Demystifying the Underground Ecosystem of Account Registration Bots

Yuhao Gao*
University of Technology Sydney
Sydney, Australia
Beijing University of Posts and Telecommunications
Beijing, China

Guoai Xu*
Harbin Institute of Technology
Shenzhen, China
Beijing University of Posts and Telecommunications
Beijing, China

Li Li
Monash University
Melbourne, Australia

Xiapu Luo
The Hong Kong Polytechnic University
Hong Kong, China

Chenyu Wang
Beijing University of Posts and Telecommunications
Beijing, China

Yulei Sui
University of Technology Sydney
Sydney, Australia

ABSTRACT

Member services are a core part of most online systems. For example, member services in online social networks and video platforms make it possible to serve users customized content or track their footprint for a recommendation. However, there is a dark side to membership that lurks behind influencer marketing, coupon harvesting, and spreading fake news. All these activities rely heavily on owning masses of fake accounts, and to create new accounts efficiently, malicious registrants use automated registration bots with anti-human verification services that can easily bypass a website's security strategies.

In this paper, we take the first step toward understanding the underground ecosystem of account registration bots, and in particular, the anti-human verification services they use. From a comprehensive analysis, we determined the three most popular types of anti-human verification services. We then conducted experiments on these services from an attacker's perspective to verify their effectiveness. The results show that all can easily bypass the security strategies website providers put in place to prevent fake registrations, such as SMS verification, CAPTCHA and IP monitoring. We further estimated the market size of the underground registration ecosystem, placing it at about US $4.8M-128.1 million per year. Our study demonstrates the urgency with which we to think about the effectiveness of our registration security strategies and should prompt us to develop new strategies for better protection.

CCS CONCEPTS

• **Security and privacy → Software and application security**.

*Yuhao Gao and Guoai Xu contributed equally to this work and share the first authorship of this work.

ESEC/FSE '22, November 14–18, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9413-0/22/11…$15.00
https://doi.org/10.1145/3540250.3549090

KEYWORDS

Account registration bots, Registration strategy, Human verification bypass, Security

ACM Reference Format:
Yuhao Gao, Guoai Xu, Li Li, Xiapu Luo, Chenyu Wang, and Yulei Sui. 2022. Demystifying the Underground Ecosystem of Account Registration Bots. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '22), November 14–18, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3540250.3549090

1 INTRODUCTION

Member services are a core part of most online systems today. For example, social networking sites like Facebook and video streaming platforms generally provide members with customised site experiences, such as video recommendations or 'you may also know' friendship feeds. We found that 94% of the top websites listed on Alexa [1] have such membership systems. But, as is well known, where there is account registration, there is also network crime. Collecting welcome coupons and fabricating false ranking data are profitable activities. Indeed, there is a small group of devious people known as the Internet Water Army [60] ready to 'flood' the internet with comments for whatever purpose and for whoever is willing to pay. As reported by Jacqueline Zote [81], online products have been embracing the power of influencer marketing because it successfully works to increase awareness of and trust in a brand. Influencer marketing is where consumers are swayed by the opinions of people we consider to be experts in a field. However, in this realm, there are those with malicious intent that perpetrate influencer fraud by creating fake accounts with followers that do not exist. A team at Mediakix experimentally shows that it is fairly easy to become a fake Instagram influencer, say, by buying fake followers [33] among other methods.

However, this demand for fake accounts raises another specter of crime. Some underground groups, called malicious registrants, have resorted to automated software to automatically create a bulk of accounts – automated software we more commonly know as registration bots. Website providers have taken many measures to prevent the negative impact of such fake accounts, such as deploying security strategies that involve different kinds of human

verification methods in the registration procedure. The most popular ones include SMS verification and CAPTCHA [59]. At the same time, detecting fake accounts by analyzing the daily activities on an accounts has also been widely used to prevent malicious registrants [66]. The consequence is that these battles between malicious registrants and websites has become a long-term war. As reported by DataDome [17], in 2019, over 2 billion fake accounts were found on Facebook. This is alarming since Facebook must boast one the best technical teams on the planet today.

Many researchers and developers have contributed to improving the security in this area [40, 44]. And, in response, malicious registrants have also entered into a cycle of continuous improvement. Underground ecosystems have sprung up where malicious registrants play out different roles and develop, test, and trade hacking technologies. Yet these groups remain relatively obscure to the research community. We lack understanding of the technologies used from the attacker's perspective. We lack knowledge of the status quo in registration bots.

This Work. Hence, with this paper, we have undertaken the first systematic study of the account registration bot ecosystem, exploring it from the attacker's perspective. To understand the techniques used by these accoutn registration bots more deeply, we first conducted a preliminary study of the security strategies used in the registration procedures. Specifically, we assembled the main human verification methods currently used in Alexa's top websites [1] (see **Section 2.1**). And, further, we also collected some registration bots from the Internet and analyzed them to find what security methods they focus on when they try to create accounts automatically. Our results show that verification methods such as SMS verification, CAPTCHA, and IP restrictions, are the most widely used website registration security strategies. Of course, this also makes them the main targets of the registration bots. With this "census" of the account registration bot ecosystem complete, a picture of the main participants and components of the malicious account registration underground ecosystem start to coalesce (see **Section 2.2**).

We then conducted an in-depth analysis of the anti-human verification services these account registration bots use to bypass human verification mechanisms. The analysis included top-line elements, such as whether the mechanisms are effective. However, we also tried to analyze their workflows and make sense of how they can amass their enormous resources such as phone numbers and IP addresses (see **Section 3**). Our next step was to conduct an experiment to evaluate the scale and effectiveness of these services. We collected resources, such as phone numbers and IP addresses, and used them on some real-world websites just like an actual attacker does (see **Section 4**). With the data collected, we further took the opportunity to estimate the impact of the malicious registration industry driven by account registration bots. This included estimates of how many resources provided by these services have been used to create fake accounts and the total market size of this underground ecosystem (see **Section 5**).

Our results provide a novel impression of the landscape of the account registration bots ecosystem and reveal some unexpected yet interesting observations:

- **Many websites use similar verification methods in the account registration process.** By analyzing the websites in Alexa's top list, we found that some similar human verification methods are used by many websites, such as SMS, CAPTCHA, and IP restrictions, which means the malicious registrants can expend less effort effectively bypassing the same methods on different websites.
- **The malicious registration ecosystem involves many elements.** Its scale is vast and its resources abundant. It includes malicious registrants, registration bot developers, account sellers and various kinds of anti-human verification services, each of which tends to comprise many components.
- **Most human verification methods are not effective.** Our results suggest that the most prevalent human verification methods, including SMS, CAPTCHA, and IP restrictions, can be successfully bypassed by anti-human verification services. This means that most registration procedures do not stop account registration bots.
- **The scale and impact of the underground registration ecosystem is enormous.** Using the data collected from our experiments with human verification services, we evaluated the impact and market size of the malicious registration ecosystem. Our estimates suggest that the number of fake accounts created by this ecosystem is vast, and that the Internet is deeply influenced by these network crimes. Moreover, the market size of this ecosystem conservatively ranges from US $4.8-128.1 million each year.

Our datasets used in this work and the tools we identified are available at https://mobile-app-research.tech.

2 PRELIMINARY STUDY

In this section, we provide the necessary background information to account registration bots, including the status quo of human verification in account registration in the wild (see **Section 2.1**) and the key players in the ecosystem (see **Section 2.2**).

2.1 Human Verification in Registration

2.1.1 Preliminary Study. To prevent account registration bots, website owners implement many different methods [17]. The most common ones add a human verification step into the registration procedures to ensure that the account is indeed (hopefully) being created by a human. While we know some common techniques like email and SMS verification or CAPTCHA, the full gamut of human verification strategies being used today is somewhat unclear. To better understand this landscape, and the corresponding ecosystem of underground registration bots designed to thwart these prevention mechanisms, we need a systematic review. Only with this will we form a comprehensive picture of the security strategies in play, the primary targets for registration bots, and the malicious registrant ecosystem overall. Hence, we conducted an exploratory study of the current practices in human verification strategies. By manually examining a set of 200 popular websites – Alexa's top websites [1] – we identified and summarized the verification strategies used. The results follow in the next section.

[1]The Alexa Top website list [1] is a collection of websites with top influence based on network traffic. The list of 200 websites can be found in our dataset.

