https://doi.org/10.2352/EI.2024.36.4.MWSF-330

Captchas on Darknet Marketplaces: Overview and Automated Solvers

York Yannikos, Julian Heeger; Fraunhofer SIT / ATHENE; Darmstadt, Germany

Abstract

Captchas are used on many websites in the Internet to prevent automated web requests. Likewise, marketplaces in the darknet commonly use captchas to secure themselves against DDoS attacks and automated web scrapers. This complicates research and investigations regarding the content and activity of darknet marketplaces.

In this work we focus on the darknet and provide an overview about the variety of captchas found in darknet marketplaces. We propose a workflow and recommendations for building automated captcha solvers and present solvers based on machine learning models for 5 different captcha types we found. With our solvers we were able to achieve accuracies between 65% and 99% which significantly improved our ability to collect data from the corresponding marketplaces with automated web scrapers.

Introduction

The darknet is a phenomenon that has captured the attention of the media and the public in the last decade. Darknet marketplaces in particular have gained popularity as platforms for illicit activities. However, despite their presence in the media, there remains a continuing lack of knowledge regarding the offered content and user activity within these marketplaces. Since darknet marketplaces have flourished in recent years, they have also been in focus of research and, of course, investigations by law enforcement agencies. This lead to various solutions that automatically collect data from these marketplaces to gain insights or to perform monitoring regularly.

Most marketplace operators then implemented countermeasures like captchas against automated web requests, explaining that these were necessary against distributed denial of service attacks, e.g. from competing individuals. However, captchas also prevent automated bots from scraping the marketplaces and analyze the collected data. Therefore, researchers and law enforcement agencies need to develop techniques to bypass these countermeasures that complicate or prevent their data collection. Since darknet marketplaces can not simply rely on very robust captcha variants like reCAPTCHA that are hard to defeat reliably, the implemented captchas are often self-made or based on publicly available toolkits for captcha generation. This could be exploited to build automated solutions that can defeat such captchas.

In this work we provide an overview about different captcha types used on darknet marketplaces. We describe solvers for 5 captcha types, and propose a workflow that allows a quick development of new captcha solvers based on machine learning.

Darknet Marketplaces

Darknet marketplaces are primarily found in the Tor network, but lately other darknets like I2P are also becoming popular for marketplace operators. In Tor, the marketplaces are onion services, i.e. anonymously hosted sites only reachable within Tor. They are typically built like large web shops with a wide range of products in different categories. The access to a marketplace usually requires a registered account and the login / register form is typically protected by a captcha. After solving the captcha. After the first login a user typically gets a short introduction into market rules and security-related advice and is then able to browse the marketplace.

Larger darknet marketplaces often look very similar with many different categories and subcategories, often ten-thousands of product offers, hundreds and sometimes thousands of vendors. Product offers typically contain information about the vendor, shipping, country of origin, payment, quantity and price. Vendor profile pages usually contain short descriptions, often PGP public keys and vendor ratings from customers with short review messages. Darknet marketplaces hosted as onion services provide technical anonymity for the participants. The operators usually employ strict security setups, e.g. enforcing all network traffic from and to the site to go through Tor, avoiding content service providers or hosting services in the clearnet, and requiring users to disable Javascript in their browser. Therefore, the captcha services used on darknet marketplaces are self-hosted and mostly self-developed, often based on publicly available open source libraries. This makes them usually much easier to solve automatically than professionally developed captchas like Google's reCAPTCHA. Also, images on darknet marketplaces are often directly embedded in the HTML source as base64-encoded data which simplifies scraping such web pages.

Captchas

To create an overview about used captchas on darknet marketplaces in Tor, we started monitoring websites like `onion.live` or `darknetone.com` that aggregate and rank marketplaces in Tor, and we checked individual marketplaces if they use captchas. During our research we analyzed 27 darknet marketplaces and could identify 15 different commonly used captcha types out of a total of 30 captchas used (see Table 1). While 10 of the 30 captchas we found were text-based with different complexity, others were based on images or shapes where the user had to identify objects, find missing pieces in a puzzle, or correctly read a clock. Figure 1 shows examples for the 15 different captchas.

