LARGE SCALE CAPTCHA SURVEY

by

Mecheal Greene

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Summer 2018

LARGE SCALE CAPTCHA SURVEY

by

Mecheal Greene

Approved: _____
Haining Wang, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____
Douglas J. Doren, Ph.D.
Interim Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

First, I would like to thank the Bridge to Doctorate program for funding me and allowing me the opportunity to pursue my Masters degree at the University of Delaware, immediately following my graduation from Cheyney University. I especially want to thank Dean Doctor Michael Vaughan for being the on campus advisor of the program and assisting me whenever a problem arose.

My advisor, Doctor Haining Wang, for guiding me on the completion of my Masters thesis and degree. You allowed me to grow and learn throughout my graduate career. Most, importantly you took a chance on a young black man that you didnt even know but still you embraced me with open arms and worked with me towards accomplishing this goal.

My partner on this project Alparslan Sari, you helped me and guided me every step of the way whether it was through meetings, conversations or emails. You knew this was my first time doing anything thesis related but you made it all easier on me and didnt mind helping me above and beyond what I even imagined.

A special thanks to my mom and dad for their unconditional love and support throughout my education and anything else I involved myself in. Mom you always pushed me, even when I didnt want to push through anymore. As always you were right I could accomplish this feat and I have done just that. I also would like to thank the rest of my family for their encouragement, love and guidance.

Last but not least; I would like to thank my Grandfather Clarence Hollis who recently passed in March of this year. I will never forget the unconditional love and support you always showered me with ever since I can remember. I know through all of my doubts you always told me I could accomplish anything and everything I put my mind to. I will always love you and miss you Pop-Pop.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this research, we scanned the top 30,000 Alexa web pages to find out how many web pages are using captcha systems. Our other goal was to classify the captcha types and evaluate the known captchas to determine if they have any kind of weaknesses or vulnerabilities. We designed a web crawler that utilized the Beautiful Soup library to parse the top 30,000 web pages and find evidence of captchas in the URL of the web pages by looking for keywords such as login, cart, subscribe, password, sign, register, join, auth, upload, account and registration. After scanning the top 30,000 web pages we discovered that only 10,017 of the web pages are using captcha systems. The captchas that we discovered were audio-based, image-based, text-based, captcha, reCaptcha, FunCaptcha, slider, math, custom and text/image-based captchas.

Chapter 1

INTRODUCTION

Completely Automated Public Turing Tests to tell Computers and Humans Apart (Captchas) [1]. Captchas are programs and/or systems that are intended to distinguish humans from machine input, typically as a way of thwarting spam and automated extractions of data from websites. The ways captchas can determine the differences between humans and bots (computers) are by implementing a test or challenge that is designed to be solved quit easily by humans but yet very complicated and difficult by bots.

These forms of tests have changed and improved throughout the years. The first form of captcha was developed in the early 2000s by researchers at Carnegie Mellon University, and it was a distorted text based captcha, where the user would have to input the correct term that was being distorted. It has only been 18 years but the captcha tool has changed a lot. Now there are not only text based captchas but image based captchas. Multiple image based captchas, checkbox based captchas, 3D captcha, logic captcha, audio based captchas, and now even mini-game captchas are being proposed and developed. Most companies that use captcha prefer to use reCaptcha because it is a more improved version of captcha. There is no significant difference between captcha and reCaptcha, it comes down to the companys preference. Most websites and apps that use captcha prefer to use a form of reCaptcha [9].

1.1 Usage of Captchas

Many different types of websites use captchas and they also use captchas for different reasons. Free email services (Google, Yahoo, Ask, Microsoft, etc.) use captchas to prevent bot attacks. Otherwise, the bots would create thousands of email accounts

every minute. The bot attacks have become less of an issue because these websites have used captchas. Social media websites (Twitter, Facebook, Instagram, Snapchat, etc), vendoring websites (Playstation App, Xbox Smartglass App, TicketMaster) and banking websites (PSECU, Wells Fargo, PNC, TD Bank, etc.) all suffer from similar attacks.

1.2 Problem Definition

Attacks on captcha systems can be categorized as "Machine learning based attacks", "Cheap or unwitting human efforts", "Insecure implementation or misconfiguration of captcha systems", and other attacks. In the real world, we do not know how many web pages have deployed the captcha mechanism as part of their security practices. Furthermore, it is not known how many captcha types are used and how secure they are. In this research, we will scan the top 30,000 Alexa web pages to find out how many web pages are using captcha systems. The other goal of this work is to classify the captcha types and finally, we will evaluate the known captchas to determine if they have any kind of weaknesses or vulnerabilities. This research will give us demographic information in captcha related environments.

The remainder of the thesis is organized as follows. In Chapter 2, we describe the background and literature review. In Chapter 3, we present captchas in detail, such as the various types, security perspectives and breaking tools. In Chapter 4, we describe the study methodology. In Chapter 5, we present our findings and results from the data set. Finally, in Chapter 6, we conclude and in Chapter 7, we discuss future work.

Chapter 2

BACKGROUND

Captcha is an acronym that stands for Completely Automated Public Turing Tests to tell Computers and Humans Apart [1], [9]. The original captcha system was developed in the early 2000s by researchers at Carnegie Mellon University. The research team was led by Luis Von Ahn, who wanted to find a way to filter out the outstanding number of spam/bots pretending to be humans [9]. They developed a program that would show the user a form of distorted text that a bot cannot possibly read but yet a form of text that a human could decipher. The user had to simply type the text in a box and then access would be granted to the user.

Text-based captchas are the most commonly used captchas and could be designed in many different ways. Text-based captchas can use multi-font, which is multiple types of fonts and font-faces, a charset which is a 128 alphanumeric set of characters, variable font sizes, distortion, which in this case means distorting either the text or the background of the text, blurring the letters of the text, rotating tilted characters into different angles, and waving that gives the text a wave like pattern [13].

The text-based captchas yet mostly suffer from many vulnerabilities. For instance (A Simple Generic Attack on Text Captchas[7]), the authors used a Gabor Filter machine learning attack on the top 20 websites that used text based captchas. The Gabor Filter would extract character components of the text based captchas along four different directions then use partition and recognition to try a combination of different adjacent components and then try the most likely combination as the correct choice [7]. Another vulnerability was found by the authors of "Generic solving of text-based captchas"[3]. They used their machine learning algorithms against various websites, such as CNN, Baidu, eBay, Wikipedia, and Yahoo, where reCaptchas were broken very

easily. The highest success rate was Baidu at 55% and their lowest success rate was Yahoo at 5%. Our last example of the text-based captchas vulnerabilities was found by the researchers who explored two machine learning algorithms on Gimpy and EZ Gimpy text-based captchas [12]. Gimpy and EZ-Gimpy are two different types of text captchas, EZ-Gimpy is a text-based captcha that has noise in the background with different color sequences and patterns [12]. The Gimpy text-based captcha distorts colors and also has multiple words in its captchas to thwart machine learning based attacks [12]. The two machine learning algorithms achieved high success rates on both the EZ-Gimpy and the Gimpy text-based captchas, the EZ-Gimpy success rate was over 90% and the Gimpy success rate was over 30%[12].

Image-based captchas are also very commonly used and vary in types. There are single image-based captchas, multiple image-based captchas and text image-based captchas. There are even new types of image-based captchas being developed. For instance, there are Symmetry image-based recaptchas being developed, which researchers believe are a positive alternative to image-based captchas/recaptchas because they are more powerful and more secure than the original image-based captchas [6]. Another alternative to the traditional image-based captchas is Asirra (A captcha that exploits Interest-aligned Manual Image Categorization) [5]. Asirra is an image-based captcha that has the user pick from a set of 12 images and the user has to pick which images are cats and which are dogs [5]. The Asirra captcha is also statistically proven to be more secure than its counterparts because it is harder for machine learning algorithms to break into its image databases [5]. Another alternative to image-based captchas is the "Whats Up Captcha? A Captcha based on image orientation" [8]. The researchers are developing a captcha that will display an image and then the user must rotate it to its upright position [8]. These image rotation captchas are harder for bots to figure out through machine learning algorithms compared to their text/image-based captcha counterparts [8]. The traditional image-based captcha and the alternatives that are being developed still suffer from vulnerabilities, which are mainly machine learning based and brute force attacks.

