Towards Evaluating the Security of Real-World Deployed Image CAPTCHAs

Binbin Zhao[1,*], Haiqin Weng[1,*], Shouling Ji[1,2] (✉), Jianhai Chen[1], Ting Wang[3], Qinming He[1],
Raheem Beyah[4]
[1]Zhejiang University, [2]Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies,
[3]Lehigh University, [4]Georgia Institute of Technology
{bbge, hq_weng, sji, chenjh919}@zju.edu.cn, ting@cse.lehigh.edu, hqm@zju.edu.cn, rbeyah@ece.gatech.edu

ABSTRACT

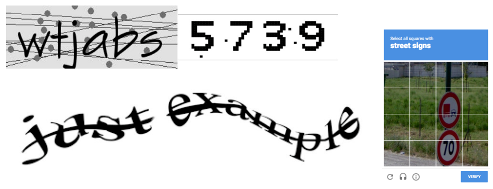
Nowadays, image captchas are being widely used across the Internet to defend against abusive programs. However, the ever-advancing capabilities of computer vision techniques are gradually diminishing the security of image captchas; yet, little is known thus far about the vulnerability of image captchas deployed in real-world settings.

In this paper, we conduct the first systematic study on the security of image captchas in the wild. We classify the currently popular image captchas into three categories: selection-, slide- and click-based captchas. We propose three effective and generic attacks, each against one of these categories. We evaluate our attacks against 10 real-world popular image captchas, including those from `tencent.com`, `google.com`, and `12306.cn`. Furthermore, we compare our attacks with 9 online image recognition services and human labors from 8 underground captcha-solving services. Our studies show that: (1) all of those popular image captchas are vulnerable to our attacks; (2) our attacks significantly outperform the state-of-the-arts in almost all the scenarios; and (3) our attacks achieve effectiveness comparable to human labors but with much higher efficiency. Based on our evaluation, we identify the design flaws of those popular schemes, the best practices, and the design principles towards more secure captchas. We believe our findings shed light on facilitating the ecosystem of image captchas.

ACM Reference Format:
Binbin Zhao[1,*], Haiqin Weng[1,*], Shouling Ji[1,2] (✉), Jianhai Chen[1], Ting Wang[3], Qinming He[1],, Raheem Beyah[4] . 2018. Towards Evaluating the Security of Real-World Deployed Image CAPTCHAs. In *AISec '18: 11th ACM Workshop on Artificial Intelligence and Security Oct. 19, 2018, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3270101.3270104

1 INTRODUCTION

CAPTCHA (Completely Automated Public Turing tests to tell Computers and Humans Apart) [48] is widely used to increase the security of websites. Generally, the popular captchas deployed in the real world can be classified as text and image captchas as shown in Figure 1. Image captchas require users to understand the images from received captchas and perform identification operations (e.g., select certain images, click certain regions) according to the guidance. Nowadays, image captchas are becoming increasingly popular for they are considered more user-friendly and more secure than text captchas. According to the report from `Tencent`'s captcha service, up to now, their image captchas have been used by ∼ 1 billion users [9]. `GEETest` [1], another captcha service, also reports that they provide image captchas for over 200, 000 top websites, including `tripadvisor.cn`, `airbnb.com`, `jingdong.com`, etc [10]. `Google` shows that millions of ReCaptcha challenges have been solved per day [11].



(a) Text captcha (b) Image captcha

Figure 1: Examples of text and image captchas.

Popular Image Captchas. Currently, popular real-world image captchas can be roughly classified into three categories: selection-based image captchas [25], slide-based image captchas, and click-based image captchas [33], as shown in Table 1. Selection-based captchas ask users to select candidate images with specific semantic meanings from sets of images. For instance, ReCaptcha, released by `Google` in 2014, is the most widely used selection-based captcha. Up till April 2018, about 0.5% of the entire Internet, 4.7% of the top 1M sites, 7.3% of the top 100K sites, and 10.9% of the top 10K sites use ReCaptcha to block abusive programs. Slide-based captchas request users to slide puzzle pieces to the right parts of images. For instance, Tencent SlidePuzzle, released by `Tencent`, is a typical slide-based captcha. It is employed by many large-scale web services, such as `qzone.qq.com`, which is reported to have 0.56 billion active users per month. Click-based captchas require users to click specific semantic regions in images. Both GEE TouClick and

*Binbin Zhao and Haiqin Weng are the co-first authors. Shouling Ji is the corresponding author.

[1] http://www.geetest.com/

AISec '18, October 19, 2018, Toronto, ON, Canada
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6004-3/18/10…$15.00
https://doi.org/10.1145/3270101.3270104

Netease TouClick are representative click-based captchas. In this paper, we investigate the security of all the above image captchas.

Status Quo. While image captchas are considered alternative superior to text captchas (e.g., richer information, larger variation spaces, more user-friendly), there are severe concerns about their security in many real-world settings.

First, the security and robustness of text captchas have been intensively studied by the research community [21, 23, 27]: many kinds of generic solvers, distortion or defensive techniques (e.g., rotation, distortion), and design guidelines for securing text captchas have been proposed. In comparison, limited studies have been conducted on the security of image captchas. Specifically, existing works focus either on synthetic image captchas [30, 39] or some special cases of captcha schemes (e.g., ReCaptcha 2015) [44, 49].

Second, the ever-advancing capabilities of computer vision and machine learning techniques gradually diminish the security of image captchas and make them vulnerable. It is reported that computers outperform human beings in many complex recognition tasks [31]. Exposed to such powerful techniques, image captchas might become vulnerable. For example, in 2016, Sivakorn *et al.* utilized deep learning techniques to break one of the most popular image captcha schemes, ReCaptcha [44]. Also, many commercial companies have deployed powerful online recognition services for various tasks, including image classification and object detection. Those services might be maliciously used by abusive programs to break image captchas.

Third, there exists a large-scale economically targeted underground market of captcha-solving services, which support almost all types of captchas and significantly threaten the security of captchas. For example, `ruokuai.com` provides services for breaking text, image and audio captchas, and its daily service requests exceed 900 million times. Moreover, the involved commercial values of some captcha-solving services reach O($million) scale, e.g., the income of the arrested service `qadati.cn` is reported as much as $3.18 million [12].

In the real world, image captchas become more and more popular. They have been employed by many of the world's top websites like `Google`, `Facebook` and `Tencent` to prevent abusive programs. However, we still lack sufficient understanding of their security and effectiveness. It is urgent to make a comprehensive evaluation on the security of image captchas for (1) understanding the vulnerability of image captchas, (2) designing more robust and secure image captchas, and (3) helping website providers defend against abusive programs.

Methodology. In this paper, we propose three effective generic attacks, *SelAttack*, *SliAttack*, and *CliAttack*, against selection-, slide-, and click-based image captchas, respectively. Our attacks are mainly built upon advanced vision techniques and a series of image classification and object detection models.

We evaluate our attacks on 10 real-world captcha schemes provided by top websites as shown in Figure 1, including `Google`, `Facebook`, `Tencent`, and `Netease`[2]. To our best knowledge, 7 out of the 10 schemes have never been broken before this work. Of the 10 schemes, our attacks achieve 45% − 70% success rate on three (GEE TouClick, Tencent TouClick, and Netease TouClick), 70% − 89% success rate on three (ReCaptcha 2015, ReCaptcha 2018, and Facebook), and 90% − 100% success rate on four (China Railway, GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle).

Then, we compare our 3 attacks with 9 popular recognition services provided by `Google`, `Microsoft`, `Tencent`, `Alibaba`, `Baidu`, and `Face++`. Compared with our attacks, those mainstream recognition services (except `Google`) do not provide satisfying attack results. Nonetheless, they are still able to break the tested captchas given that *a captcha scheme is broken when an attacker can reach a precision of at least* 1% [23]. Besides, for selection-based captchas, we also compare our attack with two state-of-the-art attacks [44], [49]. Note that for slide- and click-based captchas, we only run our attacks against them as *they are not reported to be broken before to our knowledge.* Again, the evaluation results suggest that our attack is more elegant in both of the success rate and speed compared with [44], [49].

Further, we employ human labors from 8 underground captcha-solving services, including `ruokuai.com` and `2captcha.com`, to manually break the same 10 real-world captcha schemes as evaluated by our attacks. Surprisingly, we find that our attacks outperform those of the most skilled human labors on 4 captcha schemes (China Railway, GEE SlidePuzzle, Tencent SlidePuzzle, Netease SlidePuzzle). For the remaining 6 schemes, our attacks achieve effectiveness and efficiency comparable to human labors.