We found that several marketplaces like White House Market or Kerberos used more than one captcha, e.g. one shown at the account registration form and another at the login form, or when a high amount of consecutive web requests were made. Some marketplaces did not use any captchas at all, which often indicates that these sites are fake marketplaces that try to convince new customers to pay for goods they will never receive. We were

Table 1: Captcha types per marketplace

Marketplace	Onion URL	Captcha Type
Abacus	abacuseeettcn3n2zxo7tqy5vsxhqpha2jtjqs7cgdjzl2jascr4liad.onion	Object Rec.
Archetyp	4pt4axjgzmm4ibmxplfiuvopxzf775e5bqseyllafcecryfthdupjwyd.onion	Open Circle
ARES	sn2sfdqay6cxztroslaxa36covrhoowe6a5xug6wlm6ek7nmeiujgvad.onion	Text
ASAP	asap2u4pvplnkzl7ecle45wajojnftja45wvovl3jrvhangeyq67ziid.onion	Moving Window
Bohemia	bohemiaobko4cecexkj5xmlaove6yn726dstp5wfw4pojjwp6762paqd.onion	Anti-Phishing
Cartel Marketplace	mgybzfrldjn5drzv537skh7kgwgbq45dwha67r4elda4vl7m6qul5xqd.onion	Clock
Cocorico	xv3dbyx4iv35g7z2uoz2yznroy56oe32t7eppw2l2xvuel7km2xemrad.onion	Anti-Phishing
Colombia Connection	eg5pj3r4xhybxgfkjnkhbhwgkuonp5wtla3mbpuzphzk6lxkhftnvuyd.onion	Moving Window
Cypher	6c5qaeiibh6ggmobsrv6vuilgb5uzjejpt2n3inoz2kv2sgzocymdvyd.onion	Text
Dark Bazar	vpqhwsisxuishmvxhyzy3mh5rntyd3hwyb2oj5y7atipalid3ufp67qd.onion	Text
DeepMarket	2hek7b3bml77x7q2uvsqcmzoagb2gw5gf2oqaqhue7tfipfby3bhmzqd.onion	Select Time
Digital Thrift Shop	kw4zlnfhxje7top26u57iosg55i7dzuljjcyswo2clgc3mdliviswwyd.onion	–
Incognito	incognitox3vs5grdnmh52k35m64vib5fsbdrxzilujjptiqzeyrxhid.onion	Move Pattern
Kerberos	kerberosazmnfrjinmftp3im3cr7hw4nxbavm4ngofn64g24be7h3kqd.onion	Assoc., Text, Math
Kingdom Market	kingdom6txlkrt7wba5r4lfiimfvueyzxnjfpmzutb4fqoxr6qaxajyd.onion	Sort Numbers
MGM Grand Market	duysanjqxo4svh35yqkxxe5r54z2xc5tjf6r3ichxd3m2rwcgabf44ad.onion	Text
Nemesis Market	nemesis555nchzn2dogee6mlc7xxgeeshqirmh3yzn4lo5cnd4s5a4yd.onion	Puzzle 2
Quest Market	questxwvkwvsw2qgeeljz4fbv6cq2kbmapo7tw5heu4nng2ufgykapid.onion	Text
Retro Market	apd4upslqbr4qwywhjdlkywrvrtshcc2abc3322ff7vpiwu4lxglvwid.onion	Labyrinth
ShinyFlakes	amazing3zvkbs6wulbrqd7gzvat45qvk2h6jyw7pkwgaj7r6zu5k67yd.onion	–
TorZon Market	torzon4kv5swfazrziqvel2imhxcckc4otcvopiv5lnxzpqu4v4m5iyd.onion	Clock, Text
ToRReZ	yxuy5oau7nugw4kpb4lclrqdbixp3wvc4iuiad23ebyp2q3gx7rtrgqd.onion	Text, Text
UnderMarket 2.0	puyr3jb76flvqemhkllg5bttt2dmiaexs3ggmfpyewc44vt5265uuaad.onion	Math
Vice City Market	vice3nwnin46cmmvg4j3wd3xguaaiwlwzxt2mxkd5ng6ougfcu4rdfid.onion	Match Shapes
WeAreAMSTERDAM	amster7eb42jwbx6umy7gs4migxnu5vjp5b5t5wbdtmqv2yhx3zsb7id.onion	Text
WeTheNorth Market	hn2paw7zaahbikbejiv6h22zwtijlam65y2c77xj2ypbilm2xs4bnbid.onion	Anti-Phishing
White House Market	7yipwxdv5cfdjfpjztiz7sv2jlzzjuepmxy4mtlvuaojejwhg3zhliqd.onion	Puzzle 1, Object Rec.

