ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649, pp. 110–117
http://ceur-ws.org/Vol-1649, Series ISSN 1613-0073, © 2016 M. Kopp, M. Pištora, M. Holeňa

How to Mimic Humans, Guide for Computers

Martin Kopp[1,2], Matouš Pištora[1], and Martin Holeňa[1,3]

[1] Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, 160 00 Prague
[2] Cisco Systems, Cognitive Research Team in Prague
[3] Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod Vodárenskou věží 2, 182 07 Prague

Abstract: This paper studies reverse Turing tests to tell humans and computers apart. Contrary to classical Turing tests, the judge is not a human but a computer. These tests are often called Completely Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHA). The main purpose of such test is avoiding automated usage of various services, preventing bots from spamming on forums, securing user logins against dictionary or brute force password guessing and many others.

During years, a diversity of tests appeared. In this paper, we focused on the two most classical and widespread schemes, which are text-based and audio-based CAPTCHA, and on their use in the Czech internet environment. The goal of this paper is to point out flaws and weak spots of often used solutions and consequent security risks. To this end, we pipelined several relatively easy algorithms like flood fill algorithm and k-nearest neighbours, to overcome CAPTCHA challenges at several web pages, including state administration.

Keywords: CAPTCHA, machine learning, network security, optical character recognition, speech recognition

1 Introduction

In the past few decades, the rise of the internet has revolutionised our lives. We use it for work, study, socialising, shopping and many other activities on a daily basis. With the increasing popularity of the web, many public services have became a target of a malicious activity of some kind. There were attempts to, e.g., exploit mail servers for sending massive amounts of spam messages, create numerous fake profiles on social networks or make fraudulent offers on online marketplaces. In order to block the access of automated scripts and bots, the web sites had started to use various captchas[1] based security protocols in hopes of ensuring their safety. Over the years, such schemes have evolved in one of the standard security measures.

The acronym CAPTCHA stands for Completely Automated Public Turing test to Tell Computers and Humans Apart, and was coined in 2003 by von Ahn et al [19]. The fundamental idea of its authors is to use a yet unsolved hard AI problem which is easy for humans to solve. In theoretical informatics, the Standard Turing test [18] is defined as a test in which a human judge is supposed to consistently distinguish whether he/she is communicating via text with a human counterpart or a computer pretending to be a human. However, for the automatic and effective testing, the judge must also be a computer. This is where captcha, often called a reverse Turing test, comes into play.

Nowadays, a captcha is a program that generates a test which the majority of humans are able to solve, but current computer programs are not. Its mainly used on websites to distinguish whether the user is a human or a robot. The need for this type of challenge arose with the increasing amount of internet bots and automated scripts attempting to exploit public web services. Nowadays, it is an established security mechanism to prevent mailing spam messages, mass posting on internet forums, mass voting in online polls and downloading files in large amounts.

An interesting work has been done by the Microsoft researcher Chellapilla [11] who calls these tests Human interaction proofs. His work focuses on distinguishing effective distortion features and specifying best practices for designing captchas which are resistant to computers while remaining relatively easy for humans to solve. He also states that, depending on the cost of the attack, automated scripts should not be more successful than 1 in 10 000 attempts, while human success rate should approach 90%. It is generally considered a too ambitious goal, as random guesses can be successful [10], and consequently, a captcha is considered compromised when the attacker success rate surpasses 1%.

This is a work in progress, and we started it with websites that are most familiar to our everyday life, which are websites in Czech. More precisely, we focused on webpages of the state administration and similar to show them the vulnerability of sometimes critical systems of the national infrastructure. The main purpose of this paper is to show that the captcha schemes used on such webpages are easy to solve and therefore unsafe and to alarm the responsible offices. This is especially alarming on the webpages like State Office for Nuclear Safety or the Czech State Administration of Land Surveying and Cadastre.

The rest of this paper is organised as follows. The related work is briefly reviewed in the next section. Section 3 surveys the current captcha solutions. Section 4 presents our approach to breaking text-based and audio-based captcha challenges. The experimental evaluation is summarised in Section 5 and the paper closes with a conclusion.

[1] We will write captcha in lowercase for typographical reasons.

