Towards Exploring Cross-Regional and Cross-Platform Differences in Login Throttling

Minjie Cai[1][0009−0002−3452−7292], Xavier de Carné de Carnavalet[2][0000−0003−2664−3963], Siqi Zhang[3][0009−0009−4490−8671], Lianying Zhao[1][0000−0002−6376−4062], and Mengyuan Zhang[3][0000−0001−7457−5198]

[1] Carleton University, Ottawa, Canada
minjiecai@cmail.carleton.ca, lianying.zhao@carleton.ca
[2] The Hong Kong Polytechnic University, Hong Kong SAR, China
xdecarne@polyu.edu.hk
[3] Vrije Universiteit Amsterdam, Amsterdam, Netherlands
{s.zhang4,m.zhang}@vu.nl

Abstract. This study conducts an analysis of login throttling mechanisms on both websites and smartphone apps, focusing particularly on 20 large Chinese and non-Chinese services. Our research uniquely addresses discrepancies in authentication strategies between these services, which have not been extensively covered in existing literature. We manually simulate the behavior of persistent attackers who can circumvent common anti-bot measures, such as solving CAPTCHAs and employing non-suspicious IP addresses. Our findings reveal significant variations in CAPTCHA implementation, password guessing restrictions, and the integration of multiple login throttling mechanisms between app and web interfaces. Notably, Chinese services tend to deploy more complex CAPTCHA systems and additional verification, whereas non-Chinese services are more susceptible to continuous guessing attacks. This paper also proposes a procedure for analyzing and comparing the efficacy of authentication measures in mitigating password-based attacks, contributing to future enhancements to security practices for online services.

Keywords: Authentication · Login Throttling · CAPTCHA · Password Guessing · Online Services · China.

1 Introduction

Passwords are among the most popular and widely used form of identity authentication by online services. Given the critical nature of password-related sensitive information, extensive research has been conducted to address the issue of password guessing [36]. In online guessing attacks, an attacker tries to log into an online service by attempting a list of candidate passwords. Passwords could be simply popular (e.g., *password*, *123456*) or derived from a user's personal information, in which case the success rate increases dramatically. Wang et al. [38] reported that they could reach an alarming 70% success rate with a mere 100 attempts in targeted online guessing attacks. As widely recognized, second-factor

authentication (2FA) is a prevalent and recommended measure aimed at fortifying the security of password-based authentication, e.g., security tokens [35,28,30], emails and text messages (SMS), prompted after *successful* password-based authentication. Given the operational complexity and inconvenience, 2FA is not always active by default.

To prioritize usability, online services also tend to adopt mechanisms to strengthen the login process itself, e.g., login throttling against password guessing, usually presented after *failed* login attempts. Examples include CAPTCHAs [17], temporary blocking, and account lockout. CAPTCHAs help differentiate between human users and bots, while temporary restrictions and account lockout further rate-limit unsuccessful login attempts. Additionally, risk-based authentication (RBA) [41] is widely used to assess the risk level of a login attempt based on user behavior and environmental contexts, such as IP addresses, login modes, and device characteristics [25,41].

Prior work has attempted to characterize how login throttling mechanisms are deployed in the wild. Lu et al. [23] surveyed 182 websites to enumerate the maximum number of login attempts an attacker can perform. They bypassed blocking mechanisms when possible by switching IP addresses. However, their work did not consider more persistent attackers, e.g., solving CAPTCHAs when they encounter any. They also relied on public cloud IP addresses for their login attempts, which online services could consider as inorganic requests. Finally, they used a Selenium-instrumented browser, which could be fingerprinted and flagged as malicious automated logging attempts. Golla et al. [11] manually examined the rate-limiting mechanisms of 12 prominent websites, specifically focusing on CAPTCHAs and account lockout, and conducted separate analyses on each mechanism. However, their testing environment may not resemble that of an attacker who tries to remain stealthy as they attempted logins through the Tor browser, from which traffic is likely treated differently or even as malicious [16]. Overall, prior work misses an important factor: attackers may be stealthy and resourceful to bypass anti-bot detection and use non-suspicious devices/addresses. Last but not least, online services can exist in the form of either a website, or a smartphone app, or oftentimes both.