Audio-based captchas are not common but they are still used by some websites. There are different types of audio captchas/recaptchas. Audio-based captchas suffer from machine learning, brute force and laundry attack vulnerabilities. Previous research done by the developers of DeCaptcha (Failure of Noise Based Non-Continuous Audio Captchas) has proven that audio-based captchas can easily be broken by brute force and laundry attacks [2]. The researchers tested their audio-based captcha breaker on various email providers and the Decaptcha program was more than 40% effective against all of the audio-based captchas [2].

Mini-Game/Dynamic captchas are the latest types of captchas that are being developed. It is believed that dynamic captchas that are interactive are a better alternative than the usual image/text-based captchas because they are harder for machine learning algorithms to break. Researchers have been developing "CaptchaStar" [4], which is a dynamic captcha that causes the user to produce a shape in a confusing environment. Dynamic Cognitive Game captchas [10], [11] are captcha games that the user must move 1,2,3 or even more objects to solve the captcha. The mini-game/dynamic captchas have improved security results, yet they still suffer from relay attacks, brute force attacks, automated attacks, and stream relay attacks.

Chapter 3

CAPTCHA IN DETAILS

In this chapter, we present the different captcha types, vulnerabilities that each type suffers from, security perspective of black box/gray box testing and tools known to break captchas.

3.1 Captcha Types

In this section, different captcha types are introduced, their vulnerabilities are described, visual images of each captcha type and the URL of their developers are provided.

3.1.1 Audio-Based

A tool that will protect your website from spam, bots and abuse. This captcha requires the user to listen and identify either numbers or words that were spoken. Vulnerabilities are machine learning algorithms, brute force attacks, third party attacks, laundry attacks and optical character recognition. Examples of audio-based captchas are shown in Figures 3.1 and 3.2. https://www.google.com/recaptcha/intro/v3beta.html https://captcha.org

3.1.2 reCaptcha

A free Google service that protects websites from bots, spam and abuse. The most recent vulnerability of reCaptchas is that their captchas were bypassed by attackers who used google's own web tools, but this was fixed in the latest update to reCaptcha v3. The other vulnerabilities include machine learning algorithms, brute force attacks, optical character recognition and third-party attacks. Google cookies could be

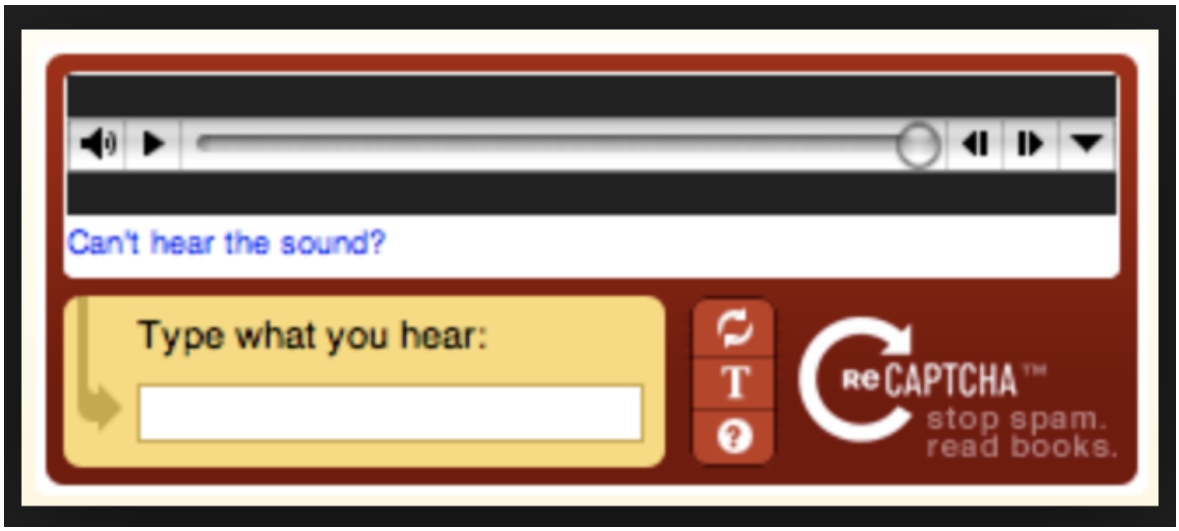

Figure 3.1: Audio Recaptcha https://patentyogi.com/latest-patents/google-filed-patent-for-generating-a-3d-audio-captcha/

Figure 3.2: Audio Captcha http://nowcaptcha.blogspot.com/2017/01/how-to-solve-audio-captchas.html

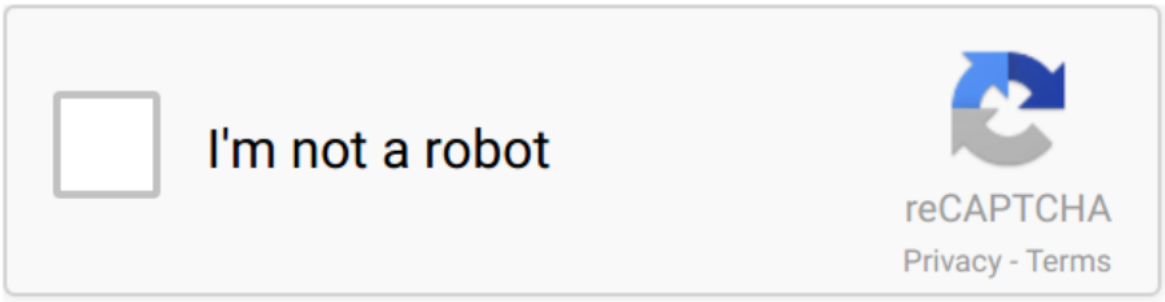

Figure 3.3: Check Box reCaptcha https://bestwebsoft.com/captcha-vs-recaptcha-what-to-choose/

used to skip all reCaptcha verification challenges prior to reCaptcha version 3. Examples of reCaptchas are shown in Figures 3.1, 3.3, 3.4 and 3.5. https://www.google.com/recaptcha/intr https://captcha.org

3.1.3 Image-Based

A tool that will protect your website from spam, bots and abuse. Vulnerabilities are brute force attacks, relay attacks, third party attacks and optical character recognition. Examples of image-based captchas are shown in Figures 3.4, 3.5 and 3.6. https://wordpress.org/plugins/image-captcha/ https://captcha.org

3.1.4 Text/Image-Based

A tool that will protect websites from spam, bots and abuse. Vulnerabilities are brute force attacks, relay attacks, third-party attacks and optical character recognition. Examples of text-based captchas are shown in Figures 3.7, 3.8, 3.9, 3.10 and 3.11. https://captcha.org

3.1.5 Slider Captcha

A simple swipe is used to validate that the user is a human and protect the websites from potential spam, bot attacks and abuse. It can be implemented on comment