2 Related work

Most papers about breaking captcha heavily focus on some particular scheme. As an example may serve [12] with scheme reCapthca 2011. To our knowledge, the most general approach is presented in [6]. This approach is based on effective selection of the best segmentation cuts. It was tested on many up-to-date text-based schemes with better results than most of specialised solutions. But even that work was focused solely on the text-based schemes. We focused our efforts in a different way and instead of targeting one particular scheme, we tried to break captchas of different types, but all in the Czech internet environment. Unfortunately, we found only text-based and audio based captcha. Therefore, we tried to break both of them on several web sites including the Czech State Administration of Land Surveying and Cadastre[1] or the State Office for Nuclear Safety[4].

The most recent approaches use neural networks like [16]. The results are still not that impressive compared to the previous approaches, but the neural-net-based approaches improve very quickly. We intend to use convolution neural networks in our future work as well. But in this paper we tried to use as simple techniques as possible and show that even with them, we were able to compromise all captcha schemes presented in this study.

Not all captcha schemes support the audio as an alternative. Consequently, there was not that much effort spent in this topic. One of the first really successful attacks is well described in [17], followed by even greater success in [8]. More recent results of the same team are presented in [7]. The reason for our investment into audio captcha is to decide if it is generally easier to break text-based or audio-based captcha when both are available. Again, we used only the most simple techniques to point out the vulnerability of audio-based captchas.

An excellent assessment of humans success rate in completing captcha challenges can be found in [9]. As our paper is work in progress, we have human results only for the audio-based schemes.

3 Captcha schemes survey

This section surveys the currently available captcha schemes and challenges they present.

3.1 Text-based

The first ever use of captcha was in 1997 by the software company Alta-Vista, which sought a way to prevent automated submissions to their search-engine. It was a simple text-based test which was sufficient for that time, but it was eventually proven ineffective. At that time, the computer recognition rates of single characters were already on par with those of humans, and thus the development of captchas shifted to the prevention of segmentation like noise addition, cluttering and other various anti-segmentation techniques. With the effort to prevent breaking of captchas with increasing the amount of distortion and cluttering, the challenges faced the risk of becoming almost illegible. The design of human friendly, yet secure captchas becomes a serious challenge. The most commonly used techniques to prevent automatic recognition can be divided into two groups called anti-recognition features and anti-segmentation features.

The anti-recognition features such as the use of different size of characters in multiple fonts was a straightforward first step to the text-based captcha schemes. Those and other anti-recognition features, like character rotation, are typically no problem for humans because we see it on everyday basis. The only exception is distortion. Distortion is a technique in which ripples and warp are added to the image. It is one of the easiest and most effective ways of reducing the classifier accuracy. But excessive distortion can make it very difficult even for humans and thus usage of this feature slowly vanishes. Due to advances in pattern recognition and optical character recognition, all those features became obsolete and were to some extend replaced by anti-segmentation features.

The anti-segmentation features are not designed to complicate a single character recognition but instead they try to make the segmentation of the captcha image unmanageable, preserving the readability by humans. The first two features used for this purpose were added noise and confusing background. But it showed up that both of them are bigger obstacle for humans than for computers. After that the occlusion lines appeared in the wild. A good implementation of occluding lines is one of the most effective and human-friendly ways of preventing segmentation, an example can be seen at Figure 1. The most recent feature is called negative kerning. It means that the neighbouring letters are moved so close to each other that they can eventually overlap. It showed up that humans are still able to read the overlapping text with only a small error rate, but for computers it is almost impossible to found the right segmentation.

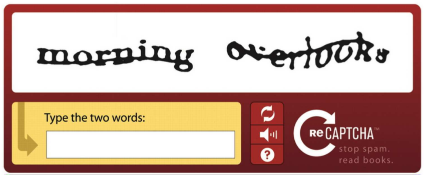


Figure 1: Older Google reCaptcha with the occlusion line.

3.2 Audio-based

From the beginning, the adoption of captcha schemes was not the ideal state. Users were annoyed with captchas that were hard to solve and had to try multiple times in order to solve them. The people affected the most were those with visual impairments or various reading disorders such as dyslexia. Soon, an alternative emerged in the form of audio captchas. Instead of looking at the image and transcribing the displayed characters, the user was given the option, usually alongside with a traditional text-based captcha, to play a sound puzzle and write the characters that he/she heard. In order to remain effective and secure, the captcha has to be resistant to automated sound analysis. For this purpose various background noise and sound distortion are added. Still a human visitor should have no problem in hearing and recognising the code. Generally, this scheme is now a standard option on major websites that implement captcha.