Table 1: Human verification methods used by top websites.

Restriction methods	Number of websites	Description.	Sample
SMS	84	Users need to provide an SMS verification code sent by the website.	www.baidu.com
Text CAPTCHA	46	Users need to enter the text shown in a picture.	www.360.cn
Google reCAPTCHA	22	Google reCAPTCHA is a human verification component developed by Google includes click puzzle, smart click and invisible CAPTCHA.	reddit.com
Sliding puzzle	15	Users need to drag a piece of the picture to complete the puzzle.	www.jd.com
Slider	12	Users need to drag a square from left to right.	www.taobao.com
Click puzzle	9	Users need to click on different parts of the picture in order according to the instructions.	www.yy.com
Third party account	8	Users need to log in with an account on a third-party website.	www.v2ex.com
Smart click	4	Users need to click a button.	www.babytree.com
Phone voice	3	Users need to answer the call and provide the text heard to the website.	mail.ru
Funcaptcha	2	Funcaptcha developed by Arkose Labs provides human authentication components such as invisible CAPTCHA and rotating puzzle CAPTCHA, requiring users to rotate a picture to the correct direction.	roblox.com

2.1.2 Human Verification Strategies in the Wild. The verification strategies used on our selected popular sites are summarized in Table 1.

Verification Methods. We identified 10 different kinds of verification methods from these 200 top websites, as shown in Table 1. SMS and various forms of CAPTCHA are the most widely used. Due to the privacy of mobile phone numbers, an SMS verification code is the most common. As for CAPTCHA, this tool, born for human verification, now has a rich array of types, such as inputting the letters in a picture or clicking to solve a puzzle. Furthermore, there are now a host of third-party CAPTCHA components such as Funcaptcha, developed by Arkose Labs [3] and Google reCAPTCHA. A few websites use some more unique methods like charging a registration fee.

Verification Strategies. A good user experience and service security both are critical for website operators. As a result, a balance between the two is vital when combining several verification methods into one registration security strategy (e.g. a combination of SMS verification and text CAPTCHA). In our research, we found that the verification strategies of most websites comprise no more than two kinds of methods. The most common strategy is a combination of SMS verification and CAPTCHA. Some shopping websites use three methods, including two kinds of CAPTCHA simultaneously.

2.2 The Ecosystem of Registration Bots

2.2.1 Preliminary Study. To the best of our knowledge, online account registration bots have not yet been systematically studied, and our community lacks an understanding of the ecosystem in which they operate. To this end, we first conducted a preliminary study towards understanding the key players and the workflow of registration bots. Specifically, we searched for available registration bots from the Internet and our collaborating anti-virus company. From this exercise, we harvested 100 automated registration tools (the registration bots). Figure 1 shows an example of one account registration tool.

Then we manually looked into their workflows and analyzed the services they used. Interestingly, all of the tools have three kinds of anti-human verification services embedded within them, including two that can bypass the human verification methods used by most websites: an SMS receiving service that can bypass SMS verification, and a CAPTCHA recognition service to bypass CAPTCHA verification. We also noticed a third embedded service, an IP proxy service,

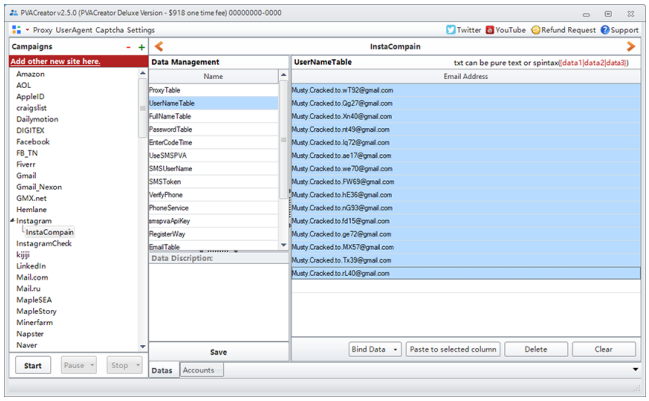


Figure 1: An example of automatic account registration tools.

which is used to bypass IP restrictions. Although it is not easy for users to directly see, almost all the registration tools use a proxy service. So, we regarded these as a third anti-human verification service to add to our research objectives.

2.2.2 Key Players in the Ecosystem. With this observation as the starting point, we manually traced these services and the users/developers involved to form a fuller picture of the ecosystem. Figure 2 shows the overall ecosystem of account registration bots. The major players in the ecosystem are introduced next.

(1) Malicious Registrants. As shown in Figure 2, some technical groups known as malicious account registrants are the core operators of the whole underground registration ecosystem. These players are connected to every other role. With automatic registration tools integrated with anti-human verification services, they can create accounts and sell them to account sellers.

Some malicious registrants have development teams that can customize a registration tools according to their own needs. Further, they can continually improve them by tracing any upgrades to a website's security strategies. These customized tools are more flexible, bypassing some special registration protection, such as complex JavaScript and back-end interactions.

Other registrants can only use registration tools developed by *tool developers*. So their targets heavily depend on the tools they can obtain, or can afford to pay for.

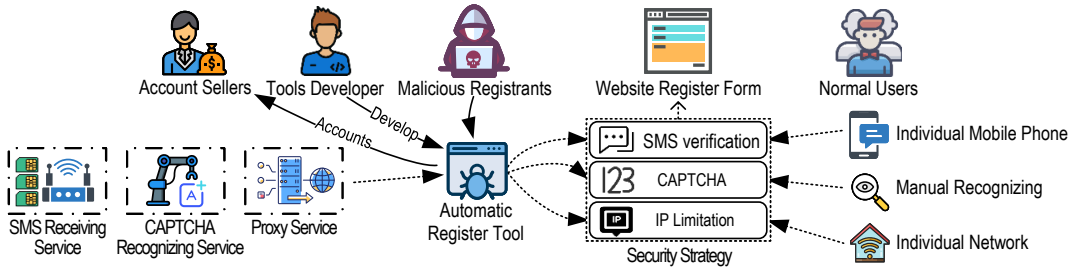


Figure 2: The ecosystem of malicious registration

(2) Anti-human Verification Services. Anti-human verification services provide a range of methods for bypassing the verification mechanisms websites put in place to ensure "I am a not a robot." As part of this study, we found the most popular services used by registration bots were *SMS receiving services* to bypass SMS verification, *CAPTCHA recognizing services* to solve CAPTCHAs, and *IP proxy services* to bypass IP restrictions. Additionally, these services are usually constantly updated to ensuring they keep pace with the latest security measures.

(3) Tools Developers. To provide registration tools for malicious registrants, *tool developers* analyze the registration procedure and develop a tool integrated with anti-human verification services, that targets a particular website. They earn money from selling their tools but they also get share bonuses from the anti-human verification services they build-in with a personal channel ID.

(4) Account Sellers. *account sellers* sell website accounts through various channels such as the dark web, social networks, and underground online trading platforms. The sale price of an account often relates to the website's influence, the scale of the account, and the ease of registration. Additionally, ongoing fake account activity is sometimes needed to bypass fake account detection and keep the account available.

3 DEMYSTIFYING ANTI-HUMAN VERIFICATION SERVICES

In this section, we detail the three primary services used by registration bots to bypass human verification methods. These are: the SMS receiving service; the CAPTCHA recognition service; and the IP proxy service, as shown in Figure 3.

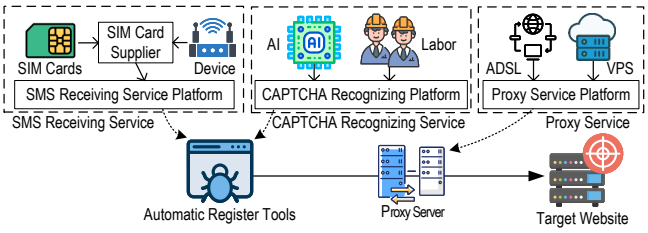


Figure 3: Services that bypass registration restrictions

3.1 SMS Receiving Service

3.1.1 Overview of SMS Receiving Services. SMS receiving services can bypass SMS verification, widely used by many websites. Generally, to create an account, a human user must provide one mobile phone number not already registered with the site. On this number, they receive an SMS code containing digits or letters. To bypass this verification step, the SMS receiving service will provide a mobile phone number for receiving SMS codes.