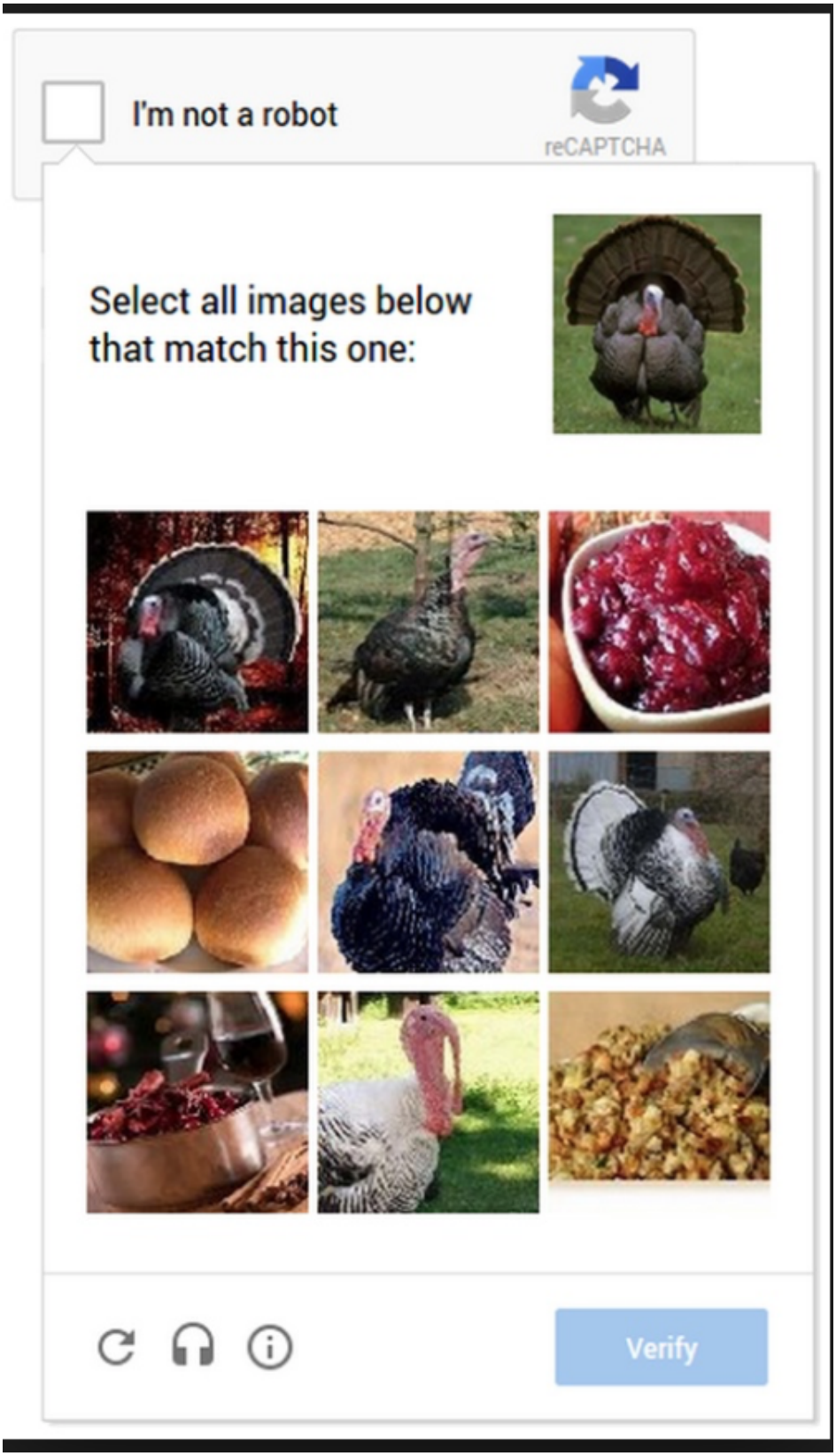

Figure 3.4: Multiple Image Captcha https://www.krishaweb.com/blog

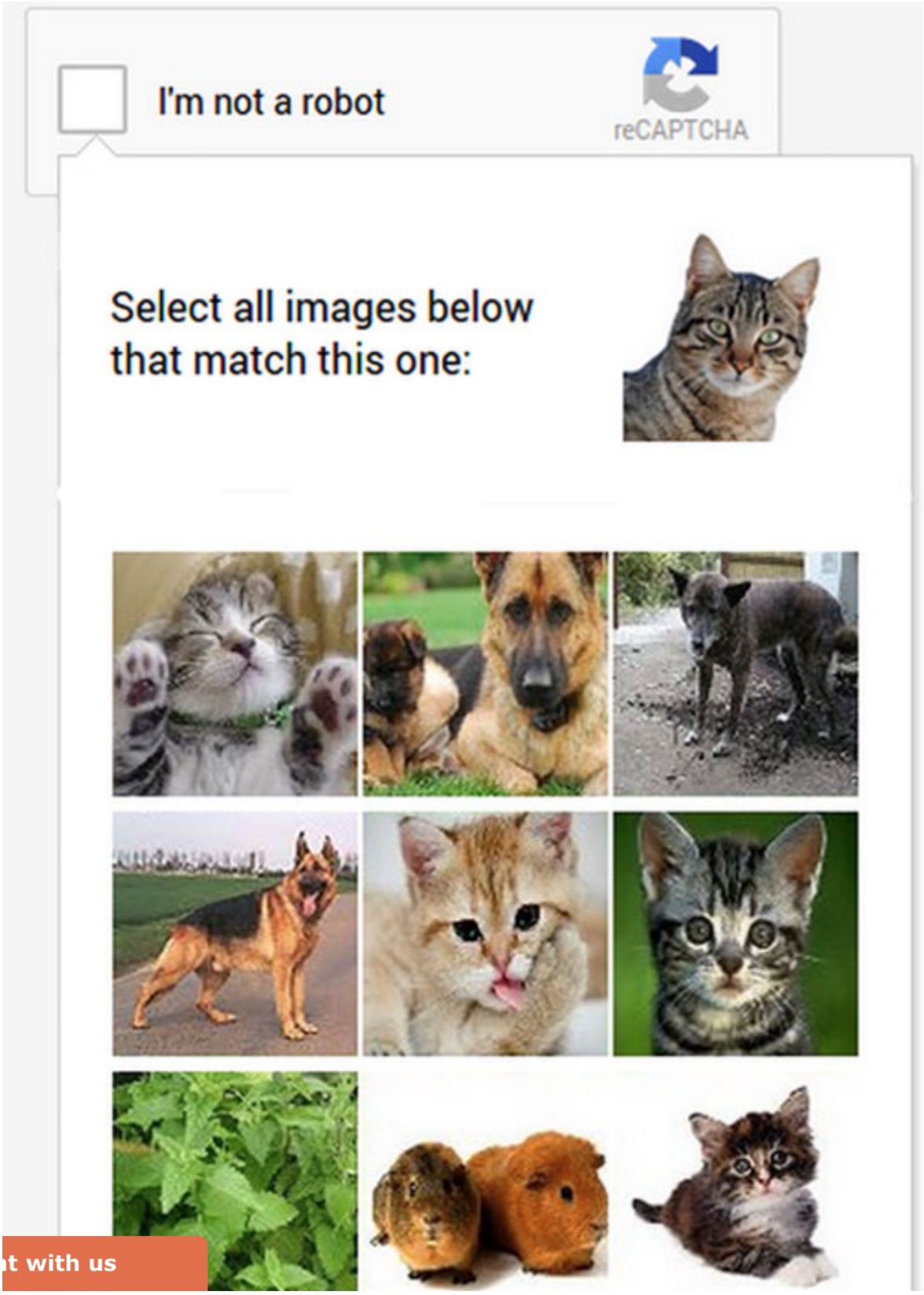

Figure 3.5: Multiple Image Captcha www.krishaweb.com/blog

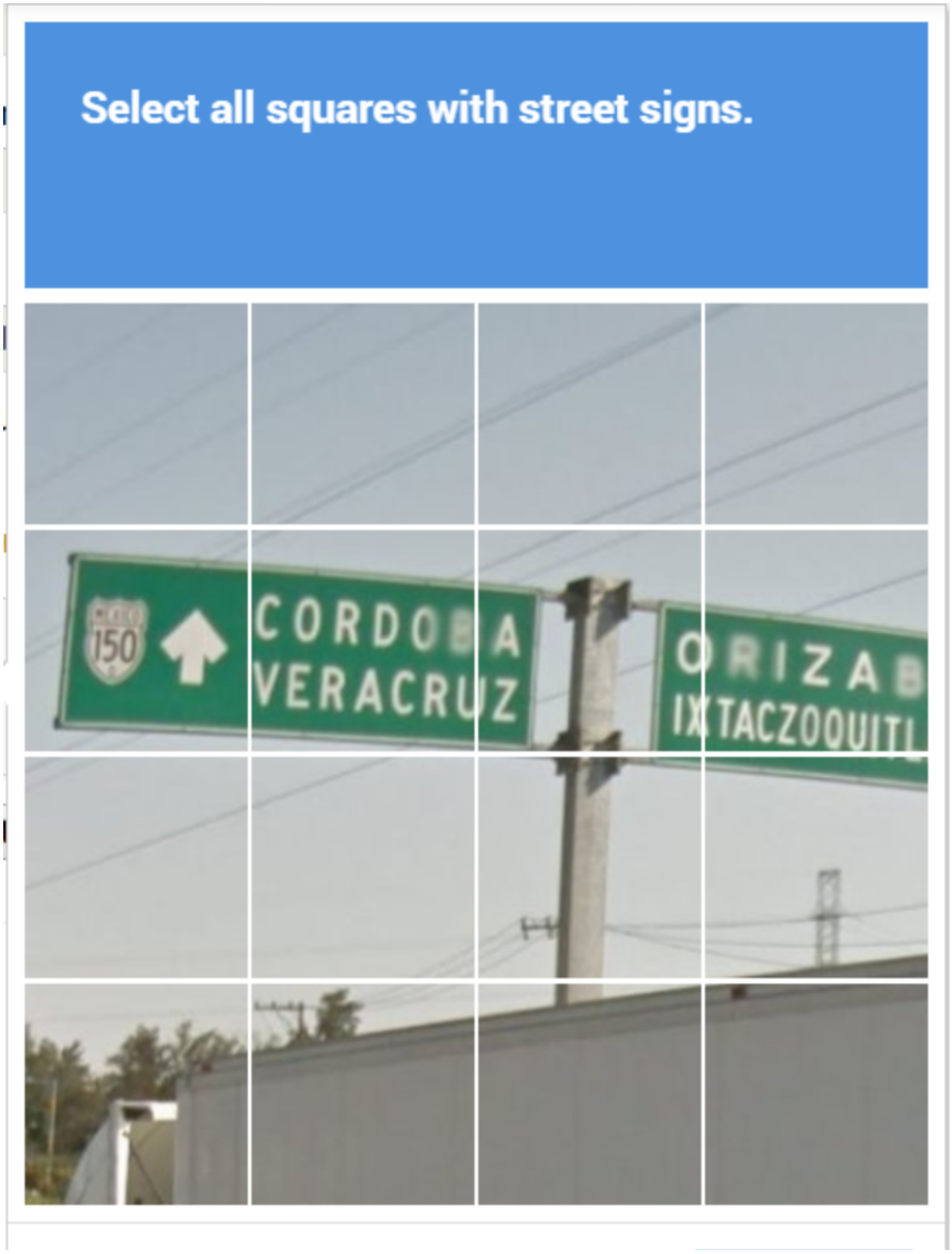

Figure 3.6: Multiple Selection Image Captcha https://blog.desdelinux.net

Figure 3.7: Spotted Background Text Captcha. https://www.accessibilityoz.com/ozwiki/captcha-alternatives-and-accessibility/

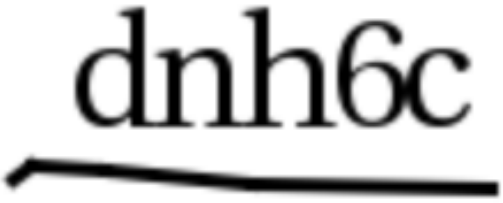

Figure 3.8: Underline Text Captcha https://access.line.me/dialog/captcha/SGH8L1WTmrRzSAeKjC52YEL5OOALKHmjYNg7xgR3TNG

Figure 3.9: Wave Pattern Text Captcha https://www.webdevelopersnotes.com/captcha-definition-why-are-you-asked-to-type-the-two-words

Figure 3.10: Wave Pattern & Ink Spot Background Text Captcha https://www.webdevelopersnotes.com/captcha-definition-why-are-you-asked-to-type-the-two-words