In our work, we conduct a measurement of the implementations of login throttling mechanisms using a procedure we propose with a special consideration of both website implementations and smartphone apps. We intend to mimic stealthy and persistent attackers by designing a manual procedure as is done by human (organic) users in the day-to-day use of a computing device. By choosing purely manual operations, we ensure our login attempts are ideal from the point of view of large-scale attackers; however, we also accept that the scale of our experiments is inherently limited by our manual testing ability and labor available. Furthermore, our selection of services also reflects significant influences of the services on individual users, e.g., user base, with at least 186 million active users each, covering diverse categories in our analysis. We also place a focus on Chinese services, as those services (and its users) have long been underrepresented in the literature, yet they have proved to possess unique security characteris-

tics compared to non-Chinese ones [42]. Discussions on Chinese passwords [37] have also highlighted significant differences compared to passwords chosen by English-speaking users. Finally, such services potentially affect more than a fifth of the world population. Therefore, we selected a top 10 Chinese and top 10 non-Chinese services.[4]

Through our analysis, we have identified discrepancies between the web version and the app version of the same service, such as variations in CAPTCHA implementations, weaker restrictions on password guessing on certain platforms, and differences in combining various login throttling mechanisms. These behavior discrepancies between websites and apps can have a substantial impact on the overall security posture of the entire service. Furthermore, notable differences exist between Chinese and non-Chinese services, such as more complex CAPTCHA implementations and verification based on phone numbers (SMS) in Chinese services and a higher likelihood of successful login after continual guessing attacks in non-Chinese services.

Contributions. This paper contributes to the security research of password-based authentication in the following aspects:

1. We propose a procedure to analyze the authentication mechanisms adopted by major online services that support both web and mobile app accesses, with regard to their behavior in response to password guessing attempts.
2. We have uncovered, based on our observations, significant discrepancies between websites and their corresponding apps, leading to one platform being less secure (i.e., the weakest link) than the other. When an app is more permissive, testing the corresponding website's login security alone would mislead security analysts. Worse, when platforms are not in sync, attackers could combine the number of permitted login attempts across platforms.
3. We have also identified significant differences in the way Chinese services operate compared to their non-Chinese counterparts in terms of login security, including reliance on phone numbers, CAPTCHAs at the expense of usability, and more stringent lockouts.

2 Terminology and Threat Model

There have been multiple similar and sometimes overloaded terms used in authentication. We first clarify their definitions involved in this paper to facilitate subsequent discussions. *Rate-limiting* is a generic term to refer to mechanisms to limit the number of allowed failed attempts within a specific period of time, which is sometimes used interchangeably with *login throttling*. Such mechanisms include CAPTCHAs and temporarily blocking accesses, e.g., based on IP addresses or cookies. Certain previous research separated account lockout from rate-limiting as was done in the work of Lu et al. [23] and Florêncio et al. [7] because it disables login even by legitimate users with the correct password. In

[4] Data collected on data.ai, formerly known as APP Annie [33], [39], [24], on Dec 2023, https://www.data.ai/en/

this paper, we consider both account lockouts and regular rate-limiting mechanisms for simplicity, referred to as login throttling, in line with other research works such as Golla et al. [11] and Bonneau et al. [5] as they both raise the bar for password guessing and slow down the attacker in one way or another.

In addition, there also exists another form of restriction we call *SMS/Email verification*, which could be triggered at different stages of the authentication process (usually after a correct password), and asks the user to enter a verification code received by either a text message or email. Its key distinction from the aforementioned rate-limiting is that SMS/Email verification per se is an authentication factor, often used in 2FA mechanisms. However, we do not set up any 2FA mechanisms in our experiments as none of the tested services required us to do so.

Although SMS/Email verification is not ideally secure (e.g. [20,31], it normally requires possession of or access to a device/account. Therefore, the appearance of SMS/Email verification will mark the end of the current password guessing attempt for the attacker.

Threat assumptions. In our analysis, we assume the attacker, whose primary goal is to gain unauthorized access to the service (e.g., for information theft, financial fraud, or other malicious purposes), would try to remain as stealthy as possible (not flagged) and is capable of bypassing bot detection mechanisms like solving CAPTCHAs. We do not consider targeted attacks where the attacker has user-specific information. Additionally, this attacker is proficient in cleaning cookies and changing IP address to avoid RBA effectively. They avoid IP addresses that are associated with cloud providers and thus possibly with bad reputations. In our case, we achieve that by utilizing a range of university IP addresses as well as cellular networks that are known to require ID registration. To enhance the likelihood of successful guessing, the attacker exploits differences in security policies and authentication mechanisms across platforms. If blocked on one platform, they shift efforts to alternative platforms.

3 Selection of Services and Passwords

We describe below our selection of 20 services (10 Chinese and 10 non-Chinese) that offer both a website and an app version (hereafter referred to as platforms), and the lists of incorrect passwords we enter on those services based on our testing procedure presented in Section 4.