How can they collect so many mobile numbers? Generally, SMS receiving services do not own any SIM cards. Instead, the *SIM card supplier* are the actual owners of mobile phones. Collecting and owning numerous mobile phone numbers is very difficult, especially in some countries like China where there is a mobile phone identification policy. So, instead, they collect SIM cards through various channels, such as buying from low-income people, recycling mobile phone numbers that are about to be abandoned, and even buying them from the black market. To operate all these SIM cards simultaneously, they use a particular *modem pool device*, that has dozens of slots for cards. This makes it possible to send or receive text messages with these SIM cards. SIM card suppliers connect their devices to a dedicated piece of client software provided by an SMS receiving platform. With this, their SIM cards automatically bring in dollars.

3.1.2 The Workflow of an SMS Receiving Service. Acquiring phone numbers for an SMS receiving service is largely based on a standard workflow, as shown in Figure 4. The SMS receiving platform provides users with various API interfaces, such as HTTP API and an Android APP. The workflow of different platforms is similar.

The Reuse of Mobile Numbers. For even more financial gain, the SMS receiving service will reuse their phone numbers on different websites. For example, user A and user B can use the same phone number to receive SMS codes from websites A and B simultaneously. For this reason, users must select a target website (called a *project*), before receiving an SMS code. SMS receiving platforms use text message templates to distinguish text messages from different websites because every website sends its verification SMS in a fixed format. A typical platform has thousands of projects to choose from, and users can also submit new ones on an as needs basis.

The Workflow of SMS Receiving Services. The workflow of an SMS receiving services. To receive an SMS code with an SMS receiving service: ❶ the user needs to obtain a list of projects and choose their target website. ❷ The receiving service then allocates a phone number, ❸ which the user submits to the target website.

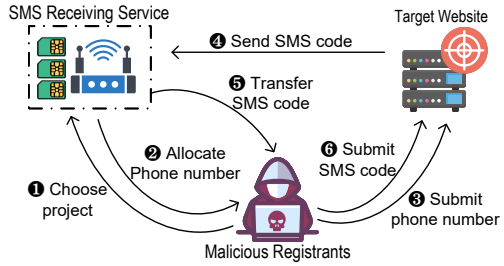


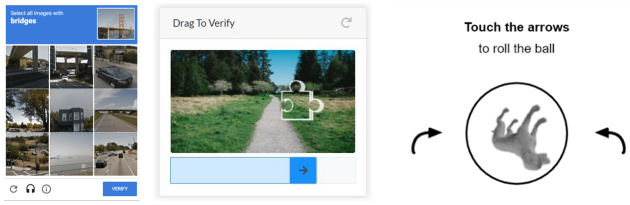
Figure 4: The workflow of SMS receiving services

❹ The website then sends an SMS code to the phone number, and ❺ the platform transfers it to the user and charges for this service. Lastly, ❻ the user submits the obtained SMS verification code to the target website and completes the account creation.

3.2 CAPTCHA Recognition Service

3.2.1 Overview of CAPTCHA. CAPTCHA is designed to distinguish humans from programs. With a long history of development, there are now many forms of CAPTCHA [43], including text-based CAPTCHA [11], image-based CAPTCHA [57], game-based CAPTCHA [67] and others [28]. For the purposes of studying these services, we have divided them into three main types: text CAPTCHA, interactive CAPTCHA, and Non-sense CAPTCHA, the former two are shown in Figure 5.

(a) text CAPTCHA



(b) interactive CAPTCHA

Figure 5: CAPTCHAs

Text CAPTCHA. Text CAPTCHA is the most convenient and straightforward CAPTCHA. A picture containing several digits, letters or a simple math problem is displayed to the user and they type what they see or the calculation result into a box. Pictures with wavy, ghostly, distorted letters are used to strengthen the security of text CAPTCHA. Nevertheless, these pictures are also problematic for users to recognize, which can lead to a frustrating user experience. Researchers are now trying to generate new text CAPTCHAs with machine learning techniques to avoid these imperfections [40, 44]. However, very few of them have been deployed in practical applications as of yet.

Interactive CAPTCHA. Interactive CAPTCHA is a newer type of CAPTCHA, which has been designed to improve usability and security at the same time. For the convenience of research, we designated all CAPTCHAs that require user interaction other than typing letters as interactive CAPTCHAs. This includes sliding puzzles, and sliders, click puzzles, and intelligent click, etc. Interactive CAPTCHAs generally rely on fairly complex technology that requires running JavaScripts on the browser and verifying behavior on a server. This feature makes interactive CAPTCHAs a more complicated tool to deploy. As a result, most websites can only use interactive CAPTCHA services provided by third parties (e.g., GeeTest [23] and Google [25]).

Non-sense CAPTCHA. To further improve user experience, a new kind of CAPTCHA called a non-sense CAPTCHA has been developed (e.g. Google reCAPTCHA v3 invisible [25]) that does not require the user to take any deliberate actions. Instead, the CAPTCHA module tracks the user's mouse trajectories and browsing behavior on the page and detect bots through complex algorithms and models. However, due to the complexity of the human verification algorithm, only a few select service providers, such as Google [25] use non-sense CAPTCHA.

Figure 6: The workflow of CAPTCHA recognition services

3.2.2 Workflow of a CAPTCHA Recognition Service. **How can a CAPTCHA be bypassed?** It may come as a surprise to learn that a CAPTCHA recognition service can solve a CAPTCHA puzzle They can even solve non-sense CAPTCHA tasks, where no puzzle is even displayed. According to the information provided by one CAPTCHA recognition service and some previous studies [8][7], most simple text CAPTCHAs are solved with a machine learning model. As a result, text CAPTCHAs are relatively quick and cheap to bypass. As for the other types of CAPTCHAs, which are too challenging for the machines to solve, the services resort to human labor. The last one is the non-sense CAPTCHA. Because there is no apparent interaction with users directly, CAPTCHA recognition services have to use different methods to bypass them, and we are not privy to the technical details of how this is done. However, based on the workflow we traced, we infer that they can crack the data collection algorithm in running a web browser and fake the same false data by machine learning or by collecting similar data from one or several manual registration attempts. Additionally, although the technical difficulty of cracking an interactive CAPTCHA or non-sense CAPTCHA is enormous, there are not many suppliers that can bypass CAPTCHA recognition services. And the few there are focus on cracking one or more of several widely used third-party interactive CAPTCHA services like Google reCAPTCHA [25].

The Workflow of CAPTCHA Recognition Services. One way to think about a CAPTCHA recognition service is that it is preparing answers for automatic registration bots, such that their workflow is like a man-in-the-middle attack [61]. As shown in Fig 6, ❶ the bots are run by malicious registrants who observe the puzzle (e.g., a picture in text CAPTCHA, an interactive CAPTCHA or some

network request for data in a non-sense CAPTCHA), and then ❷ transfer it to the CAPTCHA recognition service. The service solves the problem either with a machine or manually with human labor, and ❸ sends the answer to the bot. Lastly, ❹ the bot sends the answer to the website to finish the registration.

Notably, the formats of the puzzles and the answers needed to solve different CAPTCHAs are diverse. The most simple is the text CAPTCHA, the puzzle is a picture, and the answer is the letters or digits included in it. For an interactive CAPTCHA, it is a little more complicated. The question is a picture in general. Nevertheless, the format of an answer, which is an action command, depends on the CAPTCHA type. For example, the solution to a click puzzle might be a series of click position coordinates. For a sliding puzzle, the solution might be a drag distance. To solve the puzzle, the bot has to emulate those actions. As for a non-sense CAPTCHA, a solution at the JavaScript code level is essential for the bot. It needs to capture the parameters in a web browser as a puzzle, send those parameters to the service, and receive some data as an answer. With these data, the bot can build a network request to finish the registration. Additionally, the level of detail in the task of "behaving like a human while registering" is immense for non-sense CAPTCHAs, which means there is no standard resolution for all non-sense CAPTCHA.

3.3 Proxy Service

3.3.1 Overview of Proxy Service. Limiting access to a website based on one's IP address is a traditional technique, which is widely used by many websites and even some cloud services (e.g., Cloudflare [14], Google Cloud [26]). The main purposes of IP restrictions are to defend against DDOS attacks and to optimize the availability of services. According to our analysis of registration tools, we can infer that registration is also under an IP restriction. It is a barrier for malicious registrants to create a bulk of new accounts used by standard Internet access services for homes or businesses, which usually provides fewer IP addresses. Proxy services solve this problem by supporting a great many available IP addresses and forging IP addresses for each registration. In fact, due to IP restrictions, automatic registration bots are not the only users of IP proxy services; some illegal internet spiders also use them.