Figure 3.11: Wave Pattern Text Captcha https://www.webdevelopersnotes.com/captcha-definition-why-are-you-asked-to-type-the-two-words

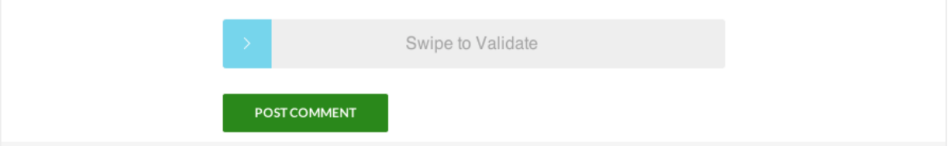

Figure 3.12: Slider Captcha https://wordpress.org/plugins/slider-captcha/

pages, registration, login, lost passwords, mailpress, contact forms and custom localizations. However, it is vulnerable to third party attacks. Examples of slider captchas are shown in Figures 3.12 and 3.13. https://wordpress.org/plugins/slider-captcha/

3.1.6 FunCaptcha

A Dynamic 3D captcha that requires the user to rotate the image to the right side up, also it is invulnerable to brute force attacks, machine learning attacks and optical character recognition. Examples of FunCaptchas are shown in Figures 3.14 and 3.15. https://www.funcaptcha.com

3.1.7 MiniGame/Dynamic Captcha

Dynamic interactive game captchas vary from rotating images to moving objects or multiple objects, which are vulnerable to machine learning algorithms, third-party