Table 1 lists the captchas, existing state-of-the-art attacks, image recognition and captcha-solving services evaluated in this paper.

Contributions. We summarize our contributions as follows.

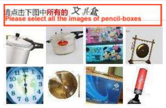
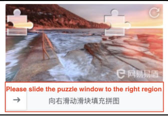
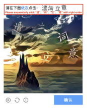
- *Security of Popular Image Captchas.* We implement 3 powerful generic attacks, which can be used to break a variety of captcha schemes. By conducting proof-of-concept studies, we have successfully attacked 10 real-world captcha schemes from popular websites, including `google.com`, `facebook.com`, `tencent.com` and`12306.cn`. We also test the effectiveness of popular image recognition services and underground captcha-solving services. The evaluation results suggest that our attacks achieve effectiveness comparable to human-based captcha-solving services in terms of attack duration and cost-effectiveness.
- *Countermeasures Towards Secure Image Captchas.* Based on our evaluation and findings, we have identified several design flaws of popular real-world captcha schemes, such as many of real-world captchas repeatedly use the same images or do not apply advanced anti-recognition techniques. We also distill our attacks, evaluation results, and identified design flaws into a set of best practices and design principles towards developing more secure image captchas. We believe that our design principles will be useful for future secure image captcha design.
- *Disclosure of Design Flaws.* We have submitted reports with our findings and recommendations to all the involved captcha providers. Among them, `Tencent` and `Netease` have responded to our reports and acknowledged our findings. We hope that the disclosure will result in more robust and secure captcha services.

2 BACKGROUND AND RELATED WORK

Threat Model. In practice, adversaries may follow three approaches for solving image captchas: using automatic captcha breaking attacks, using image recognition services, and hiring human labors.

[2] http://www.163.com

Table 1: Summary of Evaluation.

	Selection-based captchas	Slide-based captchas	Click-based captchas
Examples	 请点击下图中所有的 文具盒 Please select all the images of pencil-boxes	 网易易盾 Please slide the puzzle window to the right region 向右滑动滑块填充拼图	
Providers	facebook.com, 12306.com, google.com	geetest.com, tencent.com, 163.com	geetest.com, tencent.com, 163.com
Attacks	Sivakorn *et al.* [44], Ya *el al.* [49], **This paper**	**This paper**	**This paper**
APIs	GoogleAPI, TencentAPI, AliAPI, MirosoftAPI	–	BaiduOCR, GoogleOCR,TencentOCR, AliOCR, Face++OCR
Captcha-solving Services	ruokuai, yundama, hyocr, 2captcha, AntiCaptcha, Decaptcha, imagetyperz	ruokuai, hyocr, dama2	ruokuai, yundama, dama2

In this paper, we study all the three approaches. For automatic approaches, we design three attacks and evaluate them against 10 popular real-word captcha schemes. For recognition services, we leverage online image classification and object detection services to solve image captchas. For manual attacks, we hire human labors from a broad range of underground captcha-solving services. Regarding our research, we review the related work and background information from three areas below.

2.1 Image Captchas

For completeness, we first briefly summarize representative works on text captchas. Both specific attacks [24, 41, 50] and generic solvers [21, 23, 27] have been proposed against text captchas. Also, many works have extensively studied the security, the distortion or defensive techniques (e.g., rotation, distortion), and the design guidelines for text captchas [19, 42].

Many existing works focus on the design of image captchas [25, 26, 34, 40, 46, 47]. For example, Ahn *et al.* proposed the first use of distorted animal images for captcha design [47], Elson *et al.* presented a selection-based image captcha named Asirra [26], and Misra and Gaj presented the first face recognition based captcha scheme [40]. Most recently, Uzum *et al.* proposed a Real Time Captcha (rtCaptcha) system based on facial authentication [46]. Alternate to image captchas, there also have many audio captchas [20, 22] and sensor captchas [32, 45]. For audio captcha, it is often used together with text- and image- based captchas as a complementary means, mainly because of the usability issue. In addition to the research community, many commercial companies release image captchas. The currently popular real-world image captchas can be roughly classified into three categories: selection-based captchas [25], slide-based captchas, and click-based captchas [33], as shown in Table 1.

In this paper, *instead of studying the not-deployed-yet captchas, we focus on studying the security of real-world image captchas.* We believe such study would be more meaningful for understanding the security of the existing captcha ecosystem.

2.2 Attacks

There exist a few attacks against image captchas [30, 38, 39, 44, 49]. Golle proposed a simple classifier to break the Asirra system [30]. Lorenzi *et al.* proposed a web service based attack against image captchas [38]. Lorenzi *et al.* also proposed a recognition based attack against image captchas [39], where they examined three synthetic captchas. Most recently, Sivakorn *et al.* designed a novel attack that leveraged deep learning techniques to break ReCatpcha [44], and Ya *et al.* developed an association graph based attack to break captchas from 12306.cn [49].

Our work differentiates from the attacks mentioned above in the following aspects. First, we focus on real-world captchas from top sites, e.g., google.com, tencent.com, and 12306.cn, instead of not-deployed-yet captchas. Second, different from [44][49], we develop three effective and generic attacks while they proposed specially-designed attacks against a few cases of selection-based captchas. Furthermore, our attacks are more effective and efficient, e.g., we achieve a high success rate of 90% on China Railway, which is employed by the largest ticket system 12306.cn in China (in this paper, we use China Railway and 12306.cn exchangeably). Third, we comprehensively study the security of click- and slide-based captchas, which *is the first time to our knowledge.* Fourth, we evaluate image recognition services in breaking captchas and the manual attacks provided by underground captcha-solving services.

2.3 Computer Vision and Image Recognition

Recently, the research of computer vision has been revolutionized by deep convolutional neural networks (CNNs) [35, 36], and many basic vision tasks, e.g., image classification [28, 35] and object detection [29, 37, 43], have achieved great success. Both the advanced image classification and object detection techniques are employed by our attacks.

Benefiting from the advanced vision techniques, many commercial companies, also deploy online services for various tasks, including image classification services, character recognition services, object detection services, etc. For example, Google, Microsoft, Baidu, Tencent, Alibaba, and Face++ all provide cloud vision APIs for powerful image analysis. These powerful APIs to some extent can be utilized to perform attacks against image captchas. In our work, to evaluate the performance of these services and make a comprehensive comparison for our attacks, we test 4 image classification services: GoogleAPI [5], TencentAPI [17], MicrosoftAPI [7], and AliAPI [1] and 5 character recognition services: BaiduOCR [3], TencentOCR [18], GoogleOCR [6], AliOCR [2] and Face++OCR [4]. We

Table 2: Summary of real-world image captchas.

Type	Scheme	Provider	Scale
selection-based	ReCaptcha 2015	ReCaptcha	O(billion) users
	ReCaptcha 2018	ReCaptcha	
	China Railway	12306.cn	O(billion) users
	Facebook	Facebook	-
slide-based	GEE SlidePuzzle	GEETest	O(200k) sites
	Tencent SlidePuzzle	Tencent	O(billion) users
	Netease SlidePuzzle	Netease	O(billion) users
click-based	GEE TouClick	GEETest	O(200k) sites
	Tencent TouClick	Tencent	O(billion) users
	Netease TouClick	Neteast	O(billion) users

choose them since since they are popular in the research community and claimed to have high recognition accuracy.

3 POPULAR REAL-WORLD IMAGE CAPTCHAS

To collect representative image captchas, we consult the Alexa list of the most used websites[3] and identify 4 top sites which provide image captcha services to other sites, including ReCatpcha, GEETest, Tencent, and Netease. We collect a total of 8 schemes from these sites. Additionally, we obtain other 2 schemes of selection-based captchas from sites which design captchas by themselves: 12306.cn and facebook.com. Table 2 summarizes the 10 schemes we collect to establish our study. The 10 collected schemes are all from the 3 popular image captcha categories: selection-, slide-, and click-based captchas. Below, we show the design, the workflow, and the example of the 10 captcha schemes.

Selection-based Image Captchas. For selection-based captchas, we collect four popular schemes: ReCaptcha 2015, ReCaptcha 2018, China Railway, and Facebook.