How can they get a mass of IP addresses? According to our investigation, IP proxy services use ADSL [58] network access services, which provide a different IP address for every dial. Alternatively, a cheap VPS [65] can be used, which has a static IP address for each VPS. Once these resources have been amassed, an IP proxy service can provide many IP addresses of different qualities and features. A proxy address's lifetime and anonymity are its two most important features. Lifetime refers to the period the address is available for. This can range from several minutes to several days. Anonymity means a server cannot detect a request under an IP proxy by checking some HTTP protocol header fields, such as REMOTE_ADDR, HTTP_VIA, HTTP_X_FORWARDED_FOR. A proxy server program such as Squid [46] use these fields to show the HTTP requests are transferred via a proxy server. In fact, without specific configures, Squid will set HTTP_X_FORWARDED_FOR as the client's actual IP address when it forwards an HTTP request.

3.3.2 The Workflow of Proxy Services. Thanks to the modern network libraries in every programming language, developers can use

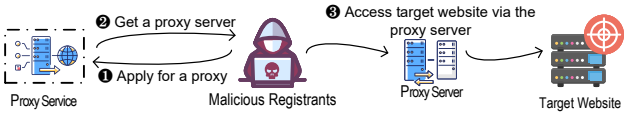


Figure 7: The workflow of proxy services

IP proxy addresses easily by drawing on either built-in libraries or third-party libraries, such as sockets in Python or Boost.Asio [5] for C++ requests. As a result, the workflow of proxy service is quite straightforward. As shown in Fig 7, when a registration bot needs to access the target website via a proxy server, the first step is to: ❶ apply for an available server from the IP proxy service. ❷ Then the service allocates an IP proxy server to the bot and sends the result, generally containing the proxy server's IP address, network port, and protocol. Lastly, ❸ the bot accesses the target website via the given proxy server. In addition, many proxy services also provide a kind of VPS with an ADSL dialling client program. Users can use these VPS as ordinary and use the dialling client program to restart their network connections and get a new IP address. But some work [76] proposed methods that can detect malicious registrations under the same local network.

4 MEASUREMENT OF ANTI-HUMAN VERIFICATION SERVICES

4.1 Research Questions

Our study aim to answer the following research questions (RQs):

- RQ1 How many SMS receiving services are there in the wild and how many mobile phone numbers can be abused for spam account registration?
- RQ2 Can existing CAPTCHA recognition services successfully bypass the diverse verification mechanisms implemented by popular websites?
- RQ3 Can an IP proxy service provide enough services to efficiently fulfill the needs of large-scale account registrations?
- RQ4 Can these anti-human verification services be used together for automatic account registration to the popular websites?

To answer these research questions, we undertook a series of experiments to measure the effects of the three most commonly used anti-human verification services on a real-world dataset of websites. We further tried to use these services on real-world websites to confirm if they were able to bypass the security strategies of websites and actually register new accounts.

4.2 SMS Receiving Service

4.2.1 Data Collection. To collate as many SMS receiving services as possible, we searched for and collected 26 SMS receiving services from a search engine and a webpage collection of services located at `www.aimazu.com`.

To understand how many phone numbers these SMS receiving services provide, we developed a mobile number crawler based on the service's workflow that automatically and continuously retrieved mobile phone numbers from an SMS receiving platform. The crawler takes phone numbers like a regular user but releases them immediately after they are taken. And because no SMSs are

either sent or received in that time, the platform does not charge any money. That said, there are a couple of that charge a fee for requesting too many mobile phone numbers without receiving any SMSs. With this method, we crawled these 26 platforms for 31 days and in return, retrieved a total of 1.6 billion pieces of data, including over 8.6 million different phone numbers.

Table 2: The phone numbers of SMS receiving services

ID	Domain	#AVG daily active phone numbers	#AVG daily new phone numbers	#AVG lifetime (days)
1	51ym.me	1,451,529	61,540	15.93
2	mangopt.com	951,641	11,271	16.01
3	cherryun.com	220,676	9,557	15.54
4	haima668.com	28,208	2,487	13.9
5	baiwanma.com	77,838	2,918	13.73
6	w6888.cn	25,746	3,890	15.57
7	yika66.com	77,864	2,003	24.42
8	yzm7.com	27,010	907	13.58
9	fxhyd.cn	1,451,601	61,931	16.26
10	js-yzm.com	24,704	6,799	7.36
11	shou-ma.com	77,844	3,175	13.57
12	51zggj.com	223,934	19,579	11.06
13	fxyzm.cn	18,507	3,497	7.3
14	yiyun66.com	15,247	3,424	7.82
15	517orange.com	139,917	7,067	16.38
16	ximahuang.com	244,441	21,400	12.8
17	api.ctep.cn	283,102	26,598	8.28
18	zxjmpt.com	12,408	2,794	13.46
19	66yzm.com	629,141	21,404	14.82
20	xinheyz.com	1,453,282	20,352	16.96
21	47.244.115.89	6,351	101	4.71
22	120.78.91.0	3,476	1,144	10.89
23	kmiyz.com	12,760	5,452	10.79
24	mili18.com	28,384	7,925	12.11
25	20982098.com	51,554	1,340	10.36
26	web.166idc.com	47,278	1,281	12.61

4.2.2 The Measurement of SMS Receiving Services. We analyzed these phone numbers from three perspectives for each SMS receiving service: daily active phone numbers, daily new phone numbers and the average lifetime of phone numbers, as shown in Table 2.

Daily Active Phone Numbers. The number of active phone numbers provided by an SMS receiving service is the most critical technical specification. Defined as the number of phone numbers crawled daily, daily active phone numbers directly relates to how many users the platform can serve and whether it can sustainably provide services. This is also the most important indicator for evaluating the size of an SMS receiving platform. According to the daily active phone numbers, these services can be roughly divided into three sizes: large, medium, and small. Larger services have the most mobile phone numbers, and they can provide users with hundreds of thousands of phone numbers every day. Medium-sized platforms can provide tens of thousands every day. As for the small-scale ones, they can only provide about 10,000 or fewer phone numbers per day. Furthermore, we also noticed that the number of phone numbers provided by each service is generally stable. The amount changes by no more than 10% per day.

Daily New Phone Numbers. Generally, most websites have a restriction that users can only create one account with one valid phone number. Due to such restrictions, if malicious users want to create many accounts, they must use a massive number of different phone numbers. For this reason, the number of phone numbers provided by an SMS receiving services is extensive, and they need to add new phone numbers continually. After many users use these numbers to register accounts, it can be challenging for users to find an unused phone number with which to create a new account. Our data show a correlation between the daily new phone numbers and daily active phone numbers on each service. Services with more active phone numbers per day also have more new phone numbers. We also noticed that the number of daily new phone numbers on each SMS receiving service was relatively stable, just like the daily active phone numbers.

The Lifetime of Phone Numbers. Although many new phone numbers are a valuable resource for a malicious registrant, if these phone numbers can only be used once, they are unlikely to help successfully bypass a website's verification system. This is because after creating a new account, they generally need to perform more operations, such as activating the account or getting their welcome coupon. For this reason, it is essential that the SMS verification process can use the same phone number several times. Indeed, almost every SMS receiving service provides an API to request a specific phone number when it is still available in this platform. To study the exact period the phone numbers provided by these services are available for, we defined the period from the first day to the last day of a phone number appearing in our records as its lifetime. Our result shows that the lifetime of most phone numbers is around 10-15 days. Only a few of them had a lifetime of more than 15 days or less than ten days.

Answer to RQ1: *We identified over 8.6 million different phone numbers from 26 SMS receiving services over a period of 31 days. Our experimental analyses reveal that (1) There are very many SMS receiving services in the wild. (2) These SMS receiving services hold a mass of active phone numbers, some with hundreds of thousands of phone numbers. (3) These services add tens of thousands of new phone numbers daily. (4) These phone numbers have a lifetime of about 10-15 days to allow for multiple uses.*

4.3 CAPTCHA Recognition Services

To analyze the features of CAPTCHA recognition services, we conducted different experiments on text CAPTCHA and interactive CAPTCHA given their workflows are different.