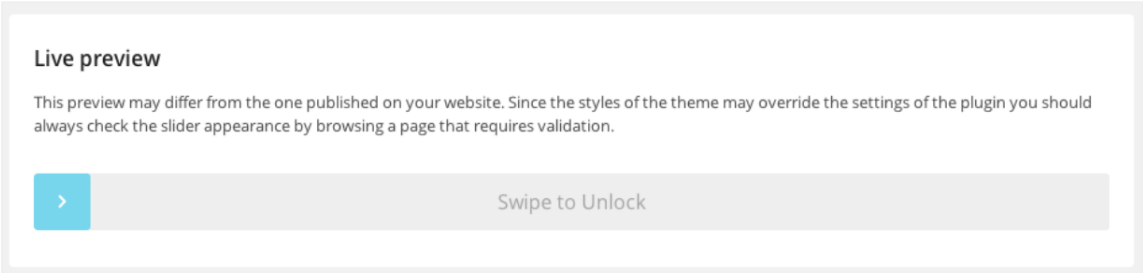

Figure 3.13: Slider Captcha https://wordpress.org/plugins/slider-captcha/

Figure 3.14: FunCaptcha https://www.funcaptcha.com/press-articles

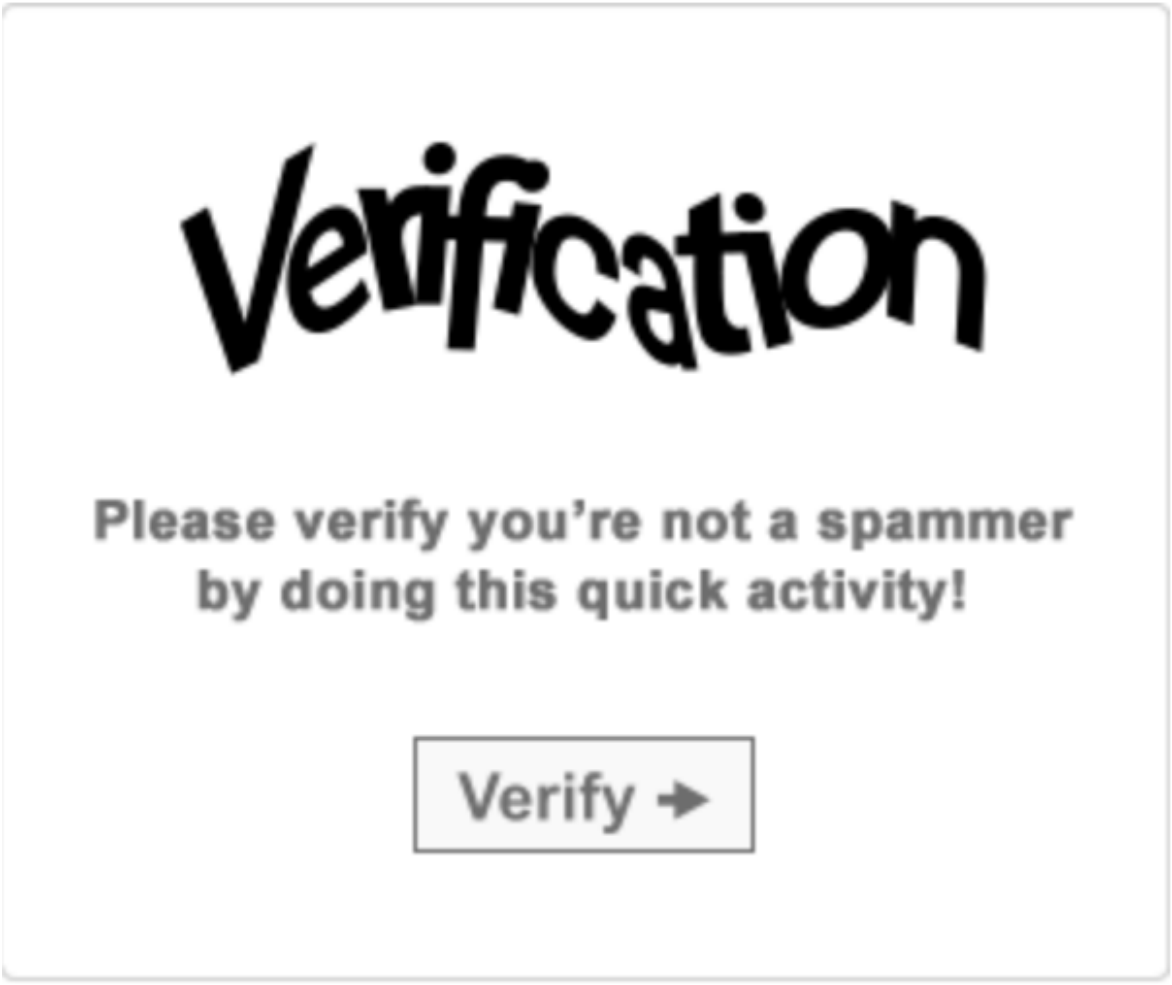

Figure 3.15: FunCaptcha Verification https://www.funcaptcha.com/press-articles

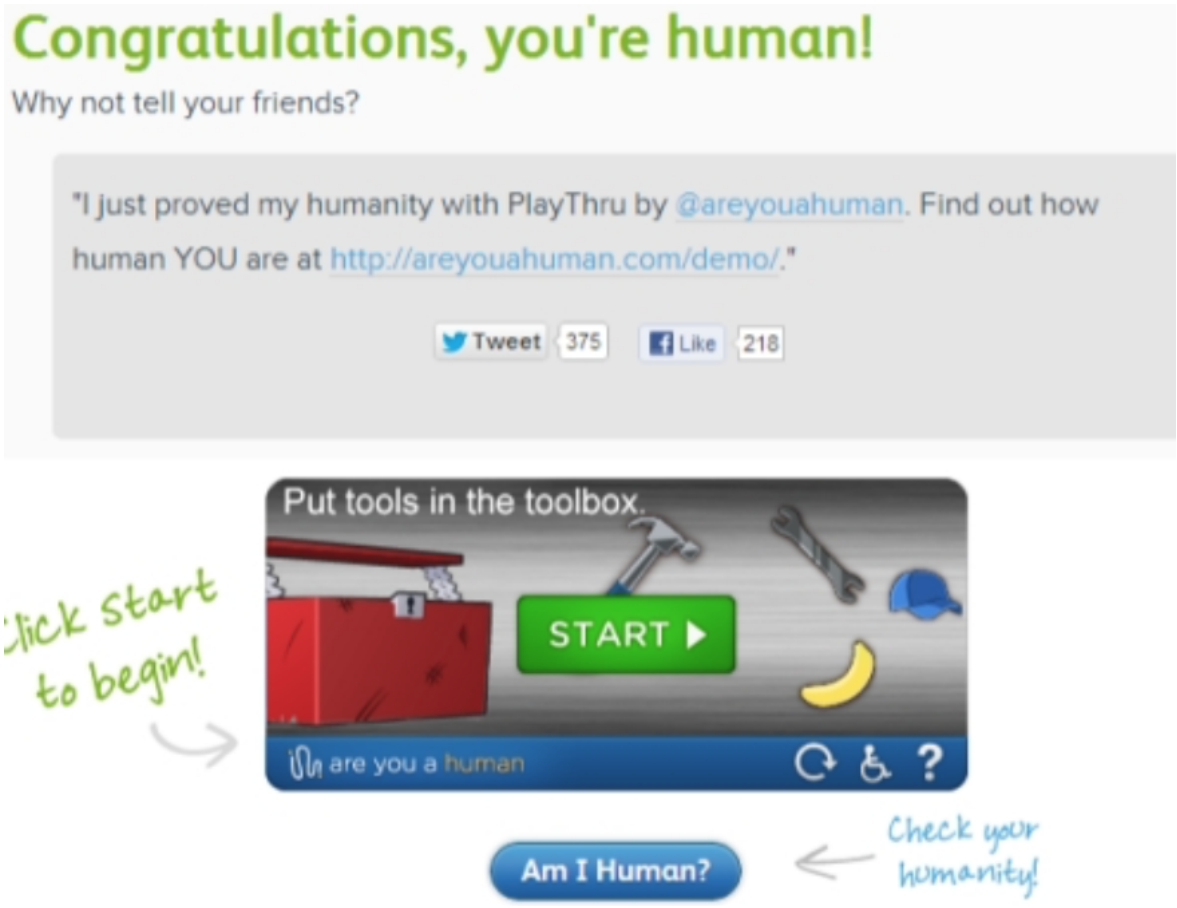

Figure 3.16: Mini Game Captcha https://www.tnooz.com/article/its-a-captcha-but-not-as-we-know-it/

attacks, brute force attacks, stream relay attacks and automated attacks. Examples of Mini-game/Dynamic captcha are shown in Figures 3.14 and 3.16.

3.2 Security Perspective

Black Box testing is a way to break captchas, it uses an intercepting fault injection proxy (usually WebScarab). The fault injection is used to identify all possible parameters that are sent, in addition to the decoded captcha value from the client to the server. The parameters usually contain encrypted or hashed values of decoded captcha and captcha ID numbers. The fault injection will attempt to send old decoded captcha values with an old captcha ID and it will also try to send old session IDs as

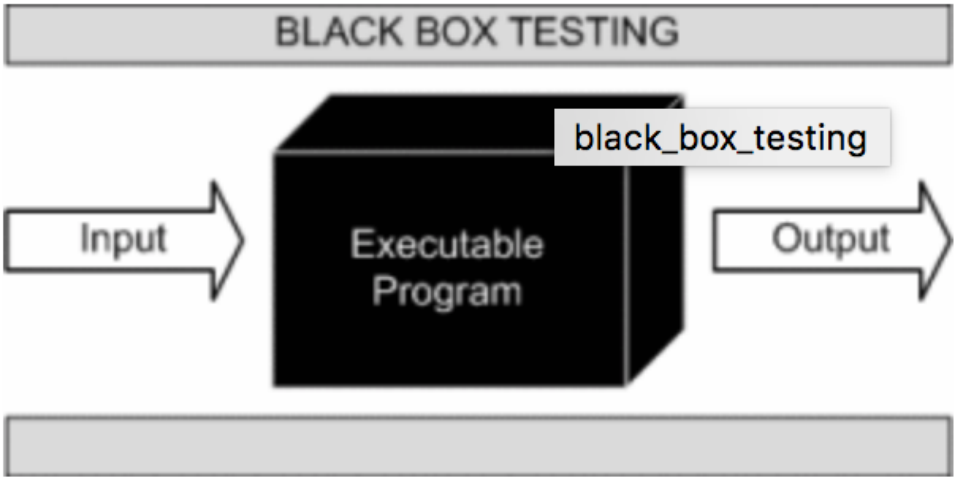

Figure 3.17: Black Box Testing http://softwaretestingfundamentals.com/black-box-testing/

well. If the application accepts either the captcha or session ID, it will be vulnerable to replay attacks.