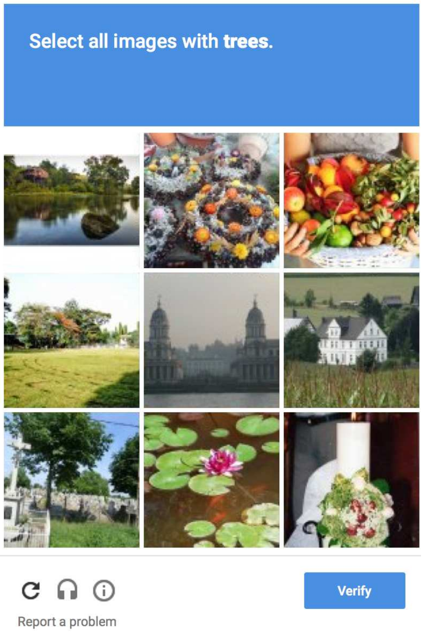
The major anti-automation tools are changing speakers, involving both males and females of ages ranging from children to retired. Most of the current solutions rely on the added noise. The level of sophistication is very diverse, ranging from buzz, singing birds to human speakers played backwards.

3.3 Image-based

With the advancement of captchas, criticism soon began to appear. The obstacle of solving a puzzle every time someone wants to enter a site is at least annoying and discouraging for the common user. It is in the everyones best interest to keep the customer satisfied all the time and make their user experience the most pleasant. In order to preserve security against spam-bots, new captcha designs were developed. The most prominent design was image-based captcha. The user is presented with a series of images showing various objects and the task lies usually in detecting which of them have a common topic and selecting them. For example a user is shown a series of images of various landscapes and is asked to select those with trees, like in Figure 2. This type of captcha has gained huge popularity on touchscreen devices like tablets and smart phones, where simply tapping the screen is the preferable option over typing the code.

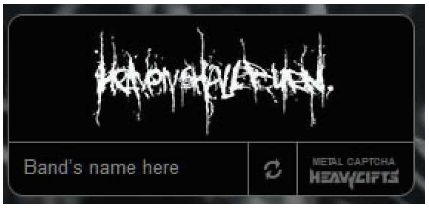
3.4 Other types

In parallel with the image-based captcha developed by google and other big players, many alternative schemes appeared. They are different variations of text-based schemes hidden in video instead of distorted image, some simple logical games or puzzles. As an example of an easy to solve logical game we selected the naughts and crosses, Figure 5. As a special type of of text-based scheme can be considered the metal captcha. This scheme shows to the user not the automatically distorted characters but a logos of metal bands which are typically unreadable, see Figure 3. All of those got recently dominated by Google's noCaptcha button, Figure 4. They say that this single button can distinguish between humans and computers. It uses browser cookies and somehow track user behaviour on the webpage, but implementation detail were not disclosed.



Select all images with **trees**.

Figure 2: Current Google reCaptcha with image recognition challenge.



Band's name here

Figure 3: An example of the HeavyGifts group Metal Captcha.

4 Recognition pipeline

In this section, the algorithm pipelines for both text-based and audio-based captcha schemes are described. We are aware that there are some very advanced approaches e.g. [6, 16] but we intentionally used simple algorithms in the basic pre-process, segment and recognize pipeline. Our motivation is to show that even using simple approaches, most currently used captchas in the Czech internet environment can be compromised.

4.1 Text-based

The text-based captchas are still the most widely used ones. Their goal is to present an image with distorted characters using anti-recognition and anti-segmentation features combined in such a way that humans can easily read it but computers do not. Our goal, on the contrary, is to successfully recognize all those characters automatically.

The first step in the intended pipeline is conversion of an image to the grayscale. The image is converted from

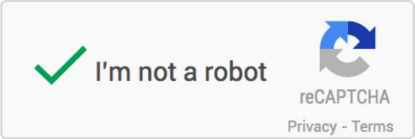


Figure 4: Current Google noCaptcha button.