Text CAPTCHA. Although they can be bypassed effortlessly by machine learning models, many websites choose text CAPTCHA as their account verification system. Typically, most CAPTCHA recognition services can identify the text quickly and solve the problem accurately. However, depending on the CAPTCHA settings, occasionally the task can be very challenging, not only for the machines but also for humans. To evaluate the effectiveness of CAPTCHA recognition services with text CAPTCHA, we collected 100 CAPTCHA pictures from each of the 32 websites using text CAPTCHA in our sample. We then used five CAPTCHA recognition services to recognize these 3,200 CAPTCHA pictures and manually checked the accuracy of the results. As shown in Table 3, from a total of 160 test cases (one website and one CAPTCHA recognition service), more than 93% cases had an accuracy rate of higher than

Table 3: The recognizing accuracy of five CAPTCHA recognition services in the real world

Target Websites	chaojiying	yundama	chaorendama	fateadm	jsdati
360.cn	94%	94%	91%	94%	78%
sina.com.cn	85%	91%	84%	88%	92%
alipay.com	89%	88%	86%	91%	85%
tianya.cn	84%	89%	86%	85%	95%
huanqiu.com	91%	94%	97%	98%	95%
caijing.com.cn	95%	92%	100%	93%	100%
jrj.com.cn	73%	91%	**27%**	86%	99%
baike.com	67%	80%	77%	85%	78%
cdstm.cn	100%	100%	98%	99%	98%
163.com	82%	90%	85%	94%	**40%**
jiameng.com	**36%**	70%	72%	81%	**0%**
scol.com.cn	75%	64%	55%	69%	79%
chinadaily.com.cn	96%	89%	87%	92%	90%
sonhoo.com	51%	55%	**46%**	**37%**	**48%**
zol.com.cn	78%	83%	78%	86%	74%
bzw315.com	94%	91%	94%	96%	95%
91jm.com	99%	91%	99%	99%	100%
efu.com.cn	83%	85%	79%	85%	72%
zhiding.cn	89%	94%	86%	94%	92%
focus.cn	71%	63%	66%	67%	**43%**
gusuwang.com	84%	63%	75%	89%	78%
haofang.net	93%	88%	92%	99%	94%
photofans.cn	93%	95%	95%	99%	96%
ceconline.com	100%	99%	98%	96%	96%
gamersky.com	75%	87%	93%	85%	50%
ibicn.com	65%	86%	86%	92%	87%
cpic.com.cn	98%	97%	97%	96%	98%
cnki.net	90%	83%	89%	90%	83%
weibo.com	91%	95%	99%	94%	96%
Apple.com	94%	96%	93%	87%	77%
Tribunnews.com	93%	94%	98%	94%	98%
Wikipedia.org	64%	59%	**10%**	**38%**	67%

50%, only 10 cases were under 50%, labelled in red. This result suggests that the automatic registration bots were able to bypass real-world text CAPTCHAs by calling the service no more than twice in most cases.

Interactive CAPTCHA and Non-sense CAPTCHA. Because most interactive and non-sense CAPTCHA cannot be solved by a machine, breaking these verification methods commands a higher price by CAPTCHA recognition services. Here, the complicated test procedures and arduousness of checking the results meant that we did not test all the websites. Instead, we selected five websites and used the CAPTCHA recognition service to solve them on one CAPTCHA recognition service that was able to solve all types, including click puzzles, slide puzzles, and non-sense CAPTCHAs. We found that they all be solved successfully and with higher accuracy (more than 96% on average) than text CAPTCHA. To be specific, we first obtained 100 CAPTCHA pictures from each website, except the non-sense CAPTCHA, which do not have any displayed puzzles. Then we sent them to the CAPTCHA recognition service. Unlike the text CAPTCHA, we got some coordinates as feedback, and we printed them in these pictures, making it easier for us to check the result manually. Next, we checked these results one by one, in the same way as we did with the text CAPTCHA. As for the nonsense CAPTCHA, we chose two websites that deploy Google reCAPTCHA and tested them 100 times. By watching the server's return value when we submitted the answer, we were able to check whether verification had been bypassed successfully.

> **Answer to RQ2:** *In most popular websites, the CAPTCHA recognition service can bypass the diverse CAPTCHA verification mechanisms, including text CAPTCHA, interactive CAPTCHA and non-sense CAPTCHA. For text CAPTCHAs, our results suggest that, in more than 93% of cases, the accuracy rate was higher than 50%. This means the automatic registration bots could usually bypass the text CAPTCHA by calling the service no more than twice. The interactive and non-sense CAPTCHAs (e.g., Google reCAPTCHA and hCAPTCHA) can be bypassed easily (with more than 96% accuracy) but having the service do so costs more money than bypassing a text CAPTCHA.*

4.4 Proxy Service

4.4.1 Data Collection. To gather IP proxy services, we searched for them using a search engine and, because proxy services are legal in almost every country, it was easy to find them. We selected five proxy services and experimented with them to analyze the validity and repetition rate of the IP addresses they provided. Since different IP proxy services have different limits on the number of IP addresses one user can use per day, the number of IP addresses we obtained was different for each provider and depended on their service limits. Every proxy service has different service plans for users; we chose the one-day plan from each one.

When collecting data, we tried to gather as many IP addresses as possible. First, we recorded every IP address and used a GeoIP [32] transform service to get the country and region of these IP addresses. Next, with these IP addresses, we accessed some websites (e.g. Google.com) to confirm their availability. Further, we deployed an echo web application that displayed the client's IP address and the HTTP header fields recorded by the server on a cloud server to check their anonymity.

Table 4: Analysis result of IP proxy services

Platform	IP Count	%Repetition	%Availability	Anonymity
xdaili.cn	1600	15%	98%	Y
moguproxy.com	36000	10%	98%	Y
kuaidaili.com	2370	0	98%	Y
zhimaruanjian.com	1075	0	97%	Y
daxiangdaili.com	34000	32%	92%	Y

4.4.2 The Measurement of Proxy Services. The results shown in Table 4 display the count, repetition rate, availability rate, and anonymity of IP addresses provided by each IP proxy service.

IP Count. The number of IP addresses a proxy service is able to provide is most important for a malicious registrant. In our research, the IP proxy services can be divided into two levels according to the number of IP addresses: small and large. Large proxy services can provide tens of thousands of IP addresses per day, while the small ones can only provide addresses in the low thousands. We believe that a registration bot is unlikely to use more than a few thousand IP addresses in one day, considering how long a registration procedure

takes. Additionally, users will generally not be banned for creating accounts simultaneously, and the websites may not strictly limit the IP addresses like phone numbers because of how many people in a business or household the share one network IP address through a NAT [63].

IP Repetition Rate. To validate the effectiveness of IP proxy services, we also studied the the repetition of IPs. Even though so many IP addresses are provided, some of them are repeated. We recorded every IP address we obtained and noticed that some proxy services might have a repetition rate of about 10%-15%, even 32% in one service. However, even though some IP addresses were repeats, we believe there were still enough to satisfy the needs of malicious registrants.

IP Availability Rate. Most of the IP addresses we obtained are available. All services we chose had an availability rate of more than 90%, and most of them sat at more than 95%. This suggests that these services would work well for accessing real-world websites.

Support Anonymous. By recording the HTTP Headers via our echo web application, we can confirm that all addresses had anonymity and supported anonymous proxies, which means the target websites could not access the real client IPs and would not even know the user was using a proxy IP address.

Answer to RQ3: *The proxy services can provide many available IP addresses for users, from thousands to tens of thousands per day. Most of these proxy IP addresses have high availability and low repetition. With their anonymity, these IP addresses can be used efficiently for malicious registrants to bypass IP restrictions and create new accounts.*

4.5 Automated Registration Test in Real World Websites

To further verify the availability of these anti-human verification services, we tested them in the same way as a malicious registration bot, which means we used them together and tried to develop an automatic registration tool. We used a headless browser with a Python script to keep the workload under control, and incorporated an SMS receiving service, a CAPTCHA recognition service, and an IP proxy service into the application. We then selected five websites (i.e. taobao.com, bilibili.com, zhihu.com, weibo.com, github.com) with different CAPTCHA types from the Alexa Top 100 website [1] to conduct this experiment. Our simple register bots worked well with these services, similar to the automatic registration tools we obtained from the Internet. Furthermore, we also noticed more restrictions than we imagined when running our bots. For example, some websites had strict limitations surrounding phone numbers. Phone numbers from some virtual mobile network operators [62] required more verification than simply receiving an SMS code; an SMS code had to be sent back to them. This meant we needed to choose from the available numbers by filtering through the APIs to bypass these required. Also note that, due to ethical considerations, we have not made the source codes of these tools we used available.