able to identify such a marketplace and analyzed in [4] how the operators fake activity on their site to defraud customers.

Captcha Solvers

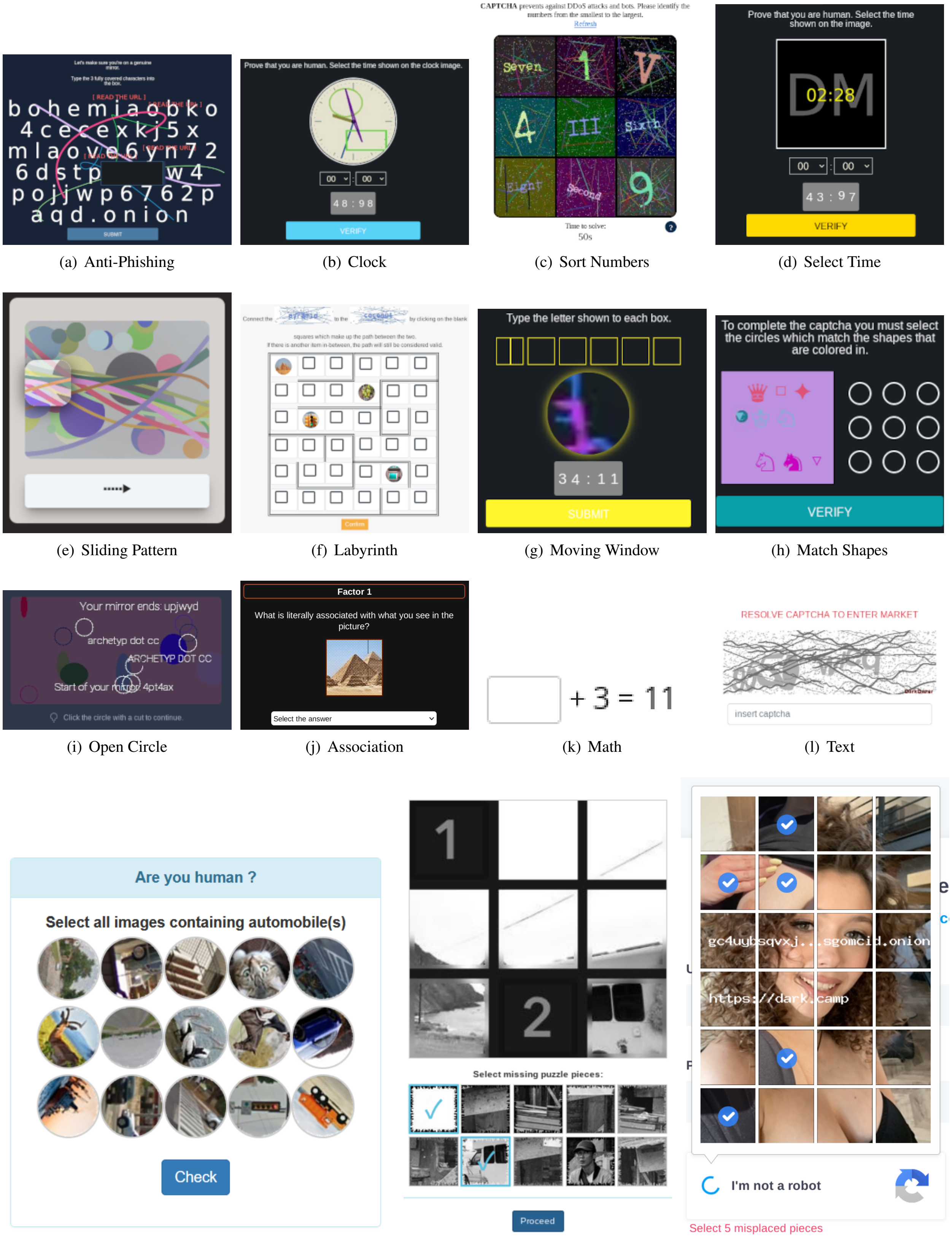
In order to build automatic captcha solvers we focused on approaches from machine learning (ML) like convolutional neural networks (CNNs) which have shown to be effective for this task [1, 8]. Since we first needed to create sufficiently-sized data sets to train our CNNs, we started looking into how the captchas were implemented. We found that several marketplaces used publicly available toolkits to generate their captchas. For these captchas we used the identified toolkits to generate large amounts of labeled training data sets for the corresponding captchas. Examples of such toolkits are *Captcha for Laravel 5/6/7/8/9* or *Securimage*. We also noticed that one specific toolkit called *EndGame* seemed to be very popular and was used to implement captchas on several marketplaces. This toolkit was created by operators of the Dread forum and White House Market and provides a clock captcha variant. For the other marketplaces where we could not find any public captcha toolkit, we automatically collected captcha images and manually labeled them with a distributed labeling tool we developed (see screenshot in Figure 2).