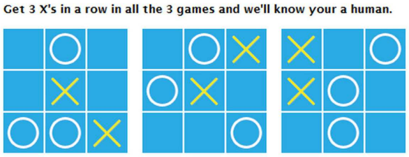

Get 3 X's in a row in all the 3 games and we'll know your a human.

Figure 5: A naughts and crosses game used as a captcha.

the RGB colorspace to the greyscale space according to the following equation:

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B. \tag{1}$$

This equation was adopted in the Rec. BT 601 standard by the International Telecommunication Union [14].

The image is then transformed to a binary image by a thresholding method. Pixels with an intensity higher than the threshold are converted to the white colour and those with a lower intensity are converted to black. For a given threshold T the equations is:

$$Y(x) = \begin{cases} 0 & \text{if } x < T \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

The threshold is computed by iterating through all possible thresholds and selecting the one which minimises the within-class variance. This method was proposed by Otsu in [13]. The class probabilities and the class variances are computed from the image brightness histogram:

$$\sigma_\omega^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{3}$$

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \tag{4}$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i) \tag{5}$$

where p is a greyscale level probability and L is the number of the greyscale levels.

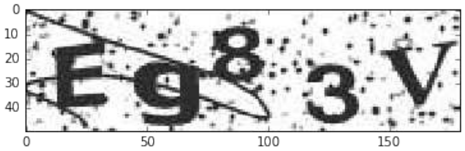


Figure 6: An example of a greyscale cuzk captcha with characters *Eg83V*

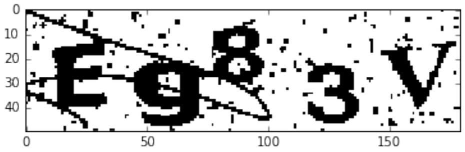


Figure 7: The thresholded captcha example

Next part is a noise removal. For this we used morphological operations followed by the flood fill algorithm. Morphological operations are a simple yet powerful approach to remove speckles and occluding lines. With the closing operation, we can fill small holes and gaps in the image, and with the opening operation loosely connected segments are disjointed and small points and lines are removed. The four basic binary morphological operations: dilation $\oplus$, erosion $\ominus$, opening $\circ$ and closing $\bullet$ are defined as follows:

$$X \oplus H = \{(x,y) : H_{(x,y)} \cap X \neq \emptyset\} \tag{6}$$

$$X \ominus H = \{(x,y) : H_{(x,y)} \subseteq X\} \tag{7}$$

$$X \circ H = (X \ominus H) \oplus H \tag{8}$$

$$X \bullet H = (X \oplus H) \ominus H \tag{9}$$

where X is the original image, H the structuring element and $H(x,y)$ the translation of H by the vector (x,y). The effect of the closing operation can be described as erasing the object border and then regrowing it back. If in the first step an object is small enough to be considered a border as a whole, there is subsequently nothing to regrow and thus it is deleted.

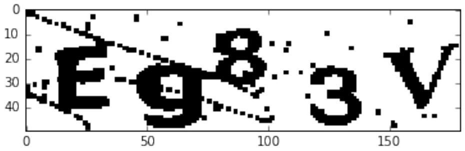


Figure 8: The effect of one iteration of closing

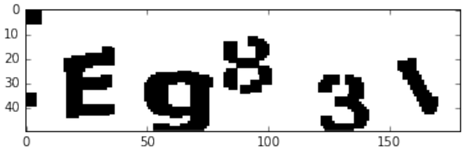


Figure 9: Deterioration of character details after three iterations of *closing*

The next approach is to count areas of all connected components (in terms of pixels it contains) and delete the ones with the area below a certain threshold. The idea is to iterate on each pixel of the image and when a white pixel is found a flood fill algorithm is used to count the number of pixels in the area. Individual characters are large objects and such can be easily distinguished from noise by empirically setting a certain threshold. The objects with area count below the threshold are then deleted, which results in an almost noiseless image.

Even with our simplistic approach, only the individual characters and a few lines remain. At first we isolate all the objects left in the image, which is done by iterating through every pixel. When an unlabelled pixel with a foreground colour is found, the flood fill algorithm is used to paint it with a new unique colour. Due to the nature of occluding lines, their position is generally horizontal. That is unlike any of the characters the captchas contain and as such the isolated lines can be easily eliminated by deleting all objects with their height under an empirically set threshold.