The ReCaptcha offered by Google aims to verify users if possible without requiring them actually to solve a tedious challenge. ReCaptcha first requires a user to click a *checkbox* and calculates a confidence score for this user according to many risk factors returned by the checkbox, e.g., browser characters and cookies of google.com. Then, ReCaptcha returns a selection-based captcha for the user with a low score. Whereas, the user with a high score can directly pass the challenge without any further authentication. In this paper, we mainly focus on the selection-based captchas returned by ReCaptcha, which has two versions, namely ReCaptcha 2015 and ReCaptcha 2018.

Figure 3(a) and Figure 3(b) (Appendix A) show examples of ReCaptcha 2015 and ReCaptcha 2018, respectively. Figure 3(c) (Appendix A) shows an example of Facebook, and Figure 3(d) (Appendix A) shows an example of China Railway. All these captchas contain one hint and different sizes of candidate images. To pass those captchas, a user is requested select all images relevant to the hint.

Slide-based Image Captchas. For slide-based image captchas, we collect three popular real-world schemes, namely GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle.

Figure 4 (Appendix A) shows examples of GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle. All of these challenges contain one puzzle and one background image. To pass those captcha schemes, a user is requested to slide the puzzle to the right part of the background image. Then, captcha providers check whether the puzzle piece is accurately placed or not, and make a risk analysis on the slide trajectory. A user is considered to pass the challenge iff the puzzle piece is rightly placed, and the slide trajectory is no suspicious.

Click-based Image Captchas. For click-based image captchas, we collect three popular real word schemes, namely GEE TouClick, Tencent TouClick, and Netease TouClick.

Figure 5(a) (Appendix A) shows an example of GEE TouClick. This challenge contains one hint of distorted characters and one background image with distorted characters. To solve this challenge, a user is asked to sequentially click the characters drawn in the background image according to the hint and in the right order. Note that the number of distorted characters in hint is the same as that of the distorted characters in the background image.

Figure 5(b) (Appendix A) shows an example of Tencent TouClick. The structure and workflow of Tencent TouClick are similar to those of GEE TouClick, except that Tencent TouClick has more distorted characters in the background image than in the hint.

Figure 5(c) (Appendix A) shows an example of Netease Touclick. This challenge consists of one hint of machine-encoded characters and one background image with distorted characters. To pass this challenge, a user is asked to click the distorted characters sequentially in the right order.

4 SELECTION-BASED CAPTCHAS

In this section, we design an attack named *SelAttack* against selection-based captchas along with evaluation and discussion.

4.1 SelAttack

Selection-based captchas require users to select right images with specific semantic meanings. Hence, it is intuitive that an image classification model can be utilized to understand the semantic meanings of candidate images and find out the right ones. Below, we first give several notations and then show the detailed steps of SelAttack.

4.1.1 Notations. A selection-based captcha contains two parts: a hint of short phrases (e.g., cars and street signs) and several candidate images. There usually exist two types of hints: *text* hint, which is presented in the format of machine-encoded text, and *image* hint, which is presented in an image of distorted characters.

4.1.2 Design of SelAttack. Based on the workflow of selection-based captchas, we design our attack as follows. (1) To bootstrap our attack, we pre-train an image classification model. We also train a character recognition model if the target scheme has an image hint. (2) Upon receiving a challenge, we first extract the candidate images and the hint from it. We then perform image recognition on the hint if it is an image hint. Note that this process is designed to transform the distorted characters of the image hint into machine-encoded text. (3) Next, we utilize the classification model to recognize candidate images and predict their semantic labels. (4) Finally, we select those images relevant to the hint as for the solution of the given captcha.

[3] http://www.alexa.com/topsites

4.2 Implementation and Evaluation

We now evaluate SelAttack on ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway.

Setup. First, we make a preliminary empirical analysis on the four tested schemes, especially for the capacity of hints (i.e., number of distinct hints). Based on the preliminary analysis, we collect 5 datasets with sufficient labeled images for bootstrapping our attack. To be specific, these datasets are used for training 5 image classification models, namely CNN$_1$, CNN$_2$, CNN$_3$, CNN$_4$, and CNN$_5$. Equipped with these models, we run SelAttack against the 4 tested schemes. As a comparison, we also run two state-of-the-art attacks, 4 recognition services and 7 underground captcha-solving services to break the same captchas as evaluated by SelAttack.

4.2.1 Preliminary Analysis. We focus on two primary questions when analyzing the captcha schemes: what are the contents of each scheme's hints, and what is the capacity of those hints (capacity stands for the number of different hints). To this end, we employ a methodology that combines the continuous observation of real online captchas and the statistical analysis on historical datasets of pre-download captchas.

Table 8 (Appendix B) lists the detailed analysis results of the 4 schemes. **ReCaptcha 2015** has the same 22 frequent hints and categories of candidate images. This result is obtained from the statistical analysis on ∼ 700 pre-download captchas since ReCaptcha 2015 is temporarily unavailable and only the historical dataset is avaliable. **Facebook** has the same 12 distinct hints and categories of candidate images. Similar to ReCaptcha 2015, we get this result from a statistical analysis on ∼ 200 pre-download captchas. **China Railway** has the same 80 distinct hints and categories of images. We obtain this through a ∼ 3-month observation, from 2017-08-15 to 2017-10-20, on real captchas from `12306.cn`. **ReCaptcha 2018** has 3 frequent hints: bridge, car, and street signs, and 10 frequent image categories. We obtain this result through a continuous one-month observation, from 2018-02-10 to 2018-03-13, on real captchas. Note that for ReCaptcha 2018, the number of frequent hints is unequal to the number of image categories.

We conjecture the reasons that all the tested schemes only have a limited size of hints as follows. (1) Implementing a selection-based captcha with a small size of hints is simple and convenient. (2) Collecting and labeling images from a wide range of categories is time-consuming and expensive although it is theoretically more secure.

4.2.2 Data Collection. For collecting sufficient data, we employ a methodology that combines automatic crawling, synthetic generation, and benchmark datasets collection. In total, we collect five labeled datasets: D_1, D_2, D_3, D_4, and D_5, which are illustrated in Table 3. D_1, D_2, D_3, and D_4 are all collected from the image searching results of `google.com` and `baidu.com`, and ImageNet, and D_5 is extracted from the pre-download China Railway challenges.

4.2.3 Attack Models. We train CNN$_1$ on D_1 for breaking ReCaptcha 2015, train CNN$_2$ on D_2 for breaking ReCaptcha 2018, train CNN$_3$ on D_3 for breaking Facebook, and train CNN$_4$ and CNN$_5$ on D_4 and D_5, respectively, for breaking China Railway. Especially, CNN$_1$, CNN$_2$, CNN$_3$, and CNN$_4$ are used to label candidate images for ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway, respectively, and CNN$_5$ is used for recognizing distorted hints for China Railway. These models are trained through the standard five-fold cross-validation with no overlap between the training and validation datasets. Moreover, CNN$_1$, CNN$_2$, CNN$_3$, CNN$_4$, and CNN$_5$ are all trained with a batch size of 16 and a learning rate of 1e−4, and on an Ubuntu server equipped with an Intel i5-7500 CPU, a GTX 1060 GPU, and 16 GB memory. Table 4 summarizes the model details, training processes, and training results.

Table 3: Labeled datasets. IC = Image Category, IPC = Images Per Category.

Dataset	#IC	#IPC	#Images	Usage
D_1	22	1,500	33,000	ReCaptcha 2015
D_2	10	1,500	15,000	ReCaptcha 2018
D_3	12	1,500	18,000	Facebook
D_4	80	1,500	120,000	China Railway
D_5	80	∼ 750	60,000	China Railway
D_6	3,755	∼ 1,400	5,257,000	GEE, Tencent, and Netease TouClick
D_7	–	–	2,000	Gee TouClick
D_8	–	–	2,000	Tencent TouClick

Table 4: Summary of pre-trained deep models.

Model Name	Model Type	Accuracy	Training Time
CNN$_1$	CNN	95.97%	18 hours
CNN$_2$	CNN	91.77%	8 hours
CNN$_3$	CNN	97.27%	9 hours
CNN$_4$	CNN	93.27%	93 hours
CNN$_5$	CNN	96.61%	25 hours
CNN$_6$	CNN	99.86%	17 hours
Fast-RCNN$_1$	R-CNN	92.01%	7 hours
Fast-RCNN$_2$	R-CNN	97.12%	12 hours

4.2.4 Attacks. Now, equipped with the five pre-trained models, we run SelAttack against captchas from ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway. For the two temporally taken down services, ReCaptcha 2015 and Facebook, we perform our attack against 684 and 200 pre-download challenges, respectively. For the two available captcha services, ReCaptcha 2018 and China Railway, to minimize our impact, we perform our proof-of-concept attack on 200 real online captchas from each of them. To validate our attack results on taken down services, we manually inspect the captcha challenges and figure out the right solutions.