3.1 Websites and Apps

Chinese services. To choose the Top 10 Chinese apps, we start with the top 50 apps from the Tencent App Store [34]. We discard apps without corresponding websites, those requiring real-name verification, and banking apps. Additionally, we filter out apps lacking password-based login support or relying solely on Single Sign-On (SSO). For companies with multiple services/apps, we only select

their most popular service, e.g., we pick *Baidu* but exclude *BaiduWangpan*. We categorize apps based on the labels sourced from `data.ai` and Google Play (if an app is listed there), and keep only the apps with the highest ranking in each category if there are multiple apps in that category. Our list of top 10 Chinese apps is *QQ*, *Baidu*, *JD*, *Meituan*, *iQIYI*, *Weibo*, *58*, *Ctrip*, *MeituPic*, and *Bilibili*.

Non-Chinese services. Similarly, we download the list of the top 500 non-Chinese apps from Androidrank [2] on Dec. 30, 2023. Then, we sort apps by the estimated number of app installations worldwide and apply the same filtering criteria used for the Chinese apps, excluding those without a website (e.g., *Chrome*, *WhatsApp Messenger*, those without password-based login.[5] We further remove apps subject to regional restrictions that impact us (*TikTok*), as well as online games without user accounts. We also noticed that many apps rely on SSO, such as *Youtube* (*Google* accounts), and *Messenger* (*Facebook* accounts). For apps from the same company, we keep only the most popular ones (*Facebook* over *Instagram*). Our list of top 10 non-Chinese apps is *Google*, *Facebook*, *Microsoft OneDrive*, *Snapchat*, *X* (formerly Twitter), *Spotify*, *LinkedIn*, *Zoom*, *Picsart*, and *Amazon Shopping*.

Account creation. To ensure that the experimental accounts closely resemble real accounts, we register new accounts on the app and set passwords using our personal devices at IP addresses unrelated to the experimental environment.

3.2 Password Lists

To simulate an attacker carrying out a password guessing attack, we created a list of popular passwords that would likely be tested in an untargeted online guessing attack. Considering the different preferences of Chinese and non-Chinese users in password selection [21,12,37], we use separate password lists for Chinese and non-Chinese services.

For non-Chinese services, we pick the top passwords from the HaveIBeenPwned v6 [13] list,[6] and extract passwords of length 6 and above into a first list English-1class6. Top passwords include "123456" and "qwerty". We further extract passwords of length 7 and above, and mangle them using John the Ripper's [27] top 5th mangling rule (capitalize pure alphabetic words and append '1') to create passwords composed of 3 classes and length 8 and above into a second list English-3class8. Top passwords include "Password1" and "Iloveyou1".

For Chinese services, we rely on a password leak from the Chinese service GFAN.com dating from 2013 [22], which is among the most recent Chinese passwords leaks publicly available. We obtained a list of 10.5M cracked account passwords, and extracted two lists Chinese-1class6 and Chinese-3class8 following

[5] We also need to exclude websites bound to their app version by scanning a QR code (without password login), which is the case of WhatsApp.

[6] The list is available as cracked hashes at: `https://gist.github.com/roycewilliams/226886fd01572964e1431ac8afc999ce`. We first un-hexed Hashcat's `$HEX[]` entries and trimmed those with a trailing newline.

the process described above. Top passwords include "123456" and "111111" in the former list, and "Zxcvbnm1", "Qwertyuiop1" in the latter one.

We select passwords from the 1class6 lists if the registration policy of a service lets users register with a 6-character password; otherwise, we select passwords from the 3class8 lists. Note that attackers may leverage their own curated list of top passwords, perhaps even with passwords targeted at specific users [38,29]. By testing more generic popular passwords, the tested services may detect our login attempts as a possible attack more easily, and therefore, our results might show more aggressive defense mechanisms.