Gray Box testing is a way to audit the application source code, so that the user can determine whether the application uses a form of captcha implementation and which version it uses. If the application sends encrypted or hashed values from the client, the gray box can also verify if the used encryption or hash algorithm is strong or not.

3.3 Breaking Tools

3.3.1 Tesseract

Tesseract is an open source optical character recognition engine, since 2006 it has been being developed by Google and its latest release was in June of 2017. It can be used to break reCaptcha, text-based, image-based, text/image-based and audio-based captchas. https://github.com/tesseract-ocr/tesseract

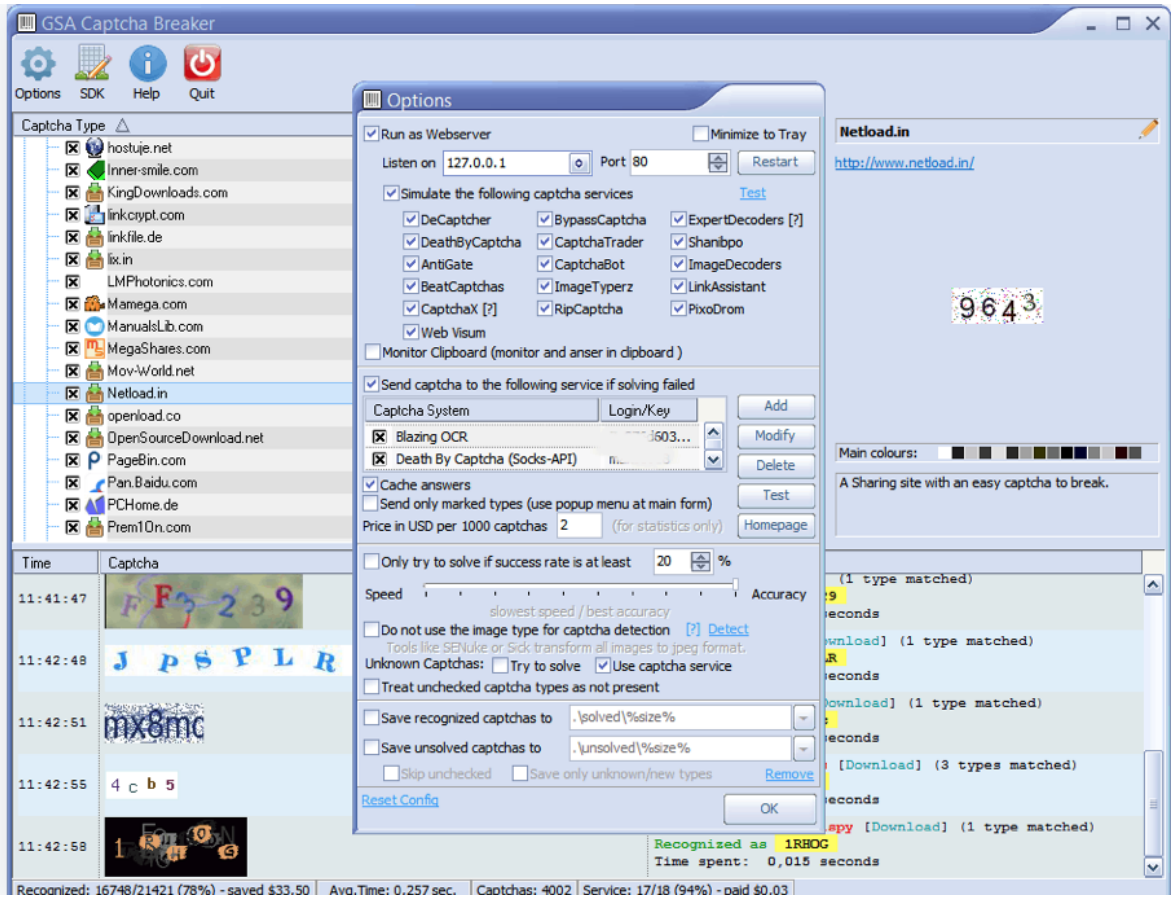

Figure 3.18: Figure shows the GSA Captcha Breaker tool in action.

3.3.2 GSA Captcha Breaker

GSA (German Software development Analytics) Captcha Breaker is a captcha breaking software that breaks almost any captcha type by using multiple optical recognition engines. It can be used to break reCaptcha, text-based, image-based, text/image-based and audio-based captchas. An example of GSA Captcha Breaker is Figure 3.18.

https://www.gsa-online.de

3.3.3 Decaptcha

Decaptcha is an audio-based captcha breaker that was developed in recent years. However, the audio-based captchas are weak and alternative audio captchas need to be developed to improve audio-based captcha security [2].

Chapter 4

STUDY METHODOLOGY

4.1 Data Collection

We first downloaded a file which contained the top 1 million Alexa web pages. The web page links were not complete, so we wrote a python script to get legitimate URLs of the top 1 million web pages. Once we were able to receive the correct links, we designed a web crawler to find evidence of captcha in the top 1 million web pages. We did not want to utilize the "Brute Force Attack" methodology while crawling as much as possible to respect resource owners and legal entities. Therefore, we designed the crawler with the following limitations:

The first limitation of the crawler is that it needs to utilize a heuristic function to evaluate each link in the crawled web page and then sends a request to access that web page. Based on our initial analysis, it is more likely to find evidence of captchas if the URL contained any of the following keywords: login, cart, subscribe, password, sign, register, join, auth, upload, account and registration. The second limitation of the crawler is that it should not parse each individual page for a given domain name. Therefore, we set a timer of 10 seconds as a stop condition, so that the crawler wouldn't get stuck on trying to crawl any one page for an extended amount of time. Moreover, if the crawler scanned 330 URLs and could not find any captcha specific information, it stops. If the crawler reached the stop condition while crawling a specific domain, it would then categorize that domain as captcha was not found. In addition, if the requested page has no HTML or text content, it is then ignored by the crawler. The last limitation of the crawler is if the requested page is bigger than 1MB, it would also be ignored by the crawler.

This script saved the successful links found in a file and then took log of the links that were not found into another file. It also logged the run time errors for us to fix at a later date. We set the following property so that the crawler would leave a signature behind as a FireFox browser while crawling:

```
headers['User-Agent'] = "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17
    (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"
```

The crawler also evaluated the found URLs in a requested page, and if any URL belonged to another domain, they would be ignored.

Our computer algorithm was:

1. crawl in a domain
2. retrieve the root page
3. check if the page has captcha or spambot keyword and if found record the url in a list, save the page content in a file for data analysis and then crawl another domain
4. all found urls will be evaluated and if the url belongs to another domain ignore that url
5. if any important keyword was presented in the url then put the url in a special list for crawling
6. continue crawling until the set stop condition is met.