Sometimes additional processing of the captchas was necessary before training, e.g. because marketplaces would randomly rotate them or insert additional noise like their onion URL into the captcha image. After experimenting with CNN models like ResNet or VGG we were able to build solvers for 5 different captcha types: One object recognition captcha, one puzzle captcha, two different text-based captcha variants, and one clock captcha (shown in Figures 1(m), 1(n), 1(l), and 1(b), respectively). In the following we describe each of our solvers in detail.

Object Recognition Captcha

Object recognition captchas like the one shown in Figure 1(m) were used on Abacus and White House Market. One captcha shows 15 small images and asks the user to select all objects from a specified category, e.g. ”Select all images containing dogs”. To build a solver we first collected 1,000 captchas from White House Market. While working on the solver, we observed that the small images were often randomly rotated and reused in another captcha. We then manually labeled the solutions of the 1,000 captchas (i.e. 3 small images on average) with their categories, which resulted in a data set of 2,942 labeled images. In the next step we converted the images into RGBA color space and augmented the data set with rotated versions of the images. For training we used Tensorflow’s Keras API[1] to build a CNN based on the VGG architecture proposed by Simonyan and Zisserman [5]. We then trained our CNN with a 80/20 train/test split. After the training we collected another 1,000 new captchas from White House Market as test data and were able to achieve 78% accuracy in solving these captchas.

[1] https://www.tensorflow.org/tutorials/images/cnn



Let's make sure you're on a genuine mirror.

Type the 3 fully covered characters into the box.

Prove that you are human. Select the time shown on the clock image.

CAPTCHA prevents against DDoS attacks and bots. Please identify the numbers from the smallest to the largest.

Prove that you are human. Select the time shown on the image.

Type the letter shown to each box.

To complete the captcha you must select the circles which match the shapes that are colored in.

Your mirror ends: upjwyd

Click the circle with a cut to continue.

What is literally associated with what you see in the picture?

RESOLVE CAPTCHA TO ENTER MARKET

Select all images containing automobile(s)

Select missing puzzle pieces:

Select 5 misplaced pieces

(a) Anti-Phishing

(b) Clock

(c) Sort Numbers

(d) Select Time

(e) Sliding Pattern

(f) Labyrinth

(g) Moving Window

(h) Match Shapes

(i) Open Circle

(j) Association

(k) Math

(l) Text

(m) Object Recognition

(n) Puzzle 1

(o) Puzzle 2

Figure 1. *Different captcha types found on darknet marketplaces*

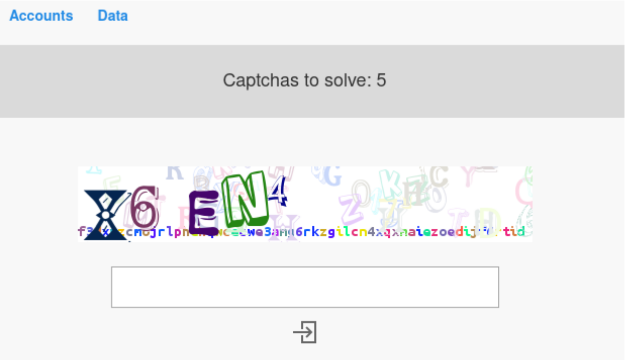


Figure 2. *Screenshot of our web-based tool for distributed manual captcha labeling*