4 Methodology

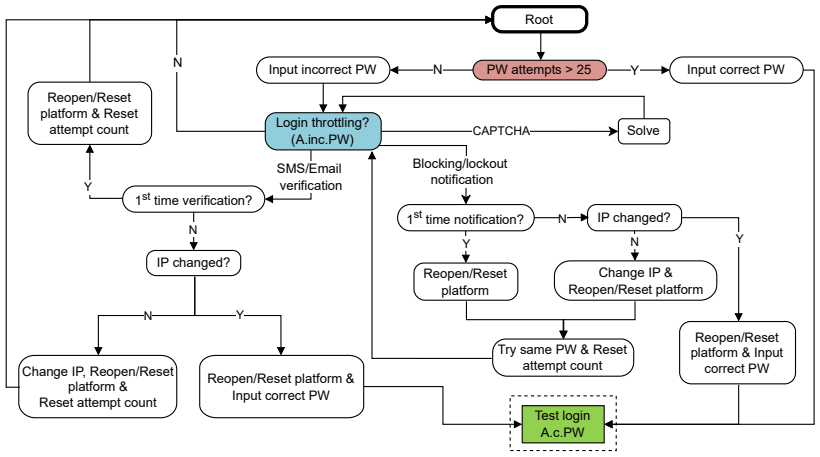
We detail below our procedure for testing the services, including our strategy to deal with various throttling mechanisms. Since these mechanisms may depend on the IP address or device used, we leverage different IP addresses and multiple devices as necessary. We leverage the Chrome browser in incognito mode to browse websites, and several Android smartphones for the counterpart apps. The procedure is illustrated in Fig. 1a and Fig. 1b. Experiment results are discussed in Section 5.

4.1 Testing Strategy

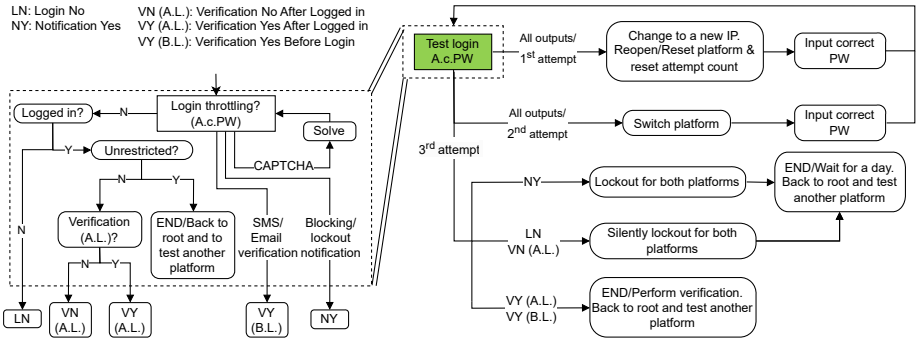
We give an overview of our testing strategy below then discuss specific aspects related to bypassing a) CAPTCHAs, b) blocking and account lockout, and c) SMS/Email verification. As part of our strategy, we also try to login with the correct account password to distinguish between various states and describe this step in more detail.

Overview. Our testing procedure starts by attempting to enter up to *25* incorrect passwords on a service's website, by following the password list compiled in Section 3.2. Before we reach this number though, we may encounter a number of mechanisms to limit/block our attempts. After each incorrect password, we expect three scenarios: CAPTCHAs, blocking/account lockout, or SMS/Email verification to throttle login attempts, as is commonly suggested in relevant guidelines (see NIST SP 800-63B [26]). In such cases, we try to solve CAPTCHAs or bypass other mechanisms by cleaning up the environment, and changing IP addresses. When we either exhaust our quota of incorrect passwords or we cannot bypass restrictions, we attempt to log in with the correct password. Similarly, we try to bypass any restriction by cleaning up the environment and changing IP address, and also switching platform. We then end the test on the website platform. If necessary, we wait for a day until the expiry of lockout periods or complete SMS/Email verification, so that we can then test the app from scratch by following the same testing strategy. We illustrate the complete procedure in Fig. 1, and further explain it below.

CAPTCHAs. Such mechanisms temporarily block the login process until a user solves the challenge. While solving CAPTCHAs is intended to distinguish



(a) General testing strategy starting from the top "Root" node. The module *Test login A.c.PW* (after correct password) depicted with a green box is expanded in the figure below.



(b) Details of the *Test login A.c.PW* module. The left side depicts the inside of this module while the right side describes possible exits, re-entries, and ending conditions.

Fig. 1: High-level flow chart of our login throttling testing strategy

humans from bots, various CAPTCHA-breaking tools have been proposed [32] and could be employed to automate login attempts. Their accuracy could be even higher than solving by humans [32]. Therefore, differentiating bots and humans based on the successful solving of CAPTCHAs is less reliable now. Instead, the role of CAPTCHAs could be primarily considered as a login throttling mechanism. Given the scale of our experiments and the diversity of CAPTCHAs we may encounter, we simply solve them manually. In our experiments, we solved over 300 CAPTCHAs. Attackers could also leverage paid services employing humans to solve them [4]. We can then discover whether additional throttling mechanisms are in place.