If the number of isolated objects is the desired number of characters, a captcha is considered successfully segmented. In the other case, we have two possibilities. If the number of objects is greater than number of characters, it implies that there are some speckles or line segments left. They are eliminated by deleting objects with the lowest pixel count. This usually provides good results. If there are fewer objects than the number of characters it indicates a connection of multiple characters either by a remaining line or by collapsing. This situation is resolved by the X-axis projection algorithm.

Its main idea for two or more joined characters is that the pixel count between them is generally lower than in the centre of the character. First, we construct the X-axis projection by summing pixels of each column. Next, all local minima are found which will be later considered for cutting points. The next step is to remove all local minima which have their pixel count under the empirically set threshold to eliminate most cutting points positioned in the middle of a character. All possible segmentations into two parts left are then considered for the subsequent classification. Finally the cutting point which maximises the classification performance is selected. Fortunately, this is a really rare event.

When the segmentation step is done, each segment is resized to 20x20 pixels, resulting in a vector of 400 binary values. These vectors are then used as features for the *k*-nn classifier. Parameters of the *k*-nn classifier are discussed in Section 5.

4.2 Audio captcha

For the audio-based captchas the pipeline is even simpler. The most advanced audio captcha looks like the one at Figure 10. A human speaker with a lot of noise making it very hard to do a good segmentation. Contrary, the ones we found on the Czech internet looks more like Figure 11. A synthetic voice was used and the level of added noise is almost negligible. Therefore, we can simply skip the noise cancelation step. Furthermore, the segmentation is much simpler than in the text-based case. The audio data are normalised to zero mean and unit variance. The segmentation is done based on amplitude thresholding with an empirically set threshold.

According to [15], speech signals are time-varying signals, which are stationary for a short time periods (5-100 ms). The change of the signal then reflects different phonemes. The information in a speech signal is actually represented by a short term amplitude spectrum of the speech wave form. Therefore, we split the character wave form into 10 bins, extracted means and variances of amplitudes from each bin and used them as features. The last feature is the length of sound wave in seconds.

Those feature vectors, containing 21 scalar values, are then presented to a *k*-nn classifier.

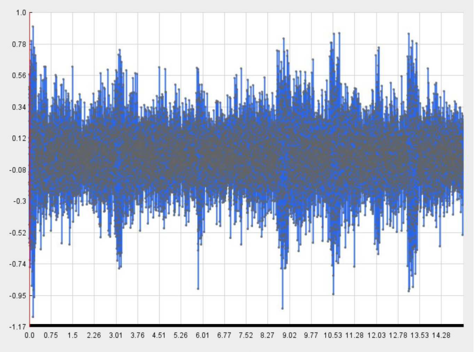


Figure 10: The visualisation of audio captcha from Securimage containing phonemes "h86gpd". The added noise effectively covers gaps between characters.

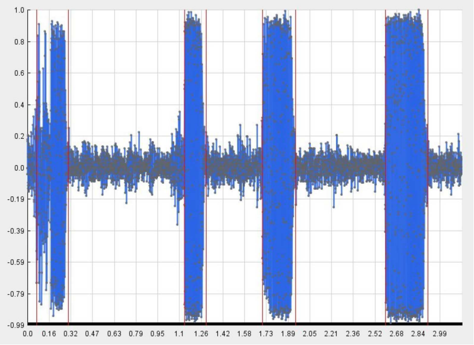


Figure 11: The visualisation of audio captcha from uloz.to. Added noise is weak and the phonemes can be simple separated by thresholding the amplitude.

5 Experimental evaluation

This section describes all the experiments we have done so far, setting of *k*-nn parameters for both audio and text-based captchas and evaluating of the successful recognition rate for each analysed scheme. Because this is work in progress, there are still some missing values and not all experiments were finished yet.

We have tested text-based captchas recognition at the following web sites: cuzk.cz[1], mojedatovaschranka.cz[3], sujb.cz[4], uloz.to[5], centralniregistrdluzniku.cz[2] and the audio-based captchas recognition at: sujb.cz[4] and again uloz.to[5].