The following libraries are used in the crawler code. We used the Beautiful Soup library for parsing the web page and for seeking captcha related evidence. The urllib2 library was used for retrieving URL content from the web pages.

```
import urllib2
import re
import sys
from bs4 import BeautifulSoup
from urlparse import urlparse

soup = BeautifulSoup(respData, "html5lib", from_encoding="utf-8")

headers['User-Agent'] = "Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.17
    (KHTML, like Gecko) Chrome/24.0.1312.27 Safari/537.17"
```

```
req = urllib2.Request(url, headers = headers)
```

4.2 Data Analysis

The data analysis required a lot of manual effort and we scanned the collected files to identify evidence related to captcha. We then wrote an analyzer script in python to make the script complete the classification process and be able to document each specific captcha that was used by the web pages security systems. If we found solid evidence of captcha in the initial data set such as google.com/reCaptcha or https://www.gstatic.com/recaptcha/api2 js libraries, it meant the page was using a version of reCaptcha. We then built a python hashmap to collect evidence and scanned through the data set manually to classify the captcha types. After each manual analysis, we then updated the script so that in each iteration the classifier would label the learned captcha types.

```
captcha = {
        "google_recaptcha":"google.com/recaptcha",
                "gstatic_recaptcha" : "https://www.gstatic.com/recaptcha/
                    api2",
        "investor" : "http://investor.salesforce.com/q4api/v1/captcha",
        "custom:chase": "/captcha/jpm-captcha.js",
        "captcha_ajax.jsp":"captcha_ajax.jsp",
        "captcha.aspx":"captcha.aspx",
        "custom3":"captcha.php",
        "Amazon2": "https://s3.amazonaws.com/ss-captchas/",
        "captchaAudio" : "captchaAudio",
        "captcha.cgi":"captcha.cgi",
         "antibot2.php":"antibot2.php",
        "antibot.php":"antibot.php",
        "securimage_show.php":"securimage_show.php"
        }
```

Chapter 5

RESULTS

After we scanned the top 30,000 Alexa web pages, we found that only 10,017 of the web pages were using captcha systems. As shown in Figures 5.1 and 5.2, the majority of captchas being used are versions of reCaptcha. The versions of reCaptcha ranged from versions 1, 2, 3 and 4. The different types of captchas that were found during the scan were audio-based captchas, captchas, image-based captchas, custom captchas, text-based captchas, text/image-based captchas, and others. The least used types of captchas were audio-based captchas, image-based captchas, and text-based captchas. Astonishingly, there were only 4 websites out of the 30,000 websites using audio-based captchas in their captcha systems, 337 websites using image-based captchas and 483 websites using text-based captchas.

There were more than 2,000 cases of Math.random Java script functions present in captcha related pages that were found in our scan of the top 30,000 websites. In the cases of Math.random that were found in the source code of the websites, we believed that the web pages have some sort of math function that is generating equations and answers randomly or image captchas that require random numbers to be generated. Every case of Math.random also had evidence of other captchas. For instance; there was evidence of image-based captchas, reCaptcha, text-based captcha, and custom captchas in the source code of the websites. We discovered quite a few interesting findings in the survey results. For instance, many of the top websites have their own custom captchas. We found over 700 websites that have their own custom captchas. They were all very similar though, most were either labeled custom, .cgi, .php, .js, .asp or .aspx. Also, there were only 7 cases of FunCaptcha and 2 cases of slider captcha throughout the whole data set. We first thought there would be more cases of FunCaptchas because it

Type	Count
AudioBased Captcha	4
Captcha	50
Custom Captcha	716
ImageBased Captcha	337
Other	2840
reCaptcha	5155
TextBased Captcha	483
Total	9585

Table 5.1: Large Captcha Survey Results For Single Captcha Systems

Type	Count
Audio-Based Captcha	4
Captcha	50
Custom Captcha	716
Image-Based Captcha	337
Other	2840
reCaptcha	5155
Text-Based Captcha	483
Text/Image-Based Captcha	432
Total	10017

Table 5.2: Large Captcha Survey Results For Single And Multiple Captcha Systems

is a new 3D dynamic captcha and it is also one of the more secure versions of captcha that can be easily implemented on websites. We also predicted to find more evidence showing web pages were implementing more than one type of captcha on their websites. We came to learn that only a few of the top 30,000 websites use more than one type of captcha on their websites. The combination of captchas that were being used were text/image-based captchas and only 432 websites were using them in their captcha systems.

The "other" category that is present on Figures 5.1 and 5.2 mostly contained traces of captcha found on the websites that were scanned but yet didn't have any known captchas when they were crawled. For instance, we received results that were captchaarea, bxcapthca, nucaptcha and so on. So we decided to label them as other because they didn't fall under any known captcha categories. We also discovered that some web pages we believed contained captcha security systems, in fact, didn't have any evidence of captcha systems when we checked the websites source code. We decided to label those cases as false positives and there were in fact a total of 887 cases of false positives in our data set.

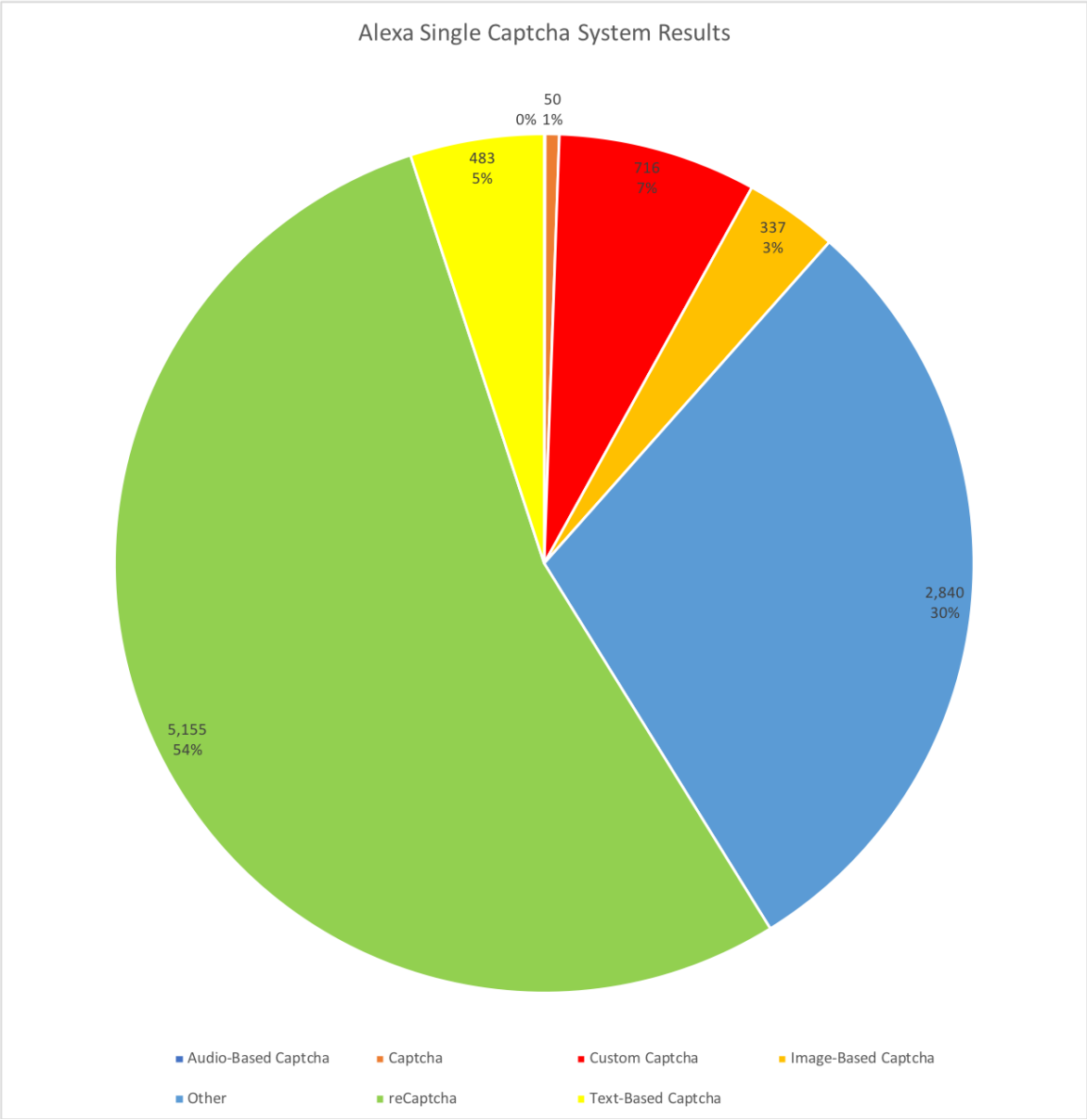

Figure 5.1: Survey Results of all web pages that only had one single type of captcha