If a website/app shows an outright login failure, we simply enter the next incorrect password. Otherwise, we handle other scenarios as described below.

Blocking and account lockout. While incorrect passwords are being attempted, a service may decide to stop serving login requests based on a number of factors. The service could block us thanks to cookie tracking or IP address. Such measures are usually temporary and could last for any duration. When the entire account is locked out, irrespective of which device state and IP address is used for login, even the legitimate user may face challenges to access the account. We learn that blocking or lockout has occurred when the service shows a notification. Our strategy is to try to bypass this verification process by clearing/resetting the browser state/app, and also changing IP address. We resume at the last incorrect password whose attempt was not finalized (we are not explicitly told whether it was a correct or incorrect password). Each time we attempt either of the two techniques, we also reset our attempt counter to zero (*Reset attempt count* in Fig. 1a) to accurately assess password guessing limits when they are not bypassed.

Afterwards, if we are still blocked, we enter the *correct* password through the last IP address that is being throttled, if possible, which helps simulate the case when an attacker stumbles upon the correct password. Services are not expected to treat this login attempt differently and should continue blocking the account. Finally, we change to a fresh IP address once again to enter the correct password on the last platform tested, but after reset, mimicking a legitimate user login. This last step helps distinguish the exact scenario, blocking or lockout. If unsuccessful, the account is locked out.

Sometimes, blocking/lockout occurs silently and the service keeps returning the same "incorrect password" error message, even against the correct password. This situation becomes apparent after we finish our series of 25 passwords and attempt to login with the correct password, as described further below.

SMS/Email verification. Users could be required to enter a code or use a link received via SMS or email after submitting either a correct or incorrect password. We also try to bypass this verification process by clearing/resetting the browser state/app, changing IP address, and we also reset the attempt counter. Afterwards, if we are still required to verify the account, we enter the *correct* password through the last IP address that is being throttled, which helps simulate the case when an attacker stumbles upon the correct password, similar to our strategy when the account is blocked. Finally, we change to a fresh IP address once again to enter the correct password on the last platform tested, similar to how we assess the type of restriction in the case of blocking or account lockout. This last step assesses whether the service imposes the additional verification account-wise or simply based on the history of failed logins by IP address.

No apparent login throttling. If none of the above login throttling mechanisms are observed, we simply try the next incorrect password within our defined limit of 25 passwords since the last throttling bypass. Depending on the exact testing path and service behavior, a total of 75 incorrect passwords may be at-

tempted at a given service; however, in practice this number remains lower than 33. It is in line with a related work [11] in which the authors tested 25 incorrect passwords at non-Chinese website services, and is usually high enough for throttling to occur while remaining tractable by manual effort.

Testing the correct password. Once we exhaust all password attempts or cannot bypass throttling mechanisms, we enter the correct password using the last IP address used. This attempt may fail due to further SMS/Email verification, blocking or account lockout, which could be explicit or not, and that correspond respectively to the three outputs in Fig. 1b: VY (B.L.) (Verification Yes Before Login), NY (Notification Yes), and LN (Login No). We also distinguish whether we can effectively log in, i.e., we enter the account, and whether the service is restricted in some ways or may require further verification, corresponding to the outpus VN (A.L.) (Verification No After Logged in) or VY (A.L.) depending on the need for verification.

In all these cases, we change to a fresh IP address to observe whether a seemingly legitimate login attempt is granted. At this stage, new throttling could occur again. If this second attempt is also unsuccessful, we again change to a fresh IP address and try to log in on the alternative platform. The test conducted on this new platform assesses the consistency between the two platforms. Any inconsistent behavior between the two platforms could potentially present a larger attack surface for that service. Our tests end afterward.

4.2 Technical and Ethical Considerations

We provide below further rationale, technical and ethical elements to support our testing strategy.

Human behavior. In our testing strategy, we ensure the behavior of an organic human user by manually entering passwords into each services. Our typing behavior and other movements on the websites naturally bypass behavioral-based anti-bot measures [15,1]. This effort significantly reduces the effect of anti-bot techniques on the login process therefore mimicking an adversary with significant resources to perform login attempts at scale.

IP addresses. If the throttling mechanisms persist, we attempt to bypass them by switching to a new IP address. In our tests, we utilized Wi-Fi networks provided by our university and hotspots generated with cellular data to obtain IP addresses that differ by at least a /21 subnet from those used in the previous attempts. This step helps us determine whether the implemented mechanisms are based on IP addresses. Note that the IP addresses used in this study are not associated with a bad reputation such as being placed on notable blacklists, nor are they cloud IP addresses.