5.1 Parameters setting

For the parameters setting, we used together 510 text-based and audio-based captchas, which were manually labelled. We used a 3-fold cross-validation, entailing 340 samples for training and remaining 170 for testing.

Our experimental results on the uloz.to dataset suggests that the best option for the text-based captchas are manhattan distance and $k = 6$, see Figure 12. For the audio-based captchas the graph looked pretty similar, but we used euclidean instead of manhattan distance. It showed up that the euclidean metric is the best and together with $k = 9$ it achieved recognition rate 86,5%, followed by the cosine metric with 84.1%.

The uloz.to was chosen as the primary testing dataset for multiple reasons. It has the most advanced captcha we found on the Czech internet in both text-based and audio-based cases. We didn't found any design or implementation flaws like for e.g. for the cuzk.cz web site. Therefore, we expected the parameters set on the uloz.to dataset will be robust enough even for the other schemes and according to Figure 13, this is more or less true.

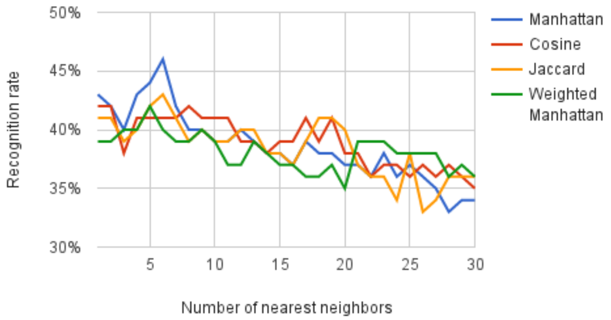


Figure 12: Comparison of different metrics and the influence of increasing k for text-based captcha.

5.2 Results

uloz.to The uloz.to is a file sharing service which uses captchas to prevent automated file downloading. They support both text-based and audio-based schemes. Their text-based scheme is very good compared to others we analysed. They use distortion, rotation a lot of noise and occluding lines. Their audio captcha use one synthetic voice with addition of a weak noise signal.

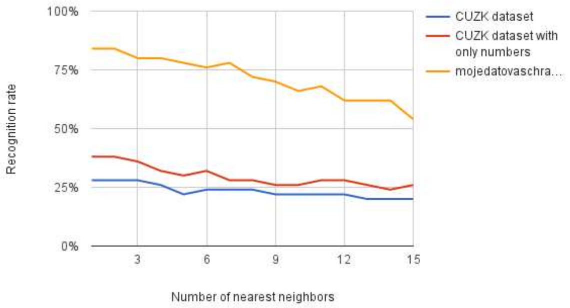


Figure 13: Checking the robustness of parameters setting estimated on uloz.to on other schemes.

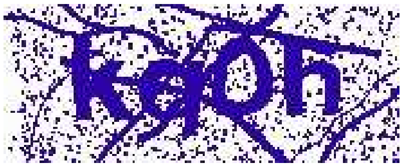


Figure 14: Example of text-based captcha from uloz.to.

We have analysed 510 samples of audio and text-based challenges. Our average recognition rate for whole captchas estimated by 10-fold cross-validation was 14% for text-based and 86% for audio-based captchas. The 14% recognition rate does not seem much, but lets recall that there is the 1% threshold to consider a captcha scheme compromised. Furthermore, we have tested up to ten humans to solve the random audio captchas and their success rate ranged from 54% to 76%. This in fact means that the computers are better than humans in test which should tell them apart.

sujb.cz The State Office for Nuclear Safety uses a captcha to secure their public forum. Both text-based and audio-based schemes are available and easy to solve. Both schemes lack noise and anti-recognition features. The text-based scheme has occluding lines, but they have a different colour than characters so it is easy to filter them out.

The overall recognition rate was 98% for audio-based and 86% for text-based captchas. But we have to admin that we used only 50 images and audio files to obtain those results.

cuzk.cz The Czech State Administration of Land Surveying and Cadastre uses only the text-based captcha to disable automatic queries to their database. The images generated by their scheme look well on the first sight but there is a serious design bug. The captcha shown on an im-

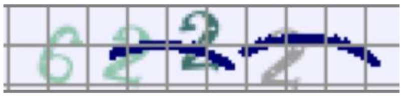


Figure 15: Example of text-based captcha from sujb.cz.