Then, we test 2 state-of-the-art attacks, 4 recognition services, and 7 underground captcha-solving services, respectively. For prior arts, we test 2 state-of-the-art attacks: Ya *el al.* [49] and Sivakorn *el al.* [44], which claim to be cost-effective and widely applicable. We fine-tune and apply them on ReCaptcha 2015, Facebook, and China Railway. Note that we do not evaluate them on ReCaptcha 2018 simply because such evaluation is time-consuming and label extensive. For recognition services, we leverage `AliAPI`, `GoogleAPI`, `MicrosoftAPI`, and `TencentAPI` to attack the considered captchas. For each recognition service, we request API calls for 400 challenges with 100 per captcha scheme. For human labors, we evaluate the effectiveness and efficiency of 7 popular human captcha-solving services, including `ruokuai`, `yundama`, `2captcha`, `hyocr`,

Table 5: Attack Results on Selection-based Image Captchas. "-" stands for Not Given.

Methods	ReCaptcha 2015		ReCaptcha 2018		Facebook		China Railway	
	success rate	speed (s)	success rate	speed (s)	success rate	speed (s)	success rate	speed (s)
Our Method								
Our method	88%	1.26	79%	4.92	86%	1.41	90%	4.14
Prior Arts								
Ya *el al.* [49]	14%	0.59	-	-	9%	0.47	52%	6.62
Sivakorn *el al.* [44]	71%	20.80	-	-	83%	25.30	37%	20.60
Image Recognition Services								
`TencentAPI`	19%	13.64	6%	20.19	25%	15.32	3%	14.97
`GoogleAPI`	62%	16.13	49%	23.31	73%	19.53	7%	17.82
`AliAPI`	37%	14.27	11%	18.40	35%	13.04	16%	12.65
`MirosoftAPI`	21%	19.95	8%	25.42	44%	21.09	2%	17.01
Captcha-solving Services								
`ruokuai`	81%	4.54	91%	6.97	88%	4.21	86%	5.57
`yundama`	89%	4.36	-	-	77%	5.18	88%	5.29
`hyocr`	-	-	85%	7.05	-	-	-	-
`2captcha`	86%	8.35	88%	4.27	90%	7.98	79%	11.37
`AntiCaptcha`	84%	6.43	92%	5.69	93%	8.71	65%	9.94
`DeCaptcha`	41%	23.16	62%	31.12	46%	25.24	-	-
`imagetyperz`	-	-	95%	41.68	-	-	-	-

`AntiCaptcha`, `DeCaptcha`, and `imagetype`. We select these 7 services since they support selection-based captchas. Due to the budget limit, for each captcha-solving service, we submit 400 challenges with 100 per captcha scheme.

Table 5 shows the success rate and speed of SelAttack on the 4 evaluated schemes. The success rate of SelAttack ranges from 79% to 90%, which is relatively high. Taking China Railway as an example, it is reported that only 2%, 27% and 65% of human users successfully pass the captcha on their first, second, and third attempts, respectively [13]. SelAttack achieves a success rate of 90% on China Railway. The lowest success rate of 79% is achieved on ReCatpcha 2018, which is still very high as compared with the successful breaking rate 1% of a successful attack suggested by [23]. The difficulty of ReCaptcha 2018 might result from the following reasons. (1) ReCaptcha 2018 has a larger size of candidate images, which might introduce more classification errors. (2) ReCaptcha 2018 has several confusing image categories, e.g., bridge and road, which are hard to be recognized even for human beings.

On average, our attack takes 1 to 5 seconds to break the tested schemes, which is relatively fast. We note that the solving time of ReCaptcha 2018 and China Railway includes an overhead of network delay, estimated at 3 seconds per captcha. The fastest speed excluded the network overhead, ∼ 1 second is achieved on China Railway, and the slowest speed excluded network overhead, ∼ 2 seconds, is achieved on ReCaptcha 2018. Interestingly, we find that the solving time excluded network overhead scales linearly as the candidate image size increases. This characteristic suggests that SelAttack is scalable in practice, and a parallel implementation of SelAttack can be applied to solve large-scale image captchas.

Table 5 also shows the running results of the 2 prior arts, 4 image recognition services, and 7 human captcha-solving services. The success rates of prior arts, recognition services, and captcha-solving services range from 9% to 83%, 2% to 73%, and 41% to 93%, respectively. The speeds including network overhead of them range from 0.47 to 20.8 seconds, 12.65 to 21.09 seconds, and 4.36 to 25.24 seconds, respectively. From the comparison with other methods (i.e., prior arts, recognition services, and captcha-solving services), we can see that (1) SelAttack is more elegant in both success rate and speed compared with existing attacks; (2) SelAttack is more powerful than online recognition services. Still, some of the online services (e.g., `GoogleAPI`) can provide a compromise option when the time and computing environment is limited; (3) SelAttack is comparable to captcha-solving services in both attack duration and effectiveness, and the gap between SelAttack and human labors is narrow and acceptable. We also surprisingly find that, on China Railway, SelAttack even has a higher success rate than that of the most skilled human labors. As for other schemes, the gap between SelAttack and skilled human labors is narrow.

In summary, the high success rate and low solving time imply that SelAttack poses a realistic threat to selection-based captchas.

4.3 Design Flaws

Based on the evaluation results, we summarize the following design flaws of real-world selection-based captchas. First, all the tested schemes have a limited size of hints. Due to this limited size, an adversary can easily enumerate all the hints and train an accurate image classification model. Second, most of the tested schemes use text hint, which can be extracted out even without any effort. Third, the candidate images have little resilience from the security perspective (usually has no noise). Therefore, a well-trained model can accurately understand their semantic meanings.

5 SLIDE-BASED IMAGE CAPTCHAS

In this section, we detail the design of *SliAttack* against slide-based captchas along with evaluation and discussion.

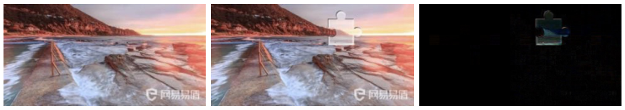
5.1 SliAttack

Slide-based captcha asks a user to a slide puzzle piece to the right part of an image. For convenience, we name this right part *puzzle region.* The key to automatically breaking this captcha is to find the puzzle region accurately, and mimic human behaviors when sliding the puzzle piece. Below, we first describe how to find the puzzle region and mimic human behaviors.

5.1.1 Puzzle Region Detection. Through analyzing 2,000 slide-based captchas, we observe that a single source image is repeatedly used to generate a great many captcha challenges in real-world captcha systems (e.g., Netease SlidePuzzle shown in Figure 6 of Appendix B). Based on this observation, it is intuitive that (1) a source image can be recovered through analyzing a set of captchas generated from this source image, and (2) the comparison between a captcha and its source image can be used to localize the puzzle region accurately. Hence, we detect the puzzle region through two steps: source image recovery and comparison-based region detection.

Source Image Recovery. Let s denote a source image, and $I^s = \left\{I_i^s | i = 1, 2, \ldots\right\}$ be the set of background images generated from s. We further define $I_i^s = \left\{I_i^s(j) | j = 1, 2, \ldots\right\}$, where $I_i^s(j)$ is the jth pixel of I_i^s. Now, we briefly introduce our source image recovery algorithm, as illustrated in Algorithm 1. (1) Single pixel reconstruction (lines 4–5): we construct the jth pixel of s through selecting the most frequent value from the pixel set, denoted by p, consisting of all the jth pixels of I^s. (2) Image reconstruction (lines 2–7): we recover s by the continuous construction process of all pixels.

Comparison-based Region Detection. The puzzle region can be detected through the comparison between the background image and its source image, e.g., a simple XOR operation can be used to detect the region. Figure 2 illustrates the process of puzzle region detection through the XOR operation between the background image and its source image.



(a) source image (b) backgournd image (c) detected region

Figure 2: The process of puzzle region detection.