Considering the significant role of IP addresses in evaluating login risks [41], our testing method utilized the IP addresses from the same country as the registered account. In such cases, the risk score associated with our incorrect password login attempts in Risk-Based Authentication (RBA) system should be medium

or even lower, especially as the IP addresses originate from reputable sources. We also factor in other indicators, such as monitoring login times against typical patterns and ensuring language consistency with registration. Therefore, our testing methodology is designed to avoid triggering negative reputations.

Ethical considerations. This paper simulates online password guessing attacks to understand authentication mechanisms in popular services, identify potential weaknesses and vulnerabilities, and evaluate the security of the throttling mechanisms. We have only conducted a limited number of password guesses (32 unique passwords with few repetitions at most), strictly controlling the number and the frequency. Note that since our list is necessarily composed of very large services, thus our experiment cannot sensibly disrupt the operations of those services nor impact the accessibility of legitimate users.

4.3 Testing Environment

Website-testing platform. We select the Chrome browser as the platform for conducting website experiments considering its world-wide popularity. Since we manually interact with the website services, we do not leverage automated programs that may get fingerprinted, e.g., based on TLS ciphersuites or alert messages [6]. Consequently, our login attempts *feel* organic, which is the best-case scenario for real large-scale attacks. We start each experiment with a fresh incognito session, guaranteeing that Chrome refrains from storing any browsing history, cookies, or website data. By doing so, we ensure that each experiment commences in a pristine and untraceable state (however, see our limitations in Section 7), thereby eliminating any potential influence from prior browsing history on the experimental outcomes.

App-testing platform. We leverage physical Android phones to conduct our experiments. We selected three Google Pixel 3 XL mobile phones with the latest stock Android 12. When testing a service, we ensure that the phone used during the website platform testing (when the app is needed to try the correct password) is different than the phone used to perform all the tests for the app itself. We parallelize the experiments across two students. Note that Google phones are good choices since they are not enforcing geographical restrictions that would prevent the download of Chinese apps. The phones are only used for research experiments and have not been significantly associated with any user activity and no real identity.

Platform reset. When any throttling mechanisms are encountered, we first try to reopen or reset the platform. Specifically, for website testing, we close the browser (incognito) window and reopen it, discarding all traces of browsing history. For app testing, we clear the Android app storage and cache to effectively reset it to its initial state. This step helps us evaluate whether the throttling mechanism relies solely on simple tracking methods, such as session cookies or stored identifiers.

5 Experiment Results and Takeaways

We illustrate the results of our tests on the 20 services in Table 1, highlight significant results and list takeaways below.

5.1 Chinese Services

Account lockout mechanisms. Six of the ten Chinese services implement account lockout mechanisms after a limited number of incorrect password attempts. For example, services like *iQIYI*, *JD*, *Meitu*, and *QQ* lock accounts after 10 incorrect attempts or fewer, often resulting in a temporary lockout period (e.g., 24 hours in the case of *iQIYI* and 3 hours for *Meitu*). These lockout mechanisms remain in place across IP changes and prevent both further incorrect attempts and legitimate logins until the lockout period expires.

SMS verification as a second factor. Eight of the ten services leverage SMS verification, often required after the correct password is entered following multiple incorrect attempts. SMS verification serves as an additional layer of protection even when lockout mechanisms are bypassed, as with *QQ*, *iQIYI* web and *Bilibili* web. Notably, SMS verification is always persistent across IP address changes and platform switching, preventing attackers from gaining access even after successfully entering the correct password.

CAPTCHA enforcement. CAPTCHAs are widely used across Chinese services to mitigate brute-force attacks, with all services leveraging CAPTCHAs on at least one platform. Most services deliver CAPTCHAs after every incorrect attempt, or after only a few attempts. Some services deploy more complex and multi-stage CAPTCHAs, such as those seen on platforms like *Bilibili* and *Ctrip*. In some cases, CAPTCHAs differ between the website and app versions, with the website requiring more frequent or different types of CAPTCHA tasks.

Disparities between app and website platforms. Six of the ten Chinese services exhibit discrepancies in serving CAPTCHAs across platforms. CAPTCHAs could either be delivered more often or sooner on the website than the app platform, or vice versa.

Pre-login notifications and warnings. Several Chinese platforms provide users with pre-login notifications when nearing the incorrect password limit. For example, *iQIYI* displays a countdown notifying users of the remaining allowed attempts before the lockout, helping legitimate users avoid triggering account restrictions. This behavior is not universally seen across non-Chinese services and represents an additional user-friendly feature within Chinese platforms.