Answer to RQ4: *Our experiments show that creating new accounts on websites can be automated simply by combining these anti-human verification services. Although sometimes more than one attempt was needed to recognize a CAPTCHA or receive an SMS, the anti-human verification tools we assembled worked relatively well on all the websites we tested.*

5 CHARACTERIZING THE IMPACT

With the data collected from the anti-human verification services, we further attempted to characterize the impact of account registration bots by answering the following research questions (RQs):

- RQ5 Which websites are their targets and how many bot accounts are created?
- RQ6 Can we estimate the overall market size of this underground ecosystem in economic terms?

To answer these two RQs, we conducted two experiments with our datasets to reveal the target websites and the economic market size of the underground registration ecosystem.

5.1 Estimating the Number of Registered Accounts

To assess the real impact of the underground account registration ecosystem on real-world websites, we selected 10,000 mobile phone numbers from the SMS receiving services to check whether they had been used to create accounts on websites. Specifically, we use the *forgot password* services provided by websites to check if a phone number had been registered. However, these websites also have security policies for this service, such as CAPTCHA or IP restrictions to combine tools so as to automate this process as we did with the last series of experiments. With the CAPTCHA recognition services and proxy services, we were able to complete our tests on four websites. The results shown in Table 5 suggests that as many as 20%-40% of the phone numbers had been used to create accounts, which suggests an absolutely huge number of bot accounts are active on a range of websites.

Table 5: The number of registered accounts

Websites	Type	Registered accounts	Percentage
baidu.com	Search engine and SNS	2012	**20.12%**
sina.cn	News and SNS	1804	**18.04%**
yylive.cn	Live online	3778	**37.78%**
zhihu.com	Q&A website	2725	**27.25%**

Answer to RQ5: *Our result reveals that all types of websites are the targets of registration bots. Moreover, about 20%-40% mobile phone numbers we collected from SMS receiving services has been used to create new accounts. Considering the large number of mobile phone numbers provided by these services,*

that is no doubt there are a great many malicious accounts on these websites.

5.2 Market Size Estimation

Table 6: The market size of the underground ecosystem

Average price of anti-human services	Lowest Price	Highest Price
SMS Receiving Service	$0.015	$0.15
CAPTCHA recognition Service	$0.0015	$0.30
IP Proxy Service	$0.00074	$0.00296
Total cost per registration	$0.01724	$0.45296
Daily new phone numbers	310,000	
Number of registered accounts per mobile number	2.5	
Daily market size	$13,361.00	$351,044.00
Yearly market size	$4,876,765.00	$128,131,060.00

We next focused on the market size of the malicious registration ecosystem, as shown in Table 6. The market size is calculated based on the unit price of each registration and the number of registrations.

The Average Price of Anti-human Services. First, we investigate the price arrangements for these anti-human services. Considering the costs are diverse, we recorded the lowest and highest prices found for each service. We then assumed that the bots were able to create an account with only one request for each service. Adding these together gave us the cost per registration.

Daily New Phone Number. We used the daily new phone numbers as the base number for the calculations.

Number of Registered Accounts per Phone Number. To evaluate how many accounts have been registered to one phone number, we used a mobile app developed by Tencent called Mobile Manager. This app can detect whether a phone number has been used on any one of 60 websites. One needs to pass SMS verification to use these services, so we obtained 100 phone numbers from an SMS receiving service. As a result, we found that each phone number had been used to register 2.5 accounts on average.

Economic Market Size. Based on these assumptions and data, we can estimate the market size of the underground registration ecosystem by multiplying the cost per registration, daily new phone numbers and the number of created accounts per phone number. The daily market size of the underground registration ecosystem should be between 13K-351K USD and about 4.8M-128.1M per year. We acknowledge that our underlying assumptions are simple, and limited by many factors including a lack of real-world accuracy in CAPTCHA service requests and other inputs, the regional bias of our data set, and so on. However, our guess is that the actual market size may be far more than our estimation.

Answer to RQ6: *We estimate the market size of the malicious registration ecosystem sits at between US $4.8-128.1 million per year. However, because we cannot cover all the data of these services, this estimate is conservative. The actual market size is bound to be much larger.*

6 DISCUSSION

6.1 Implications

Bypass Types. According to the differences in workflows, there are two major types among existing human-verification bypass services: (1) The first type includes SMS receiving services and IP proxy services, which collect a vast amount of phone numbers and IPs to bypass the SMS verification and IP restrictions. (2) The second type includes the CAPTCHA recognition services, which solve the CAPTCHA by machine learning or labour. We believe tracing phone number reuse and/or disrupting bypassing workflows could throttle these two bypass types.

Malicious Situations and Detection. Malicious situations are manipulated by fake accounts, such as coupon collection, fake ranking, fake comments and fake followers. Behaviour analysis may help recognize these malicious situations.

Efforts and Developments Based on Our Findings. According to our findings, the following lists several actions can be used to defend against the registration bots: (1) Based on the workflows of bypassing services (Sections 3.1.2 and 3.2.2), developers can adopt random SMS templates, deploy diverse CAPTCHA or even display them randomly to disrupt the workflows. (2) Based on the reuse mechanism (Sections 3.1.2) and short lifetime (Table. 2) of phone numbers, we can track malicious registrations with the same phone numbers across websites in a short period. (3) Our data collection method (Sections 4.2.1) can help uncover some malicious phone numbers before an attack's registrations.

Accounts after Created. In this research, we aim to evaluate whether different bypassing services can be used to automatically create accounts on real-world websites. Keeping existing fake accounts from behaviour detection is an orthogonal but interesting research topic. In fact, account sellers can keep the automatically created accounts alive from being detected using generated fake activities. For example, in a social network, such as Facebook, the account sellers can use fake avatars and background information to disguise a fake account and then randomly send some posts using this account to make it look like a regular account. Behaviour and content analysis [10, 20, 45, 54] can also help detect these fake accounts.

Implication to Software Engineers. We believe our research is timely and vital to software engineers: (1) For the website developers, our work may help website development with more secure registration strategies to defend against bypassing through the methods we discussed above, such as disrupting the workflows of registration bots. (2) Security engineers can upgrade their security products by tracing the abnormal reuse of phone numbers among different websites. (3) The testing engineers can challenge new software testing procedures using combinations of bypassing services to test the website registration strategy.

6.2 Limitation

There are some limitations to our research. First, due to the lack of discovery methods, it was difficult for us to cover all the anti-human verification services, so there could be more of these services than we know. Second, due to workload involved, we could not develop automatic registration scripts to verify whether all websites could be bypassed. This leaves open the idea that some websites may have unique methods to prevent registration bots. Further, we could not cover all the accounts registered by a phone number, so the estimated market size may be far less than the reality. Lastly, because of the rapid development of both the registration bots and website's security strategies, more and more methods are being invented and used to protect account registration and to bypass it. This battle is a continuous one between a thriving underground account registration ecosystem and besieged website administrators. Our results only provide a snapshot of the current circumstances.

7 RELATED WORK

7.1 CAPTCHA

Research on CAPTCHA has always been a hot spot in community research.

Due to the widespread use of CAPTCHA, since its inception, many attackers have looked to find ways to automatically identify CAPTCHAs. Early on, attackers tried to use different programming methods to attack CAPTCHAs with different schemes [8, 21, 22, 36–38, 47, 69, 70]. These methods placed high requirements on the attacker. In recent years, many studies have tried to solve CAPTCHAs with machine learning and particularly neural networks [9, 27, 41, 42, 73]. These CAPTCHA recognition methods using neural networks pose a greater threat to CAPTCHA problems.

Other researchers are working the opposite side of the fence; they are constantly trying to improve the strength of text CAPTCHAs against the machine learning recognition models. The simplest method is to add background noise to CAPTCHA pictures. Merged characters and character distortion [8] make CAPTCHAs challenging. Further, to counter the great success of neural network methods in CAPTCHA recognition, adversarial perturbations have also been used [40, 44].