Algorithm 1: Source Image Recovery

```
Input: $I^s$
Output: $s$
1 Initialize $s \leftarrow \emptyset$
2 for $j \in \{1, 2, \ldots\}$ do
3     p ← $\emptyset$
4     for $i \in \{1, 2, \ldots\}$ do
5         p ← p$\cup I_i^s(j)$
6     candidate ← the most frequent value in $p$
7     $s \leftarrow s\cup$candidate
```

5.1.2 Human Behavior Simulation. Some slide-based schemes detect malicious behaviors (e.g., a fast and direct move to the puzzle region), which are considered to be machine-generated behaviors. Therefore, we mimic human behaviors leveraging 4 simulation functions: Sigmoid [14], Softmax [15], ReLu [8], and Tanh [16], to bypass such detection.

Let b denote the distance between the puzzle piece and region. Let $D = \{D_i | i = 1, 2, \ldots, k\}$, where $\sum_{D_i \in D} D_i = b$, denote the length set of moving steps, and $T = \{T_i | i = 1, 2, \ldots, k\}$ denote the time set of moving steps. To bypass the malice detection, we generate D and T as follows. Consider the Sigmoid function, $f(x) = \frac{1}{1+e^{-x}}$, as an example. (1) We assign the length of each step as $D_i = b \times \left(\frac{1}{1+e^{-i/2+4}} - \frac{1}{1+e^{-(i-1)/2+4}}\right)$, where i is an integer and $1 \leqslant i \leqslant k$. Note that, to meet the constraints that $\sum_{D_i \in D} D_i = b$, we set $D_1 = b \times \frac{1}{1+e^{-1/2+4}}$ and $D_k = b \times \left(1 - \frac{1}{1+e^{-k/2+4}}\right)$. (2) Then, we randomly shuffle D to get the final sequence of moving steps. For T, we randomly generate the moving time for each moving step.

As for the other 3 functions: Softmax, ReLu, and Tanh, their working mechanisms are similar to that of Sigmoid.

5.1.3 Design of SliAttack. We design SliAttack as follows. We collect a set of captcha challenges from the target scheme, and use Algorithm 1 to recover the set of source images. This process is mainly used to bootstrap our attack. After that, we can automatically solve real-world captchas from the target scheme. When receiving a captcha challenge, we first extract the background image and find its corresponding source image. Then, we localize the puzzle region by comparing the background image and the source image. Next, we mimic human behaviors and slide the puzzle to the detected puzzle region.

5.2 Implementation and Evaluation

To evaluate SliAttack against slide-based captchas, we conduct a series of experiments on 3 different schemes: GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle.

Setup. We perform SliAttack against slide-based image captchas as follows. First, we recover source images for the tested schemes if possible to bootstrap our attack. Specially, we have recovered 10 source images from 2,000 pre-download challenges for Tencent SlidePuzzle, 11 source images from 2,000 pre-download challenges for Netease SlidePuzzle, and 8 source images from 2,000 pre-download challenges for GEE SlidePuzzle. Then, we run the 6 settings of SliAttack against all the schemes: the attack based on direct moving, random moving, Sigmoid moving, Softmax moving, Tanh moving, and ReLu moving. Note that we evaluate direct moving and random moving for confirming whether the target scheme employs any malice detection strategy. To minimize our impact on real systems, for each scheme, we attack 200 real online challenges.

We compare SliAttack with human labors from 3 underground captcha-solving services: `ruokuai`, `hyocr`, and `dama2`. We select these services since they are popular and support slide-based captchas. Due to the budget limit, for each service, we submit 300 challenges with 100 per scheme.

5.2.1 Results. Table 6 summarizes the success rate and speed of SliAttack on each scheme. Below, we first discuss the effectiveness of our human behavior simulation, and then the overall success rate and speed of SliAttack. Finally, we make a comparison between SliAttack and the captcha-solving services.

Table 6: Attack Results on Slide-based Image Captchas. SR = success rate, SD = Speed.

Methods	GEE		Tencent		Netease	
	SR	SD (s)	SR	SD (s)	SR	SD (s)
Our Method						
Sigmod	96%	5.30	100%	4.01	98%	1.98
Softmax	59%	5.27	95%	4.18	72%	2.15
Tanh	0%	5.16	100%	4.06	98%	2.24
ReLu	54%	5.68	99%	4.27	54%	5.68
Random	16%	5.33	97%	4.33	81%	2.35
Direct	0%	2.37	100%	0.88	0%	1.71
Captcha-solving Services						
ruokuai	88%	8.82	96%	7.94	91%	6.06
hyocr	93%	9.69	92%	5.73	87%	7.71
dama2	91%	11.03	97%	6.13	95%	8.17

We can observe from Table 6 that our attack based on the Sigmoid function has the highest success rate on all schemes. This result suggests that SliAttack's behavior simulator is effective in practice.

GEE SlidePuzzle is the most robust one among the three tested schemes. On GEE SlidePuzzle, our attack achieves the best success rate of 96% by using the Sigmoid function, while the success rate decreases significantly if we use other functions. Surprisingly, we find that Tencent SlidePuzzle probably has no mechanism for malice detection. Our attack has a 100% success rate even when we directly move the slide puzzle to the puzzle region. We conjecture that it is a design flaw. This conjecture is confirmed by Tencent after we report our findings to it.

SliAttack's success rates are all above 96%, and the best success rate reaches 100%. Such a big success rate not only indicates the effectiveness of our attack, but also reveals the vulnerabilities of real-world slide-based captcha schemes.

On average, it takes 1 to 6 seconds for SliAttack to break all schemes. The fastest speed is on Tencent SlidePuzzle, about 1 second. The slowest speed is on GEE SlidePuzzle, nearly 5 seconds – still very fast as compared to the speed requirement of human users ($\sim$ 30 seconds). The following reason explains why it takes much less time to break Tencent SlidePuzzle than others. Tencent SlidePuzzle does not inspect the slide trajectory, and therefore, our attack can directly slide the puzzle piece to the puzzle region, significantly reducing the solving time. In contrast, for GEE SlidePuzzle and Netease SlidePuzzle, our attack randomly stops and waits for $1 - 2$ seconds to evade the malice detection.

Table 6 also shows the success rate and speed of the 3 captcha-solving services. From the comparison between SliAttack and the 3 captcha-solving services, we have the following interesting findings. (1) Skilled human labors fail to achieve a better success rate than that of SliAttack on the tested schemes. (2) Skilled human labors averagely require $7-10$ seconds to solve the captchas, a much slower speed as compared to SliAttack. We conjecture the reasons why human labors are less effective as follows: the measurement errors produced by human labors can significantly affect the positioning accuracy of the puzzle region, leading to a wrong solution. Again, from the comparison with human labors, we conclude that SliAttack is highly effective and the current common practice of slide-based captchas, however, is invalid.

5.3 Design Flaws

Based on our evaluation results, we summarize the following design flaws of slide-based captchas. First, most schemes repeatedly use the same source images to generate challenges, which makes it easy for an adversary to localize puzzle regions. Second, the malice detection methods used by the tested real-world schemes are invalid and cannot defend against adversaries, and some scheme even does not employ any detection algorithm.

6 CLICK-BASED IMAGE CAPTCHAS

In this section, we introduce the design of *CliAttack* against click-based captchas along with evaluation and discussion.

6.1 CliAttack

6.1.1 *Notations.*

A click-based captcha consists of two parts: one image hint and one background image. Similar to selection-based captchas, this hint has two formats: text hint and image hint. Those characters contained in the hint are also drawn on the background image.

6.1.2 *Design of CliAttack.*

We show the workflow of CliAttack as follows. (1) To bootstrap our attack, we pre-train a character recognition model for recognizing distorted characters from both the hint and the background image. Moreover, we also pre-train a character detection model on a dataset of captcha challenges with annotated semantic regions, which is collected from the target captcha scheme. (2) Once receiving a challenge, we extract the hint and use the character recognition model to recognize the hint's distorted characters if necessary. (3) We then localize potential semantic regions leveraging the pre-trained character detection model. The semantic meanings of those regions are recognized by the character recognition model as well. (4) After comparing the potential semantic regions with the hint, we sequentially click the right semantic regions of distorted characters drawn on background images.

6.2 Implementation and Evaluation

To evaluate CliAttack, we run a set of experiments against GEE TouClick, Tencent TouClick, and Netease TouClick.

To bootstrap the attack, we collect and manually label 3 datasets, and train 3 models on these labeled datasets for detecting and recognizing distorted characters.