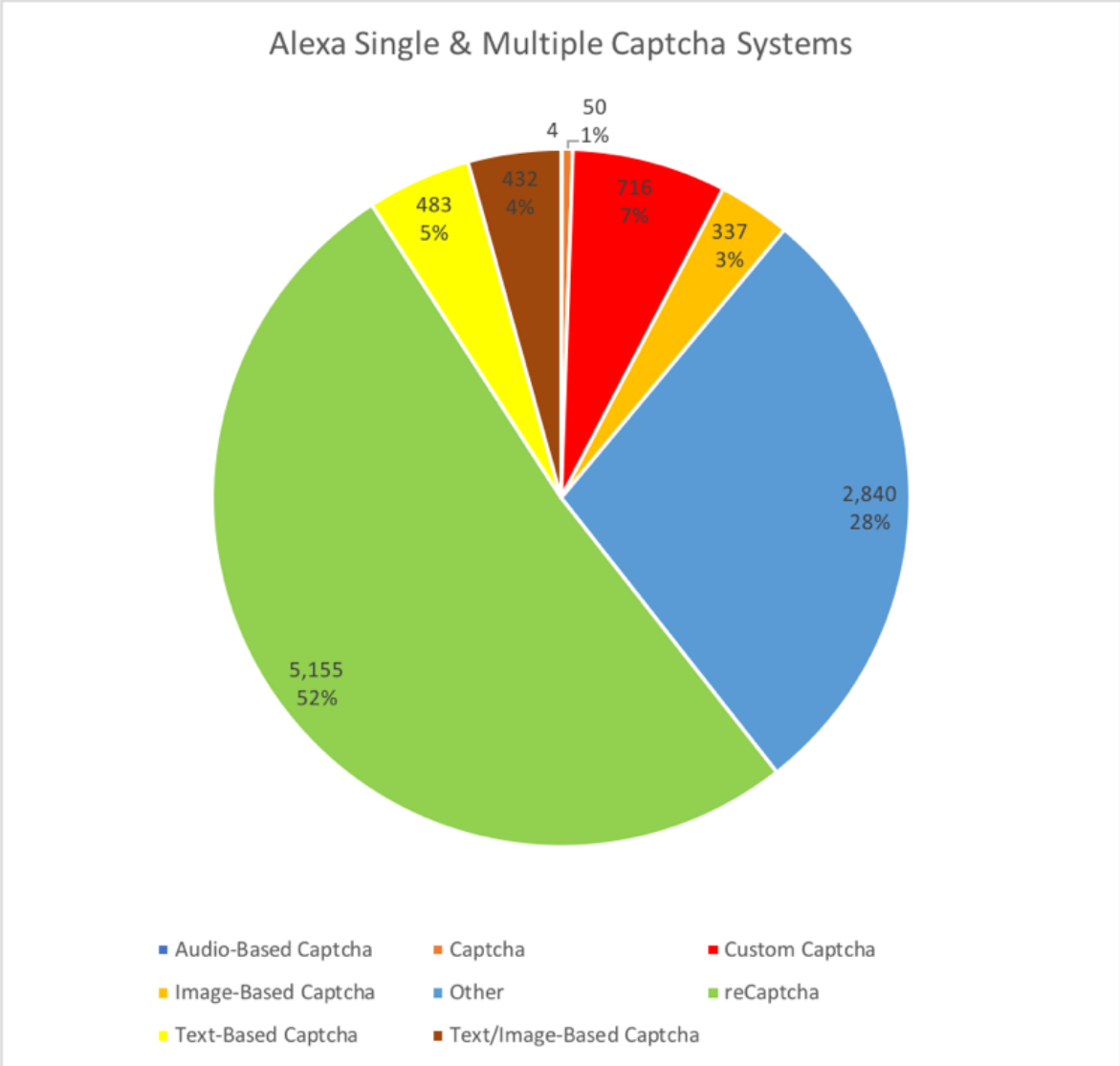

Figure 5.2: Survey Results of all web pages that had one or more types of captchas

Chapter 6

CONCLUSION

We took the 10,017 captchas that were being used by the top websites and surveyed the captchas in detail. We presented the captcha types, security perspectives for all the captchas and proposed known tools/methods to break every type of captcha that was implemented in their security systems. We thoroughly described our study methodology, how we collected our data and how we analyzed that data. Then we interpreted the data that we retrieved by crawling the top web pages and presented the facts and findings that were found throughout the large data set.

In this thesis, we have scanned the top 30,000 Alexa web pages and discovered how many web sites use captcha mechanisms in their security systems. We also learned what types of captchas are being used. We successfully classified the captcha types that are being used. In addition, we also determined what weaknesses or vulnerabilities of each type of captcha systems suffered from.

Chapter 7

FUTURE WORK

In the future, we would like to expand upon our current data set of 30,000 websites by scanning the top 100,000 Alexa websites and conducting a measurement study on how many of those websites are using captcha systems. Eventually; we would like to perform a vulnerability analysis on the 100,000 website data sets by trying to break/exploit every weakness/vulnerability of each websites current captcha systems.

REFERENCES

[1] Luis Von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. In *Proceedings of the 22Nd International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'03, pages 294–311, Berlin, Heidelberg, 2003. Springer-Verlag.

[2] E. Bursztein, R. Beauxis, H. Paskov, D. Perito, C. Fabry, and J. Mitchell. The failure of noise-based non-continuous audio captchas. In *2011 IEEE Symposium on Security and Privacy*, pages 19–31, May 2011.

[3] Elie Bursztein, Jonathan Aigrain, Angelika Moscicki, and John C. Mitchell. The end is nigh: Generic solving of text-based captchas. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, San Diego, CA, 2014. USENIX Association.

[4] Mauro Conti, Claudio Guarisco, and Riccardo Spolaor. Captchastar! A novel CAPTCHA based on interactive shape discovery. *CoRR*, abs/1503.00561, 2015.

[5] Jeremy Elson, John (JD) Douceur, Jon Howell, and Jared Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007.

[6] C. Funk and Y. Liu. Symmetry recaptcha. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5165–5174, June 2016.

[7] Haichang Gao, Jeff Yan, Fang Cao, Zhengya Zhang, Lei Lei, Mengyun Tang, Ping Zhang, Xin Zhou, Xuqin Wang, and Jiawei Li. A simple generic attack on text captchas. In *NDSS*, 2016.

[8] Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja. What's up captcha?: A captcha based on image orientation. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 841–850, New York, NY, USA, 2009. ACM.

[9] Kate Horowitz. The Surprisingly Devious History of CAPTCHA, 2016.

[10] Manar Mohamed, Song Gao, Nitesh Saxena, and Chengcui Zhang. Dynamic cognitive game captcha usability and detection of streaming-based farming. 2014.

[11] Manar Mohamed, Niharika Sachdeva, Michael Georgescu, Song Gao, Nitesh Saxena, Chengcui Zhang, Ponnurangam Kumaraguru, Paul C. van Oorschot, and Wei-Bang Chen. A three-way investigation of a game-captcha: Automated attacks, relay attacks and usability. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, pages 195–206, New York, NY, USA, 2014. ACM.

[12] Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'03, pages 134–141, Washington, DC, USA, 2003. IEEE Computer Society.

[13] Jeff Yan and Ahmad Salah El Ahmad. A low-cost attack on a microsoft captcha. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, pages 543–554, New York, NY, USA, 2008. ACM.

Appendix

SCRIPTS

The scripts used in this study can be found on following repository:

https://github.com/alparslansari/ca-crawler