Puzzle Captcha

For the puzzle captcha shown in Figure 1(n), which was used on White House Market, we first tried building a solver without using ML. We found that the captcha always showed two missing pieces and ten possible solution pieces to choose from, where eight of these did not belong to the captcha image. We then created all 90 possible permutations of the image by inserting every combination of two solution pieces. For each of these images we removed edge distortions and black bars between the pieces and calculated a score based on the Sobel filter to detect edges within the image. To solve the captcha we then chose the image with the lowest score, i.e. the lowest amount of detected edges. This approach worked very well and we could achieve a captcha solving accuracy of about 83% (tested on 500 new captchas from White House Market). Therefore, we did not spend further time on optimizing the solver by using an ML-based approach.

Text Captchas

We could observe several variants of text captchas on darknet marketplaces like ToRReZ, Dark Bazar, or ARES. Since we were interested in collecting data from ToRReZ market, we focused on solving the two text captchas found there. The first captcha variant shown in Figure 3 was used as DDoS protection and had to be solved before we could proceed to the login form to access the market.

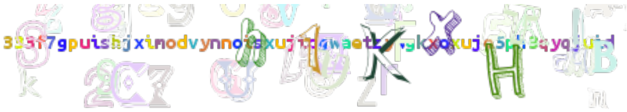


Figure 3. *Text captcha variant 1 from ToRReZ market*

In order to prepare a training data set for a CNN, we first downloaded 667 captchas from ToRReZ market and applied the following pre-processing steps: We converted each captcha to grayscale with a black background, removed the bright background characters, then converted the image to binary color and color-inverted the image. Using the resulting data set with a 80/20 train/test split, we then trained a CNN which we based on Torchvision's ResNet-18 model[2]. After we reached an accuracy of about 65% we considered this sufficient for our crawling purpose.

The second captcha variant shown in Figure 4 was used on the login form of ToRReZ market.

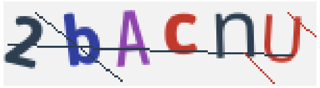


Figure 4. *Text captcha variant 2 from ToRReZ market*

Creating training data for this captcha was comfortable for us because we could identify and find the captcha generator used by the marketplace operators. We then used the same code[3] from GitHub to generate 1,000 labeled captchas and trained a CNN in combination with a Recurrent Neural Network (RNN) with a 80/20 train/test split. By that we reached an accuracy of about 90%.

Clock Captcha

We found the clock captcha as shown in Figure 1(b) on marketplaces like Cartel or TorZon. To build a solver we followed the approach proposed in [1]: We used the EndGame GitHub repository[4] to first generate labeled training data and then trained a ResNet-50 model. Like the authors described in their paper, we could also achieve an accuracy of about 99% with our solver.

While the authors of the proposed solution also found the clock captcha on the Cocorico marketplace and the Dread forum, we observed that these sites now stopped using this type of captcha and moved to another variant instead.

Workflow

After implementing our solvers we integrated them in a microservice deployed with a RESTful API. This helped us to develop new darknet marketplace scrapers that could communicate with our captcha solvers in a unified way. Based on our experiments and results we propose the following workflow for a quick development and integration of captcha solvers for darknet marketplaces (also shown as diagram in Figure 5:

1. Research on the used captcha toolkit to find out if it is publicly available
2. If the toolkit could not be identified or found, collect a sufficient amount of captchas from the marketplace for labeling
3. Create a high-quality training set of labeled captcha samples (we recommend using at least 1,000 captcha samples, ideally generated by identified toolkits, if available)
4. Train suitable CNN models like ResNet or VGG
5. If necessary, tune parameters and/or improve pre-processing to achieve a sufficient accuracy for web scraping
6. Integrate the trained model in a microservice with an API for communication with web scrapers
7. Scrape the marketplace while automatically solving the captchas using the API