6.2.1 *Data Collection.*

For breaking GEE TouClick, Tencent TouClick, and Netease TouClick, we collect a total of 3 labeled datasets: D_6, D_7, and D_8 as illustrated in Table 3. D_6 contains $5,257,000$ images of $3,750$ commonly used Chinese characters, which is generated by 16 different font generation algorithms. D_6 is used for training a character recognization model. D_7 and D_8 consist of $2,000$ pre-download GEE TouClick challenges and $2,000$ pre-download Tencent TouClick challenges, respectively, with manually annotated regions of distorted characters.

Both Netease TouClick and Netease SlidePuzzle use the same set of images to generate their challenges, which is discovered during our attack against slide-based captchas. Hence, we can directly detect distorted characters through a comparison between the captcha challenge and its source image.

Table 7: Attack Results on Click-based Image Captchas. SR = success rate, Speed = SD.

Methods	GEE		Tencent		Netease	
	SR	SD (s)	SR	SD (s)	SR	SD (s)
Our Method						
Our method	46%	4.63	74%	4.78	69%	4.13
Image Recognition Services						
BaiduOCR	4%	6.55	36%	6.14	12%	5.70
GoogleOCR	5%	13.37	27%	11.22	3%	12.15
TencentOCR	2%	6.09	51%	6.53	7%	6.41
AliOCR	3%	7.54	13%	6.60	3%	7.17
Face++OCR	8%	7.96	30%	8.79	5%	8.38
Captcha-solving Services						
ruokuai	84%	9.47	93%	7.09	89%	8.04
yundama	89%	8.86	86%	7.37	87%	7.28
dama2	85%	9.98	90%	6.84	94%	9.11

6.2.2 Attack Models. We train three models for breaking the tested schemes: CNN_6, Fast-RCNN_1, and Fast-RCNN_2. Specifically, we train CNN_6 on D_6, Fast-RCNN_1 on D_7, and Fast-RCNN_2 on D_8. CNN_6 is further used for character recognition on all schemes, Fast-RCNN_1 is used for character detection on GEE TouClick, and Fast-RCNN_2 is used for character detection on Tencent TouClick.

CNN_6 is trained with a batch size of 16 and a learning rate of 1e−4, while Fast-RCNN_1 and Fast-RCNN_2 are all trained with a batch size of 300 and a learning rate of 1e−5. For the training environment, CNN_6 is trained on an Ubuntu server with two Intel-Xeon E5-2640V4 CPUs, a GTX 1080Ti GPU, and 128 GB memory. Fast-RCNN_1 and Fast-RCNN_2 are trained on an Ubuntu server with an Intel i5-7500 CPU, a GTX 1060 GPU, and 16 GB memory. All the models are trained through the standard five-fold cross-validation. Table 4 summarizes the model details, training processes, and training results.

6.2.3 Attack Results. Now, we evaluate our attack against real online captchas from GEE TouClick, Tencent TouClick, and Netease TouClick. To minimize our impact on real systems, we run our attack against 200 real-world challenges from each scheme.

For a comparison with online recognition services, we run 5 OCR (optical character recognition) services, including BaiduOCR, TencentOCR, GoogleOCR, AliOCR, and Face++OCR. We select these 5 OCR services since they are reported to have an accurate and rapid recognition for distorted characters drawn on images. For each online service, we test 300 challenges with 100 per scheme.

For a comparison with human labors, we evaluate the attack results of 3 captcha-solving services: ruokuai, yundama, and dama2. We choose these 3 captcha-solving services due to their popularity in the underground captcha-solving market and their support for click-based captchas. Due to the budget limit, for each captcha-solving service, we submit 300 captchas with 100 per captcha scheme.

Table 7 shows the results of our attack. Our attack's success rates are all above 45%, and the best success rate, 74%, is achieved on Tencent TouClick. Those results suggest that CliAttack is effective in practice.

From Table 7, we can also observe that the most challenging scheme is GEE TouClick, on which our attack's success rate is 46%. We conjecture that the following two reasons make GEE TouClick very challenging. (1) The similarity between the color of distorted characters and background images adds the difficulty in localizing distorted characters. (2) The distorted characters might be generated by a large number of different font generation algorithms. Our character recognition model, CNN_6, however, is trained on distorted characters from a limited size of fonts (16 fonts). It is reasonable that our attack loses some accuracy on GEE TouClick.

Our attack's speed ranges from 3.49 to 4.63 seconds, which is relatively fast since an excessive usability requirement is to demand a user to solve a captcha within 30 seconds. We note that the solving time of all schemes icludes an overhead of network delay, estimated at 3 seconds per captcha challenge. This fast attack speed suggests that our attack poses a realistic threat to all these schemes.

Table 7 also summarizes the evaluation results of the 5 OCR services and 3 captcha-solving services as well. The success rates of the recognition and captcha-solving services range from 7% to 51% and 80% to 91%, respectively. The speeds of them range from 5.70 to 12.15 seconds and from 6.84 to 9.98 seconds, respectively. The comparison among CliAttack, OCR, and captcha-solving services suggests that (1) CliAttack with a pre-trained character recognition model is more powerful in attacking click-based captchas than online OCR services; (2) CliAttack achieves acceptable success rates compared with human labors while much faster.

6.3 Design Flaws

Based on our results, we summarize the design flaws of click-based captchas as follows. The most serious design flaw is that some captcha providers, e.g., Netease, use the same image set to generate challenges for different schemes. Second, some of the tested schemes do not perform any anti-recognition operations (e.g., rotation) on the characters drawn on background images.

7 COUNTERMEASURES

We distill our automatic captcha-breaking attacks, the evaluation of online vision services, and the analysis of underground captcha-solving services into a set of best practices and design principles for facilitating the design of secure captchas.

The Scalability of Captcha Corpus. The scalability measures the number of challenges a captcha scheme can generate without sacrificing its robustness and security. Among the 10 tested schemes, none of them is scalable since they either have a limited size of hint categories which can be easily enumerated, or their source images and candidate images are repeatedly used. Focusing on the scalability, we discuss the following 3 countermeasures and design principles for defending against our attacks.

(1) *The Size of Hint Categories.* A large size of hint categories means that it takes more time to enumerate the hint corpus, to collect sufficient datasets, and to train an accurate model. Hence, this countermeasure can slow down SelAttack. (2) *The Size of Source Images.* Don't repeatedly use one single source image to generate challenges, while using a large-scale corpus of source images. The best practice is to use a single source image for only once. This best practice can defend against SliAttack as the once only strategy prevents our attack from detecting the puzzle region. However, this strategy may also increase the security cost. (3) *The Size of*

Candidate Images. Use a wide broad of candidate images. Candidate images should belong to categories that are excluded to the hint. This countermeasure might increase the number of labeling errors proceeded by pre-trained classification models and reduce the success rate of our attack.

Risk Analysis. Do perform risk analysis on simple captcha schemes. It is expected to evaluate the possibility whether the captcha solution is given by abusive programs or not. For Tencent SlidePuzzle as an example, our attack will be mitigated if it makes a risk analysis on the slide trajectory.

Anti-Recognition. To be secure, anti-recognition techniques could be implemented in image captchas. We discuss 3 simple anti-recognition techniques. (1) Apply distortion techniques (e.g., varied fonts, varied font size, and rotations) on characters on the background image or in the hint. The distortion technique could directly reduce character recognition accuracy, and therefore it may mitigate the threat posed by SelAttack and CliAttack. (2) Add noises on background images. For the slide-based image captcha as an example, if we randomly add a deceptive empty region in the background image, the success rate of SliAttack will be reduced by half. (3) Generate adversarial images that are imperceptible to humans while can fool deep image classification models (Figure 7 of Appendix B shows such an example). This countermeasure can not only defend against our attacks but also mitigate the threat posed by recognition service based attacks.

8 DISCUSSION

Ethical Issues. Most evaluation results and findings of this paper are made on datasets crawled from the public domain. While performing the attack against image captchas cannot be prohibited, our evaluation is designed to minimize the impacts on those captcha providers' websites. We haven't affected the websites in other ways except for acquiring captcha challenges.

The human labors employed to solve image captchas are from the underground captcha-solving services, which might be illegal. Hence, the captcha-solving service request is designed as little as possible to minimize the potential negative effects on workforce.

Furthermore, we have disclosed reports with our findings and recommendations to all the involved captcha providers, in an effort to make their captchas more robust to automated attacks. Up till now, `Tencent` and `Netease` have responded to our reports (as shown in Figure 8 of Appendix B), and they also acknowledged our findings and recommendations. Specially, `Tencent` temporarily shuts down Tencent TouClick, and `Netease` updates Netease TouClick. We hope that the disclosure of our findings will result in more robust captcha services.