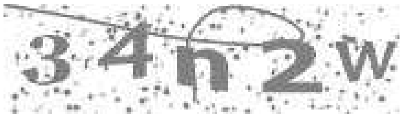


Figure 16: Example of text-based captcha from cuzk.cz.

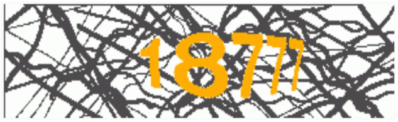


Figure 17: Example of text-based captcha from mojedatovaschranka.cz.

age is not a standard GIF or JPEG format but rather a *.axd* file, which is the HTTP Handler used by ASP.NET applications. Therefore, the image is generated on runtime. Simply refreshing the image (not the whole page) then generates a new captcha challenge containing the same characters.

Thanks to the bug, we were able to obtain and label 2100 different images. This flaw can be easily exploited to achieve a nearly 100% precision, by downloading more and more images until we are sure about correct recognition. To be fair we did not used this bug in our evaluation and still were able to obtain 46% captcha recognition rate.

mojedatovaschranka.cz This scheme is pretty weak, lacking any anti-segmentation features, with a differently coloured noise and using only digits. Our result is a 82% success recognition rate over the testing set of 50 samples.

centralniregistrdluzniku.cz This page serves as the central registry of debtors and captcha must be solve before you can upload your customer experience with some company. Adopted scheme is again easy to solve, the distortion is weak and occluding lines have a different colour than the characters. Our result is a 61% success recognition rate over a testing set of 50 samples.

5.3 Summary

The final results are summarised in Table 1. The reported numbers are captcha recognition rates, estimated by a 10-fold cross-validation. Some values are missing, because the audio-based captcha alternative is available only for uloz.to and sujb.cz.

Finally, the overall misclassification overview is given for the text-based captcha in Figure 19, and for the audio-based in Table 2.

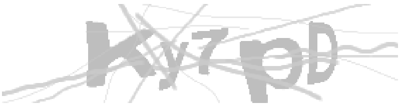


Figure 18: Example of text-based captcha from centralniregistrdluzniku.cz.

webpage	audio	text
uloz.to	0.86	0.14
sujb.cz	0.98	0.86
cuzk.cz	-	0.46
mojedatovaschranka.cz	-	0.82
centralniregistrdluzniku.cz	-	0.61

Table 1: The summary off successful recognition rates on all tested captcha challenges.

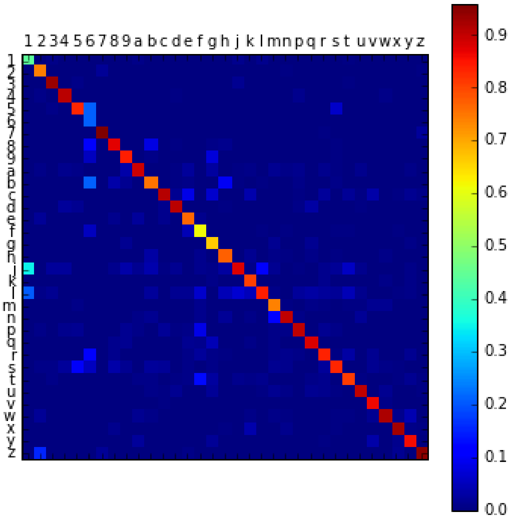


Figure 19: Confusion matrix for all text-based schemes.

6 Conclusion

This research was driven by curiosity of security enthusiasts and will be used for academic purposes only. None of us have any malevolent or business intentions.

We have tested the security of several captcha solutions across Czech internet environment. We intentionally used the out of the shelf algorithms to simulate simple attacks. The final result is that the current state is alarming. All tested solution have been compromised with recognition rate highly over 1%. The most secure solution was the text-based scheme at uloz.to, where we achieved only 14% recognition rate. On the other hand we were about 10% more accurate than humans in terms of average recognition rate on their audio-based captchas.

The second most secure were challenges generated at the web site of the Czech State Administration of Land Surveying and Cadastre. The captcha is used to block automated queries to the database and it should prevent massive downloads of private informations about the ownership of real estates. Our recognition rate was almost one half, more precisely 46%. But due to the design flaw of this captcha, described in Section 5, it can be easily boosted to almost 100% precision.