[2] `https://pytorch.org/vision/stable/models/resnet.html`

[3] *Captcha for Laravel 5/6/7/8/9*: `https://github.com/mewebstudio/captcha`

[4] *EndGame V2 - Onion Service DDOS Prevention Front System*: `https://github.com/onionltd/EndGame`

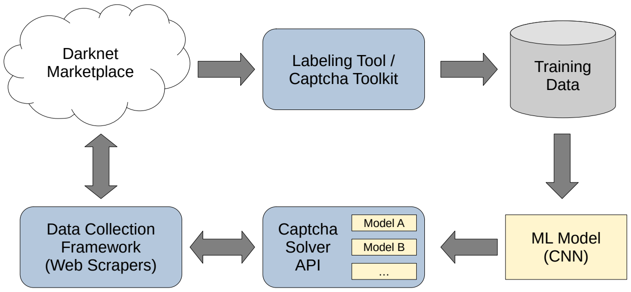


Figure 5. Workflow for scraping darknet marketplaces protected by captchas

Discussion

We tested our captcha solvers with the corresponding marketplaces and could measure accuracies between 65% and 99%, depending on the individual captcha. However, we argue that even a rather low accuracy of 20–50% is already sufficient to enable automated data collection on darknet marketplaces: First, even humans often do not solve captchas with a much higher accuracy (e.g. if texts or objects are not easy recognizable), so it is not very likely that we would raise the suspicion of marketplace operators when crawling their site. Secondly, solving a captcha only in every third or fourth attempt does not slow down automated data collection significantly and still allows very fast web scraping and crawling (especially when using multiple scrapers in parallel). And thirdly, since we are conducting this research in the darknet, we have the freedom to fail without consequences: Even if marketplace operators would detect our automated web requests, they could not block us effectively, e.g. based on our IP range, as we could evade any blocking attempts by establishing a new Tor circuit.

Related Work

Previous research on captcha solving mostly discusses individual approaches to solve single captcha variants but rarely provides insights about which captchas are actually used in the darknet. Other research focuses on defeating very robust captcha variants like reCAPTCHA which can not be used by darknet marketplace operators because it provides a point of attack on their anonymity.

In [3] the authors investigated 41 darknet marketplaces and 35 vendor shops. They present an overview about the used access and authentication mechanisms (including captchas) as well as observations about products and purchases, shipping and delivery, vendor reputation, support, disputes and community aspects.

In [1] the authors present a solution for the clock captcha they found on three darknet sites. They were able to achieve an accuracy of about 99% using a found captcha toolkit to generate labeled training data for training a CNN based on ResNet-50. We also used the proposed approach in this work and could achieve similar accuracy.

Zhang *et al.* propose a framework for breaking text-based captcha on the dark web using a Generative Adversarial Network (GAN) to counter the typical noise used to make the text on the captcha unrecognizable [10]. With three captcha sources and an open-source synthesizer, they utilized 500 captchas from each source and several thousands from the synthesizer for training purposes. The proposed solution surpassed the state-of-the-art and enabled the authors to monitor the dark web on a large scale. The same authors updated their work in [11], providing detailed insights they gained from the darknet marketplace Yellow Brick.

Bostik *et al.* [2] present a semi-supervised approach for training a captcha-breaking system that does not require manual data set creation. The authors demonstrate that their approach is particularly successful on shorter captchas consisting of only 5 digits, resulting in an accuracy of about 95%. However, when length and complexity of the captcha text increase, the accuracy of the classification rapidly declines.

We previously published our own research on scraping single vendor shops or marketplaces like Dream Market, Wall Street Market, or White House Market [6, 7, 8, 9]. In these works we did not focus on captcha solving but on the analysis of offered goods or the relationship between users, bought goods, and transaction amounts. However, we implemented solvers for individual captchas that we encountered, which are described in more detail in the corresponding publications.