Limitations. We believe our work can be improved in many perspectives. Below, we discuss the limitations of this work along with future work.

First, we focus on the security of 3 categories of popular image captchas, and propose simple yet powerful attacks. Also, we evaluate our attacks against 10 real-world captcha schemes and reveal the design flaws of them. Though our research is useful and effective, it is excepted to consider more captcha categories and schemes.

Second, these 3 attacks still have the possibility to be improved. For SelAttack, we train the image classification model on a small labeled dataset of images. Therefore, the success rate of SelAttack could be improved if we train a more accurate image classification model. For SliAttack, we have implement 4 simulation functions that are effective in bypassing the malice detection of real-world captchas. Nonetheless, more simulation functions or other possible human behavior simulation methods are expected to be invented for acquiring a better performance in bypassing the malice detection. For CliAttack, its success rate is limited to the recognition accuracy of Chinese characters. Therefore, there needs a more delicate recognition model, which is trained on a large-scale dataset of labeled distorted characters from a variety of fonts.

More Future Work Directions. Our study reveals the vulnerability of current popular captcha schemes. To mitigate the captcha threat, more future work can be considered. We give two possible future work directions below.

Malicious API Call Detection. For those vision service providers (e.g., `Google`, `Microsoft`, `Baidu`, etc.), they are expected to make a risk analysis on the incoming API calls. This risk analysis may detect malicious API calls from miscreants for many improper uses, e.g., labeling candidate images of captchas and recognizing distorted characters. Therefore, one of the future work directions is to develop a risk analysis system for online vision services.

Underground Market Mining. While the captcha threat posed by human attacks is hard to defend against, we can turn to monitor and detect the underground captcha-solving services, which can mitigate the threat on root.

9 CONCLUSION

In this paper, we study the security of real-world popular image captchas. To this end, we propose 3 proof-of-concept attacks against selection-, slide-, and click-based captchas. We evaluate our attacks on 10 popular real-world captcha schemes, provided by `google.com`, `tencent`, etc., and successfully break all of them. We also compare our attacks with 2 prior arts, 9 online image recognition services, and 8 human-based captcha-solving services. The evaluation results show that our attacks pose a significant and realistic threat to various real-world image captchas. Then, we distill our attacks, the evaluation of recognition services and the underground captcha-solving services, into a set of best practices and design principles towards designing secure captchas. We believe that our study in this paper will be useful for securing the current captcha ecosystem.

ACKNOWLEDGEMENT

This work is partly supported by NSFC under No. 61772466, the Provincial Key Research and Development Program of Zhejiang, China under No. 2017C01055, the Alibaba-ZJU Joint Research Institute of Frontier Technologies, the CCF-NSFOCUS Research Fund under No. CCF-NSFOCUS2017011, and the CCF-Venustech Research Fund under No. CCF-VenustechRP2017009. This work is also based upon work supported by the National Science Foundation under Grant No. 1566526 and 1718787.

REFERENCES

[1] *AliAPI.* https://data.aliyun.com/ai?spm=a2c0j.9189909.810797.13.64c6547a3VOVGD#/image-tag

[2] *AliOCR.* https://www.aliyun.com/product/cdi/

[3] *BaiduOCR.* https://cloud.baidu.com/product/ocr.html

[4] *Face++OCR.* https://www.faceplusplus.com.cn/general-text-recognition/

[5] *GoogleAPI.* https://cloud.google.com/vision/

[6] *GoogleOCR.* https://cloud.google.com/vision/docs/ocr

[7] *MicrosoftAPI.* https://azure.microsoft.com/zh-cn/services/cognitive-services/computer-vision/

[8] *ReLu.* https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

[9] *Report.* https://cloud.tencent.com/product/yy#featuresV2

[10] *Report.* http://www.geetest.com/case.html

[11] *Report.* https://www.google.com/recaptcha/intro/

[12] *Report.* http://kqga.qfc.cn/news/d-1786.html

[13] *Report.* https://baike.baidu.com/item/12306%E9%AA%8C%E8%AF%81%E7%A0%81/16963369?fr=aladdin

[14] *Sigmoid.* https://en.wikipedia.org/wiki/Sigmoid_function

[15] *Softmax.* https://en.wikipedia.org/wiki/Softmax_function

[16] *Tanh.* https://brenocon.com/blog/2013/10/tanh-is-a-rescaled-logistic-sigmoid-function/

[17] *TencentAPI.* https://youtu.qq.com/#/img-content-identity

[18] *TencentOCR.* https://ai.qq.com/product/ocr.shtml#identify

[19] Ahmad Salah El Ahmad. 2012. *The robustness of text CAPTCHAs.* Ph.D. Dissertation. University of Newcastle Upon Tyne, UK. http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.576635

[20] Jeffrey P. Bigham and Anna Cavender. 2009. Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use.

[21] Elie Bursztein, Jonathan Aigrain, Angelika Moscicki, and John C. Mitchell. 2014. The End is Nigh: Generic Solving of Text-based CAPTCHAs. In *8th USENIX Workshop on Offensive Technologies, WOOT '14, San Diego, CA, USA, August 19.*

[22] Elie Bursztein and Steven Bethard. 2009. Decaptcha: breaking 75% of eBay audio CAPTCHAs. In *Proceedings of the 3rd USENIX conference on Offensive technologies.*

[23] Elie Bursztein, Matthieu Martin, and John C. Mitchell. Text-based CAPTCHA strengths and weaknesses. In *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011.*

[24] Kumar Chellapilla and Patrice Y. Simard. 2004. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). In *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS 2004, December 13-18, 2004, Vancouver, British Columbia, Canada].*

[25] Monica Chew and J Doug Tygar. 2004. Image recognition captchas. In *International Conference on Information Security.* Springer, 268–279.

[26] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007.*

[27] 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 *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016.*

[28] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.*

[29] Ross B. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015.*

[30] Philippe Golle. Machine learning attacks against the Asirra CAPTCHA. In *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008.*

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015.*

[32] Thomas Hupperich, Katharina Krombholz, and Thorsten Holz. 2016. Sensor Captchas: On the Usability of Instrumenting Hardware Sensors to Prove Liveliness. In *Trust and Trustworthy Computing - 9th International Conference, TRUST 2016, Vienna, Austria, August 29-30, 2016, Proceedings.*

[33] Kuo-Feng Hwang, Cian-Cih Huang, and Geeng-Neng You. A Spelling Based CAPTCHA System by Using Click. In *2012 International Symposium on Biometrics and Security Technologies, ISBAST 2012, Taipei, Taiwan, March 26-29, 2012.*

[34] Jonghak Kim, Joonhyuk Yang, and Kwangyun Wohn. 2014. AgeCAPTCHA: an Image-based CAPTCHA that Annotates Images of Human Faces with their Age Groups. *TIIS* (2014).

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.*

[36] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* 1, 4 (1989), 541–551.

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I.*

[38] David Lorenzi, Jaideep Vaidya, Shamik Sural, and Vijayalakshmi Atluri. Web Services Based Attacks against Image CAPTCHAs. In *Information Systems Security - 9th International Conference, ICISS 2013, Kolkata, India, December 16-20, 2013. Proceedings.*

[39] David Lorenzi, Jaideep Vaidya, Emre Uzun, Shamik Sural, and Vijayalakshmi Atluri. Attacking Image Based CAPTCHAs Using Image Recognition Techniques. In *Information Systems Security, 8th International Conference, ICISS 2012, Guwahati, India, December 15-19, 2012. Proceedings.*

[40] Deapesh Misra and Kris Gaj. Face Recognition CAPTCHAs. In *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), 19-25 February 2006, Guadeloupe, French Caribbean.*

[41] Greg Mori and Jitendra Malik. 2003. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA.* 134–144.

[42] Lei Pan and Yan Zhou. 2013. Developing an Empirical Algorithm for Protecting Text-Based CAPTCHAs against Segmentation Attacks. In *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013 / 11th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA-13 / 12th IEEE International Conference on Ubiquitous Computing and Communications, IUCC-2013, Melbourne, Australia, July 16-18, 2013.* 636–643.

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 779–788.

[44] Suphannee Sivakorn, Iasonas Polakis, and Angelos D Keromytis. 2016. I am robot:(deep) learning to break semantic image captchas. In *Security and Privacy (EuroS&P), IEEE European Symposium on.* 388–403.