Permissive guesses. Five services allow at least 25 incorrect login attempts on at least one platform (the maximum we tested), with either CAPTCHAs as the only throttling method or no throttling. This permissive threshold could allow an attacker to effectively find the correct password with enough attempts. However, in all cases, the attacker would be presented with an SMS verification request before successfully logging in, preventing the attack.

Table 1: Summary of our login throttling analysis results

Services		Guesses	CAPTCHA	Lockout		Login A.c.PW	Verification		General Flow
				Notified?	Bypass		By	Bypass	
Chinese services									
QQ	Web	8	Everytime	Y	**S.P.**		SMS	✗	
	App	6	Everytime	Y	**S.P.**		SMS	✗	
Baidu	Web	**25**	None	-	-		SMS	✗	
	App	**25**	Randomly	-	-		SMS	✗	
JD	Web	10	Everytime	Y	✗		-	-	
	App	10	Everytime	Y	✗		-	-	
Meituan	Web	25?	Everytime	N	✗		SMS	✗	
	App	1	None	Y	-		-	-	
iQIYI	Web	10	New IP + c.PW	Y	**S.P.**		SMS	✗	
	App	10	New IP + c.PW	Y	✗		-	-	
Weibo	Web	**25**	Everytime	-	-		SMS	✗	
	App	**25**	None	-	-		SMS	✗	
58	Web	**25**	Everytime from 4	-	-		SMS	✗	
	App	**25**	Everytime	-	-		SMS	✗	
Bilibili	Web	10+10	Everytime	Y	**C.U., S.P.**		SMS	✗	
	App	**25**	Everytime from 11	-	-		SMS	**S.P.**	
Ctrip	Web	**25**	None	-	-		SMS	✗	
	App	**25**	Everytime from 5	-	-		SMS	✗	
Meitu	Web	7	Everytime until 8	Y	✗		-	-	
	App	7	Randomly	Y	✗		-	-	
Non-Chinese services									
Google	Web	**25**	Randomly	-	-		-	-	
	App	**25**	None	-	-		-	-	
Facebook	Web	25?	None	N	✗		-	-	
	App	12	None	Y	✗		-	-	
Microsoft OneDrive	Web	9	None	Y	**c.PW**		-	-	
	App	9	None	Y	**c.PW**		-	-	
Snapchat	Web	13+19	None	Y	**C.U., c.PW**		App	✗	
	App	**25**	None	-	-		-	-	
X	Web	5+5	None	Y	**New IP**		-	-	
	App	5+5	None	Y	**New IP**		-	-	
Spotify	Web	25?	None	N	**New IP**		-	-	
	App	**25**	None	-	-		-	-	
LinkedIn	Web	20	Everytime from 6	Y	✗		-	-	
	App	15	Everytime from 6	Y	✗		-	-	
Zoom	Web	5	None	Y	✗		-	-	
	App	5	None	Y	✗		-	-	
Picsart	Web	**25**	None	-	-		-	-	
	App	**25**	None	-	-		-	-	
Amazon	Web	6+6+6	None	-	-		Email	**C.U.**	
	App	6+6+6	None	-	-		Email	**C.U.**	

Legend: "Guesses" represents the number of password guesses we could attempt; a sum represents the series of attempts after each bypass of a blocking or SMS/Email verification mechanism. "Lockout – Notified?" means we are blocked or the account is locked out and the service notifies this (Y), the service does it silently (N), or there is no blocking (–). Under "Login A.c.PW" (login after correct password): The correct password is accepted, no throttling occurs, and it leads to full control of the account (only), or any throttling mechanism occurs in the order given. Under sub-columns "Bypass": Mechanism bypassed by switching platform (S.P.), inputting the correct password (c.PW), and/or cleaning up the environment (C.U.), could not be bypassed (✗), or there is no mechanism to bypass (–). Under "Verification – By": Method of verification (SMS, Email, or service app) and always occured before login is completed, or there is no verification (–). "General Flow" represents the sequence of throttling mechanisms (or lack thereof) we encoutered until a successful login or an non-bypassable mechanism. Colored cells represent a discrepancy between the website and app's results for the same service. **Bold** results are noteworthy and discussed in Section 5.3. Icons: →CAPTCHA, →SMS/Email verification, →Account blocked/lockout notification, →No throttling, →Warning notification, →Login successful.

