reddit-account-registrations/>, Jan 2015
- [20] <http://acquireconvert.com/conversion-optimization/>
- [21] H. Brignull. <http://www.90percentofeverything.com/2011/03/25/fk-captcha/>, Mar 2011
- [22] <https://trends.builtwith.com/widgets/captcha>, Apr 2017
- [23] <https://trends.builtwith.com/widgets/reCAPTCHA>, Apr 2017
- [24] <https://www.scrapesentry.com/scraping-wiki/need-know-captcha/>, 2016
- [25] L. von Ahn, M. Blum, N.J. Hopper, and J. Langford. CAPTCHA: Using Hard AI Problems For Security, 2013
- [26] Brad Yale. The CAPTCHA: A History, A Problem, Possible Solutions, <http://www.informit.com/blogs/blog.aspx?uk=Why-Are-CAPTCHAs-So-Awful>, Sep 2014
- [27] Nasser Mohammed Al-Fannah. Using Aesthetic Judgements to Distinguish between Humans and Computers. Apr 2017

LIST OF FIGURES

| | | |
|----|---|---|
| 1 | The popularity of reCaptcha | 1 |
| 2 | Text-based CAPTCHA | 2 |
| 3 | Audio-based CAPTCHA | 2 |
| 4 | Image-based CAPTCHA | 3 |
| 5 | Origin of human operators | 3 |
| 6 | Negative kerning | 4 |
| 7 | Image annotation modules | 5 |
| 8 | Anti-Captcha: CAPTCHA solved | 6 |
| 9 | Anti-Captcha: reCaptcha checked | 6 |
| 10 | CaptchaSolutions: The results from 9 CAPTCHAs | 6 |