[45] B Srinivas, G Kalyan Raju, and Koduganti Venkata Rao. 2011. Advanced CAPTCHA technique using Hand Gesture based on SIFT. *Assistant Professor, Computer Science and Engineering Department, MVGR College of Engineering* (2011).

[46] Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. rtCaptcha: A Real-Time CAPTCHA Based Liveness Detection System. (????).

[47] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. CAPTCHA: Using Hard AI Problems for Security. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings.*

[48] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. 2003. CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques.* 294–311.

[49] Heqing Ya, Haonan Sun, Jeffrey Helt, and Tai Sing Lee. 2017. Learning to Associate Words and Images Using a Large-scale Graph. *arXiv preprint arXiv:1705.07768* (2017).

[50] Jeff Yan and Ahmad Salah El Ahmad. 2008. A low-cost attack on a Microsoft captcha. In *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, October 27-31, 2008.*

Appendix A EXAMPLE OF IMAGE CAPTCHAS

In this section, we illustrate representative real-world examples of selection-based, slide-based and click-based image captchas. Specifically, Figure 3 shows examples of ReCatpcha 2015, ReCatpcha 2018, Facebook, and China Railway, respectively. Figure 4 shows examples of GEE SlidePuzzle, Tencent SlidePuzzle, and Netease SlidePuzzle, respectively. Figure 5 shows examples of GEE TouClick, Tencent TouClick, and Netease TouClick, respectively.

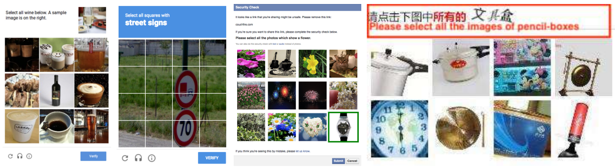
Appendix B OTHERS

Table 8 shows the preliminary analysis results of the four tested selection-based captchas: ReCaptcha 2015, ReCaptcha 2018, Facebook, and China Railway.

Figure 7 presents an example of adversarial images, where the dog is recognized wrongly as flower after inserting elaborately crafted noises.

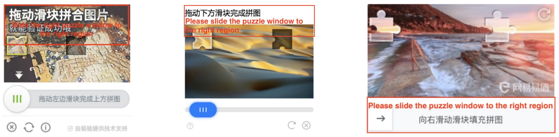
Table 8: Statistics of the 4 schemes.

Scheme	#Categories	Categories
ReCaptcha 2015	22	artichoke, avocado, banana, beer, bread, cabbage, cake, cat, coffee, dog, guinea pig, hamburger, ice cream, pasta, pizza, rice dish, rose, sandwich , soup, steak, sushi, wine
ReCaptcha 2018	10	sea, bridge, grass, house, road, sky, street sign, telephone pole, tree, car
Facebook	12	bicycle, cat, chair, cloud, dog, fireworks, flower, guitar, lion, tiger, waterfall, wristwatch
China Railway	80	Chinese knot, dashboard, bus card, refrigerator, band Aid, embroidery, paper cut, seal, tape measure, double-sided adhesive, whistle, beer, helmet, corkscrew, palm print, typewriter, cuff, mop, wall clock, ventilator, pencil case, calendar, notebook, portfolio, cotton swab, cherry, woolen, sandbags, salad, poster, seaweed, seagull, funnel, candlestick, hot-water bottle, archway, lion, coral, electronic scales, wire , rice cooker, plate, basketball, jujube, red bean, red wine, mung bean, tennis racket, tiger, earplug, aircraft carrier, fly swatter, tea table, tea cup, pill, pineapple, steamer, french fries, ant, bee, candle, lizard, stapler, plum, palette treadmill, street light, chili sauce, pyramid, clock, bell, spatula, gong, pennant, rain boots, firecrackers, campanula , pressure cooker , blackboard , dragon boat


Select all wine below. A sample image is on the right.
Verify
Select all squares with
street signs
VERIFY
Security Check
It looks like a link that you're sharing might be unsafe. Please remove this link:
couchfire.com
If you're sure you want to share this link, please complete the security check below.
Please select all the photos which show a flower.
If you think you're seeing this by mistake, please let us know.
Submit
Cancel
请点击下图中所有的 文具盒
Please select all the images of pencil-boxes

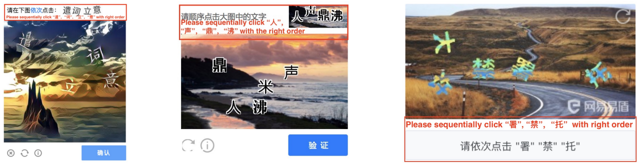
(a) Recaptcha 2015 (b) Recaptcha 2018 (c) Facebook (d) China Railway

Figure 3: Examples of selection-based image captchas.


拖动滑块拼合图片
就能验证成功哦
Slide the right puzzle window to the right region
拖动左边滑块完成上方拼图
由极验提供技术支持
拖动下方滑块完成拼图
Please slide the puzzle window to the right region
网易易盾
Please slide the puzzle window to the right region
向右滑动滑块填充拼图

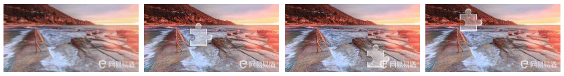
(a) GEE SlidePuzzle (b) Tencent SlidePuzzle (c) Netease SlidePuzzle

Figure 4: Examples of slide-based image captchas.


请在下图依次点击: 遣 词 立 意
Please sequentially click "遣", "词", "立", "意" with right order
遣
词
立
意
确认
请顺序点击大图中的文字
Please sequentially click "人", "声", "鼎", "沸" with the right order
人 声 鼎 沸
鼎
声
米
人
沸
验 证
网易易盾
Please sequentially click "署", "禁", "托" with right order
请依次点击 "署" "禁" "托"

(a) GEE TouClick (b) Tencent TouClick (c) Netease TouClick

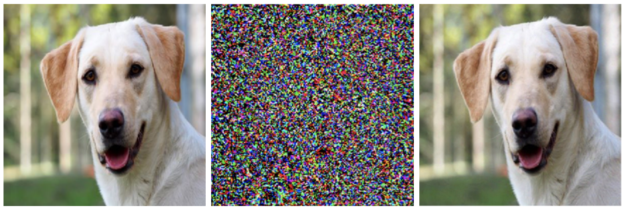
Figure 5: Examples of click-based image captchas.



(a) source image (b) generated captcha (c) generated captcha (d) generated captcha

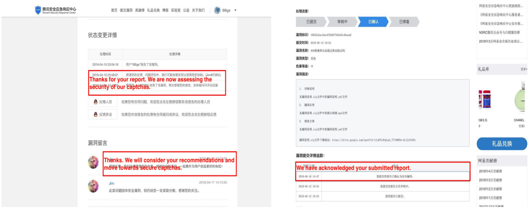
Figure 6: 3 captchas and their corresponding source image.

Figure 8 shows the responses of our reposts from Tencent and Netease. Both Tencent and Netease acknowledged our findings and recommendations.



(a) Original image (b) Elaborately crafted noises (c) Adversarial image

Figure 7: A defense example of adversarial images.


腾讯安全应急响应中心
Tencent Security Response Center
首页 提交漏洞 英雄榜 礼品兑换 博客 实验室 公益 关于我们
BBge
处理进度:
已提交 审核中 已确认 已修复
礼品库
更多+
CHANEL
礼品兑换
网易贡献榜

状态变更详情

处理时间	处理详情
2018-04-10 23:04:16	用户"BBge"报告了该漏洞。
2018-04-10 23:58:01	感谢您的反馈，问题评估中，我们可能会联系您以获得您的协助。(Jim@TSRC)

Thanks for your report. We are now assessing the security of our captchas.

处理人员 如果您有任何问题，欢迎您点击左侧按钮联系该报告的处理人员

反馈异议 如果您对该报告的处理有任何疑问或异议，欢迎您点击左侧按钮反馈

漏洞留言

Thanks. We will consider your recommendations and move towards secure captchas.

Jim 2018-04-11 14:13:55
此类问题因非安全漏洞，但仍给您一定奖励分数，感谢您的关注。

漏洞提交详情追踪:

We have acknowledged your submitted report.

2018-04-16 14:47	您提交的报告已确认为安全漏洞。
2018-04-12 19:54	您提交的报告正在审核中。
2018-04-12 19:54	您的报告已提交。

2018年4月贡献榜
2018年3月贡献榜
2018年2月贡献榜
2018年1月贡献榜

(a) Tencent's response (b) Netease's response

Figure 8: Responses from Tencent and Netease.