The key messages of this paper should be: do not rely on any captcha as the only defence agains automation and

phoneme	success	misclassified as	
	rate	phoneme	rate
a	97.2	r	2.8
b	96.8	t	3.2
c	82.1	s	10.7
d	92.1	r	5.3
e	92.3	a	3.8
f	96.3	x	3.7
g	96.3	n	3.7
h	97.4	k	2.6
i	100	-	-
j	93.1	a	3.4
k	100	-	-
l	80	r	20
m	96.8	b	3.2
n	100	-	-
o	100	-	-
p	93.3	o	6.7
q	96.4	r	3.6
r	100	-	-
s	87.5	x	3.1
t	87.9	k	6.1
u	89.3	o	3.6
v	95.7	g	4.3
x	92.3	s	2.6
z	95.1	g	2.4

Table 2: The misclassification rate for the audio captchas.

never use captcha as the only security solution and for the attacker it is: if you can choose, try audio captchas, they are typically easier to break.

As to our future work, we are still preparing a more complete survey of captcha solutions used on the Czech internet. We are especially searching for more state administration pages, that use completely insufficient solutions or design flaws. Currently we are devoting our research efforts to the application of convolution neural networks in this context as we believe that they can replace our whole text-based pipeline. We are also starting to pay attention to image-based captchas like the one in Figure 2

Acknowledgement

The research reported in this paper has been supported by the Student Grant SGS16/119/OHK3/1T/18.

References

[1] Nahlížení do katastru nemovitostí [online], 2004-2016. [Cited 2016-06-01].

[2] Centrální registr dlužníků [online], 2016. [Cited 2016-06-01].

[3] Datové schránky [online], 2016. [Cited 2016-06-01].

[4] Státní úřad pro jadernou bezpečnost [online], 2016. [Cited 2016-06-01].

[5] Ulož.to [online], 2016. [Cited 2016-06-01].

[6] 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)*, 2014.

[7] Elie Bursztein, Romain Beauxis, Hristo Paskov, Daniele Perito, Celine Fabry, and John Mitchell. The failure of noise-based non-continuous audio captchas. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 19–31. IEEE, 2011.

[8] Elie Bursztein and Steven Bethard. Decaptcha: breaking 75% of ebay audio captchas. In *Proceedings of the 3rd USENIX conference on Offensive technologies*, page 8. USENIX Association, 2009.

[9] Elie Bursztein, Steven Bethard, Celine Fabry, John C Mitchell, and Dan Jurafsky. How good are humans at solving captchas? a large scale evaluation. In *2010 IEEE Symposium on Security and Privacy*, pages 399–413. IEEE, 2010.

[10] Elie Bursztein, Matthieu Martin, and John Mitchell. Text-based captcha strengths and weaknesses. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 125–138. ACM, 2011.

[11] Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski. Designing human friendly human interaction proofs (hips). In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 711–720. ACM, 2005.

[12] Claudia Cruz-Perez, Oleg Starostenko, Fernando Uceda-Ponga, Vicente Alarcon-Aquino, and Leobardo Reyes-Cabrera. Breaking recaptchas with unpredictable collapse: heuristic character segmentation and recognition. In *Pattern Recognition*, pages 155–165. Springer, 2012.

[13] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[14] ITUR Rec. Bt 601: Studio encoding parameters of digital television for standard 4: 3 and wide-screen 16: 9 aspect ratios. *ITU-R Rec. BT*, 656, 1995.

[15] Urmila Shrawankar and Vilas M Thakare. Techniques for feature extraction in speech recognition system: A comparative study. *arXiv preprint arXiv:1305.1145*, 2013.

[16] F. Stark, C. Hazırbaş, R. Triebel, and D. Cremers. Captcha recognition with active deep learning. In *GCPR Workshop on New Challenges in Neural Computation*, 2015.

[17] Jennifer Tam, Jiri Simsa, Sean Hyde, and Luis V Ahn. Breaking audio captchas. In *Advances in Neural Information Processing Systems*, pages 1625–1632, 2008.

[18] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[19] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In *Advances in Cryptology—EUROCRYPT 2003*, pages 294–311. Springer, 2003.