The above mentioned efforts on CAPTCHA differ from our research in the following three aspects: (1) These previous approaches only focus on CAPTCHA, whereas we study the whole underground registration ecosystem. (2) Some of these works were done very early even in 2011 or earlier. Moreover, most of them only cover CAPTCHA in the text form, whereas we investigate up-to-date types in recent few years (e.g. interactive and non-sense CAPTCHA). (3) Some of these papers only test the CAPTCHA recognition tools they proposed, whereas we evaluate the mainstream anti-human verification services.

7.2 Detection of Social Bots

In addition to the use of human verification to defend against social bots during registration, many websites are implementing detection methods for abusive accounts and Sybil accounts, which are used in a Sybil attack [64]. These methods analyze account behaviors and the relationships between them. A number of research activities lead to this development and many different methods are used, such as behavior and content analysis [10, 20, 45, 54], machine learning [15, 18, 29, 51], heuristics [48], social graph with features [2, 6, 19, 24, 31, 34, 35, 39, 49, 53, 55, 68, 71, 72, 74, 75, 79, 80], honeypot [30, 48, 72], probabilistic models [16] and action stream models [56]. Leveraging program analysis (e.g., static analysis) is another promising future direction to detect registration tools. Some recent approaches [12, 13, 77] focus on machine-learning-based static analysis to capture potential malicious behaviour and their features can detect malicious families and recognise the resources they used. As for the registration tools running on particular platforms (e.g., Android), some approaches [50, 52, 78] in analyzing and monitoring Android apps can help trace these registration tools on that platform.

7.3 Online SMS Receiving Websites

There is also previous work [4] focusing on studying websites receiving SMS messages/services which can be used to create accounts. However, Our research scale and scope are more significant and larger: (1) The SMS receiving websites in the previous work provide much fewer phone numbers per day than SMS receiving services in our work (hundreds V.S. tens of thousands on average in Table 2). (2) Unlike R1, our study also includes CAPTCHA recognition services, IP proxy services and bypassing workflows by combing these services from the perspective of real-world attackers (Section 4.5).

8 CONCLUSION

With this paper, we undertook the first systematic study of underground account registration bots and anti-human verification services from an attacker's perspective. We explored registration security strategies and registration tools website providers use to protect their sites, and we reviewed the tools malicious registrants use to break through them. We unveiled the ecosystem in which malicious registrants operate and conducted experiments with anti-human verification services, confirming that most services can easily bypass the verification mechanisms website providers put in place. We further estimated the impact and market size of this landscape finding that, conservatively, these malicious activities are netting up to US $128 million per year. Our research results provide a new perspective on registration bots and website registration security. We believe that our research can inspire researchers to focus more effort on website protection and human verification.

ACKNOWLEDGMENT

This work is supported by an ARC Discovery Early Career Researcher Award DE200100016, ARC Discovery Projects DP210101348 and DP200100020, and a scholarship from the China Scholarship Council. Yulei Sui is the corresponding author.

REFERENCES

[1] Alexa. [n. d.]. Alexa top websites. http://www.alexa.com/topsites/category/Top/Computers/Internet/DomainNames.
[2] Lorenzo Alvisi, Allen Clement, Alessandro Epasto, Silvio Lattanzi, and Alessandro Panconesi. 2013. Sok: The evolution of sybil defense via social networks. In *2013 ieee symposium on security and privacy*. IEEE, 382–396.
[3] arkoselabs. [n. d.]. Arkose Labs. https://www.arkoselabs.com/.

[4] Md Hajian Berenjestanaki, Mauro Conti, and Ankit Gangwal. 2019. On the Exploitation of Online SMS Receiving Services to Forge ID Verification. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 1–5.

[5] boost. [n. d.]. Boost.Asio. https://www.boost.org/doc/libs/1_78_0/doc/html/boost_asio.html.

[6] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, Jose Lorenzo, Matei Ripeanu, and Konstantin Beznosov. 2015. Integro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs.. In *NDSS*, Vol. 15. 8–11.

[7] 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)*.

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

[9] Michal Busta, Lukas Neumann, and Jiri Matas. 2017. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Proceedings of the IEEE international conference on computer vision*. 2204–2212.

[10] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 477–488.

[11] Kumar Chellapilla and Patrice Y Simard. 2005. Using machine learning to break visual human interaction proofs (HIPs). In *Advances in neural information processing systems*. 265–272.

[12] Xiao Cheng, Haoyu Wang, Jiayi Hua, Guoai Xu, and Yulei Sui. 2021. Deepwukong: Statically detecting software vulnerabilities using deep graph neural network. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 30, 3 (2021), 1–33.

[13] Xiao Cheng, Guanqin Zhang, Haoyu Wang, and Yulei Sui. 2022. Path-sensitive code embedding via contrastive learning for software vulnerability detection. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*. 519–531.

[14] Cloudflare. [n. d.]. What is rate limiting? | Rate limiting and bots. https://www.cloudflare.com/learning/bots/what-is-rate-limiting/.

[15] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. 2004. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 99–108.

[16] George Danezis and Prateek Mittal. 2009. Sybilinfer: Detecting sybil nodes using social networks.. In *Ndss*. San Diego, CA, 1–15.

[17] datadome. [n. d.]. how-to-detect-prevent-fake-account-creation-websites-apps. https://datadome.co/bot-management-protection/how-to-detect-prevent-fake-account-creation-websites-apps/.

[18] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2018. Analysis of classifiers' robustness to adversarial perturbations. *Machine learning* 107, 3 (2018), 481–508.

[19] David Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. 2016. Who Are You? A Statistical Approach to Measuring User Authenticity.. In *NDSS*, Vol. 16. 21–24.

[20] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. 2010. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 35–47.

[21] Haichang Gao, Mengyun Tang, Yi Liu, Ping Zhang, and Xiyang Liu. 2017. Research on the security of microsoft's two-layer captcha. *IEEE Transactions on Information Forensics and Security* 12, 7 (2017), 1671–1685.

[22] Haichang Gao, Wei Wang, Jiao Qi, Xuqin Wang, Xiyang Liu, and Jeff Yan. 2013. The robustness of hollow CAPTCHAs. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 1075–1086.

[23] geetest. [n. d.]. GeeTest. https://www.geetest.com/.

[24] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. 2014. Sybilbelief: A semi-supervised learning approach for structure-based sybil detection. *IEEE Transactions on Information Forensics and Security* 9, 6 (2014), 976–987.

[25] Google. [n. d.]. Google reCAPTCHA. https://www.google.com/recaptcha/about/.

[26] Google. [n. d.]. Rate-limiting strategies and techniques. https://cloud.google.com/architecture/rate-limiting-strategies-techniques.

[27] Md Imran Hossen, Yazhou Tu, Md Fazle Rabby, Md Nazmul Islam, Hui Cao, and Xiali Hei. 2020. An Object Detection based Solver for {Google's} Image {reCAPTCHA} v2. In *23rd international symposium on research in attacks, intrusions and defenses (RAID 2020)*. 269–284.

[28] Kuo-Feng Hwang, Cian-Cih Huang, and Geeng-Neng You. 2012. A spelling based CAPTCHA system by using click. In *2012 International Symposium on Biometrics and Security Technologies*. IEEE, 1–8.

[29] Xin Jin, Cindy Xide Lin, Jiebo Luo, and Jiawei Han. 2011. Socialspamguard: A data mining-based spam detection system for social media networks. *Proceedings of the VLDB Endowment* 4, 12 (2011), 1458–1461.

[30] Kyumin Lee, James Caverlee, and Steve Webb. 2010. Uncovering social spammers: social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 435–442.

[31] Changchang Liu, Peng Gao, Matthew Wright, and Prateek Mittal. 2015. Exploiting temporal dynamics in sybil defenses. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 805–816.

[32] Inc. MaxMind. [n. d.]. GeoIP. https://www.maxmind.com/en/geoip-demo/.

[33] mediakix. [n. d.]. fake-instagram-influencers-followers-bots-study. https://mediakix.com/blog/fake-instagram-influencers-followers-bots-study/.

[34] Abedelaziz Mohaisen, Nicholas Hopper, and Yongdae Kim. 2011. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *2011 Proceedings IEEE INFOCOM*. IEEE, 1943–1951.

[35] Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. 2010. Measuring the mixing time of social graphs. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 383–389.