Conclusion

In this work we provided an overview about captchas used on 27 darknet marketplaces and described approaches to build automated solvers for 5 different captchas: one clock, one puzzle, two text-based, and one object recognition captcha. While CNNs (and especially ResNets) have shown to be a suitable basis to build solvers for different types of captchas, we found that for some captchas it is not necessary to use ML at all: For example, very simple text captchas could sometimes be solved just by using OCR. Other captcha types like puzzle captchas could be solved by applying standard image processing or computer vision methods.

We could also see that darknet marketplace operators often rely on publicly available captcha generators. If this is the case, the same generators could be used to create arbitrary amounts of labeled training data in order to build captcha solvers with a very high accuracy. However, we think that even a low accuracy, e.g. between 20% and 50%, is still sufficient for web scraping in the darknet because it does not affect the effectiveness and efficiency of the scraping task significantly.

Acknowledgments

We would like to thank our colleague Florian Platzer for his contribution to this work. The research for this publication was conducted as part of the SEPIA[5] project funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE.

References

[1] David Audran, Marcus Andersen, Mark Hansen, Mikkel Andersen, Thomas Frederiksen, Kasper Hansen, Dimitrios Georgoulias, and Emmanouil Vasilomanolakis. Tick tock break the clock: Breaking captchas on the darkweb. In *Proceedings of the 19th International Conference on Security and Cryptography-SECRYPT, INSTICC, SciTePress, Lisbon, Portugal*, pages 357–365, 2022.

[2] Ondrej Bostik, Karel Horak, Lukas Kratochvila, Tomas Zemcik,

[5] `https://senpai.athene-center.de/en/projects#c7154`

and Simon Bilik. Semi-supervised deep learning approach to break common CAPTCHAs. *Neural Computing and Applications*, 33(20):13333–13343, October 2021.

[3] Dimitrios Georgoulias, Jens Myrup Pedersen, Morten Falch, and Emmanouil Vasilomanolakis. A qualitative mapping of darkweb marketplaces. In *2021 APWG Symposium on Electronic Crime Research (eCrime)*, pages 1–15. IEEE, 2021.

[4] Florian Platzer and York Yannikos. Trust assesment of a darknet marketplace. In *Proceedings of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, UK*, 2023. (to be published).

[5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[6] York Yannikos, Julian Heeger, and Maria Brockmeyer. An analysis framework for product prices and supplies in darknet marketplaces. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–7, 2019.

[7] York Yannikos, Julian Heeger, and Martin Steinebach. Data acquisition on a large darknet marketplace. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–6, 2022.

[8] York Yannikos, Julian Heeger, and Martin Steinebach. Scraping and analyzing data of a large darknet marketplace. *Journal of Cyber Security and Mobility*, pages 161–186, 2023.

[9] York Yannikos, Annika Schäfer, and Martin Steinebach. Monitoring product sales in darknet shops. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, page 59. ACM, 2018.

[10] Ning Zhang, Mohammadreza Ebrahimi, Weifeng Li, and Hsinchun Chen. A Generative Adversarial Learning Framework for Breaking Text-Based CAPTCHA in the Dark Web. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6, Arlington, VA, USA, November 2020. IEEE.

[11] Ning Zhang, Mohammadreza Ebrahimi, Weifeng Li, and Hsinchun Chen. Counteracting Dark Web Text-Based CAPTCHA with Generative Adversarial Learning for Proactive Cyber Threat Intelligence. *ACM Transactions on Management Information Systems*, 13(2):1–21, June 2022.

Author Biography

York Yannikos received his Diplom (equiv. Master's degree) in computer science from the University of Rostock, Germany in 2008. Since then he has worked as research associate in the Media Security and IT Forensics department at the Fraunhofer Institute for Secure Information Technology (SIT) and at the National Research Center for Applied Cybersecurity (ATHENE) in Darmstadt, Germany. His research interests include darknet marketplaces, open source intelligence, and digital forensic tool testing.

Julian Heeger is a research associate in the Media Security and IT Forensics department at the Fraunhofer Institute for Secure Information Technology (SIT) and a researcher at the National Research Center for Applied Cybersecurity (ATHENE) in Darmstadt, Germany. He holds a Master's degree in IT security from the Technical University of Darmstadt.