[36] 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, 2003. Proceedings.*, Vol. 1. IEEE, I–I.

[37] Gabriel Moy, Nathan Jones, Curt Harkless, and Randall Potter. 2004. Distortion estimation techniques in solving visual CAPTCHAs. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 2. IEEE, II–II.

[38] Yoichi Nakaguro, Matthew N Dailey, Sanparith Marukatat, and Stanislav S Makhanov. 2013. Defeating line-noise CAPTCHAs with multiple quadratic snakes. *computers & security* 37 (2013), 91–110.

[39] Shirin Nilizadeh, François Labrèche, Alireza Sedighian, Ali Zand, José Fernandez, Christopher Kruegel, Gianluca Stringhini, and Giovanni Vigna. 2017. Poised: Spotting twitter spam off the beaten paths. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1159–1174.

[40] Margarita Osadchy, Julio Hernandez-Castro, Stuart Gibson, Orr Dunkelman, and Daniel Pérez-Cabo. 2017. No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2640–2653.

[41] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence* 39, 11 (2016), 2298–2304.

[42] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4168–4176.

[43] Chenghui Shi, Shouling Ji, Qianjun Liu, Changchang Liu, Yuefeng Chen, Yuan He, Z Liu, R Beyah, and T Wang. 2020. Text Captcha Is Dead? A Large Scale Deployment and Empirical Study. In *The 27th ACM Conference on Computer and Communications Security*.

[44] Chenghui Shi, Xiaogang Xu, Shouling Ji, Kai Bu, Jianhai Chen, Raheem Beyah, and Ting Wang. 2021. Adversarial captchas. *IEEE Transactions on Cybernetics* (2021).

[45] Jonghyuk Song, Sangho Lee, and Jong Kim. 2011. Spam filtering in twitter using sender-receiver relationship. In *International workshop on recent advances in intrusion detection*. Springer, 301–317.

[46] Squid. [n. d.]. Squid: Optimising Web Delivery. http://www.squid-cache.org/.

[47] Oleg Starostenko, Claudia Cruz-Perez, Fernando Uceda-Ponga, and Vicente Alarcon-Aquino. 2015. Breaking text-based CAPTCHAs with variable word and character orientation. *Pattern Recognition* 48, 4 (2015), 1101–1112.

[48] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2010. Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*. 1–9.

[49] Gianluca Stringhini, Pierre Mourlanne, Gregoire Jacob, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. 2015. {EVILCOHORT}: Detecting communities of malicious accounts on online services. In *24th USENIX Security Symposium (USENIX Security 15)*. 563–578.

[50] Yulei Sui, Yifei Zhang, Wei Zheng, Manqing Zhang, and Jingling Xue. 2019. Event trace reduction for effective bug replay of Android apps via differential GUI state analysis. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1095–1099.

[51] Enhua Tan, Lei Guo, Songqing Chen, Xiaodong Zhang, and Yihong Zhao. 2013. Unik: Unsupervised social network spam detection. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 479–488.

[52] Yutian Tang, Yulei Sui, Haoyu Wang, Xiapu Luo, Hao Zhou, and Zhou Xu. 2020. All your app links are belong to us: understanding the threats of instant apps based attacks. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 914–926.

[53] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. 2014. Consequences of connectivity: Characterizing account hijacking on twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 489–500.

[54] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. 2013. {Trafficking} Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In *22nd USENIX Security Symposium (USENIX Security 13)*. 195–210.

[55] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. 2010. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review* 40, 4 (2010), 363–374.

[56] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y Zhao. 2013. You are how you click: Clickstream analysis for sybil detection. In *22nd USENIX Security Symposium (USENIX Security 13)*. 241–256.

[57] Haiqin Weng, Binbin Zhao, Shouling Ji, Jianhai Chen, Ting Wang, Qinming He, and Raheem Beyah. 2019. Towards understanding the security of modern image captchas and underground captcha-solving services. *Big Data Mining and Analytics* 2, 2 (2019), 118–144.

[58] wikipedia. [n. d.]. ADSL. https://en.wikipedia.org/wiki/Asymmetric_digital_subscriber_line.

[59] wikipedia. [n. d.]. CAPTCHA. https://en.wikipedia.org/wiki/CAPTCHA.

[60] wikipedia. [n. d.]. Internet_Water_Army. https://en.wikipedia.org/wiki/Internet_Water_Army.

[61] wikipedia. [n. d.]. Man-in-the-middle attack. https://en.wikipedia.org/wiki/Man-in-the-middle_attack.

[62] wikipedia. [n. d.]. Mobile virtual network operator. https://en.wikipedia.org/wiki/Mobile_virtual_network_operator.

[63] wikipedia. [n. d.]. NAT. https://en.wikipedia.org/wiki/Network_address_translation.

[64] wikipedia. [n. d.]. Sybil attack. https://en.wikipedia.org/wiki/Sybil_attack.

[65] wikipedia. [n. d.]. VPS. https://en.wikipedia.org/wiki/Virtual_private_server.

[66] Teng Xu, Gerard Goossen, Huseyin Kerem Cevahir, Sara Khodeir, Yingyezhe Jin, Frank Li, Shawn Shan, Sagar Patel, David Freeman, and Paul Pearce. 2021. Deep entity classification: Abusive account detection for online social networks. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*.

[67] Yi Xu, Gerardo Reynaga, Sonia Chiasson, J-M Frahm, Fabian Monrose, and Paul Van Oorschot. 2012. Security and usability challenges of moving-object captchas: Decoding codewords in motion. In *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*. 49–64.

[68] Jilong Xue, Zhi Yang, Xiaoyong Yang, Xiao Wang, Lijiang Chen, and Yafei Dai. 2013. Votetrust: Leveraging friend invitation graph to defend against social network sybils. In *2013 Proceedings IEEE INFOCOM*. IEEE, 2400–2408.

[69] Jeff Yan and Ahmad Salah El Ahmad. 2007. Breaking visual captchas with naive pattern recognition algorithms. In *Twenty-Third annual computer security applications conference (ACSAC 2007)*. IEEE, 279–291.

[70] Jeff Yan and Ahmad Salah El Ahmad. 2008. A Low-cost Attack on a Microsoft CAPTCHA. In *Proceedings of the 15th ACM conference on Computer and communications security*. 543–554.

[71] Chao Yang, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. 2012. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*. 71–80.

[72] Chao Yang, Robert Chandler Harkreader, and Guofei Gu. 2011. Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 318–337.

[73] Guixin Ye, Zhanyong Tang, Dingyi Fang, Zhanxing Zhu, Yansong Feng, Pengfei Xu, Xiaojiang Chen, and Zheng Wang. 2018. Yet another text captcha solver: A generative adversarial network based approach. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 332–348.

[74] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. 2008. Sybillimit: A near-optimal social network defense against sybil attacks. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 3–17.

[75] Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. 2006. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. 267–278.

[76] Dong Yuan, Yuanli Miao, Neil Zhenqiang Gong, Zheng Yang, Qi Li, Dawn Song, Qian Wang, and Xiao Liang. 2019. Detecting fake accounts in online social networks at the time of registrations. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1423–1438.

[77] Yanxin Zhang, Yulei Sui, Shirui Pan, Zheng Zheng, Baodi Ning, Ivor Tsang, and Wanlei Zhou. 2019. Familial clustering for weakly-labeled android malware using hybrid representation learning. *IEEE Transactions on Information Forensics and Security* 15 (2019), 3401–3414.

[78] Yifei Zhang, Yulei Sui, and Jingling Xue. 2018. Launch-mode-aware context-sensitive activity transition analysis. In *Proceedings of the 40th International Conference on Software Engineering*. 598–608.

[79] Yao Zhao, Yinglian Xie, Fang Yu, Qifa Ke, Yuan Yu, Yan Chen, and Eliot Gillum. 2009. BotGraph: Large Scale Spamming Botnet Detection.. In *NSDI*, Vol. 9. 321–334.

[80] Haizhong Zheng, Minhui Xue, Hao Lu, Shuang Hao, Haojin Zhu, Xiaohui Liang, and Keith W Ross. 2018. Smoke Screener or Straight Shooter: Detecting Elite Sybil Attacks in User-Review Social Networks. In *NDSS*.

[81] Jacqueline Zote. [n. d.]. What are fake influencers and how can you spot them? https://sproutsocial.com/insights/fake-influencers/.