5.2 Non-Chinese Services

Account blocking and lockout. Seven of the 10 non-Chinese services implement blocking or account lockout. *Facebook*, *LinkedIn*, and *Zoom* implement a

lockout that we could not bypass. *Facebook*'s website silently locks the account while the app only allows 12 incorrect attempts before announcing the lockout. Compared to Chinese services, two services (*Spotify* web, *X*) only block by IP address, and can therefore get bypassed by changing IP address.

CAPTCHAs. Only two services serve CAPTCHAs (*Google* and *LinkedIn*), sometimes randomly and not on both platforms. This result comes in direct opposition to how CAPTCHAs are used on Chinese services.

No Email/SMS verification. None of the non-Chinese services rely on SMS as a channel for verification. These services do not always require a phone number to register either. *Snapchat* web requires the user to confirm from the *Snapchat* app. Amazon only suggests a "password assistance" after 6 login failures, which is easily bypassed by cleaning up the environment. There did not seem to be a limit after we tested already 18 incorrect passwords from the same IP address.

Failed and silent blocking. *Microsoft OneDrive* allows a user with the correct password to log in after the account is locked out. In other words, the blocking notification replaces the incorrect password message, but does not block an attacker. The same issue appears with *Snapchat* web. Cleaning up the browser state also helps bypass *Snapchat* web (similar to *Bilibili*). However, it will require verification through the *Snapchat* app after the user enters the correct password. Interestingly, for *Spotify* web, a notification appears that reads "Oops! Something went wrong, please try again or take a look at our help area" after continuously trying incorrect passwords. However, it keeps appearing after entering the correct password, indicating that the account is blocked silently. However, this can be bypassed by changing to a different IP address.

Disparities between app and website platforms. Similar to Chinese services, half of the non-Chinese services exhibit some form of discrepancy across platforms. *Google* does not present CAPTCHAs on the app, *Facebook* web and *Spotify* web silently block login attempts but not on the app. *Snapchat* and *LinkedIn* allow different number of guesses.

Permissive services. Four services allow at least 25 attempts from one platform, with only *Google* web randomly serving CAPTCHAs along the way. Amazon also appears to allow more than 18 incorrect attempts without further throttling mechanisms. Compared to Chinese services, non-Chinese services do not offer a default safety net to catch successful onliny guessing attacks.

5.3 Takeaways

From the results of our study and additional tests, we are able to draw the following takeaways.

Takeaway 1: Higher reliance on SMS verification in Chinese services. Eight out of 10 Chinese services ultimately require authentication by a verification code sent via SMS or email; only a single non-Chinese service does the

same. This measure prevents an attacker with the right password from successfully logging in, and cannot be bypassed easily.

To observe the platforms' default login mechanism and then define the function of SMS/email verification as a baseline, we conduct additional tests using our own regular accounts and enter the correct passwords in private browsing mode on the websites where SMS/Email verification was found.

We discovered that all those tested services need SMS/email verification. Its implementation can vary based on the service's security requirements and the perceived level of risk associated with a login attempt. On the one hand, it can serve as a secondary authentication factor (selectable) or as part of multiple mandatory authentication factors, enhancing the security of user accounts (*Weibo*, *Snapchat*). Take *Weibo* as an example, it indicates that "You have enabled login protection, please verify via SMS.". In this case, SMS verification acts as a secondary authentication factor to prevent an attacker from successfully logging in even with a correct password. On the other hand, it can be implemented as a risk-based authentication measure against an unusual or suspicious login attempt by detecting unfamiliar IP addresses, login devices, or browser metadata (*QQ*, *iQIYI*, *Meituan*, *58*, *Ctrip*). In this case, services tend to indicate that "There are risks in the current login. Please verify your mobile phone number before logging in." or "You are logging in on a new device and need to authenticate". Therefore, SMS/email verification functions as a throttling mechanism because it could appear selectively to reduce risks in certain situations.

Takeaway 2: Throttling mechanisms not in sync between platforms. In our study, each service offers two distinct platforms: a website and a mobile app. User accounts are the same on both platforms. Therefore, an account-wise security evaluation should be implemented to mitigate online password-guessing attacks. However, three services (*QQ*, *iQIYI*, and *Bilibli*) still implement security measures in isolation, focusing on a single platform rather than the account as a whole. On these services, blocking or SMS/Email verification on five of the six corresponding platforms is bypassed by switching to the alternative platform (see Table 1 columns "Lockout–Bypass" and "Verification–Bypass"). Consequently, this can elevate the risk and success rate of targeted attacks.

Takeaway 3: One platform allowing more login attempts. Websites (or apps) may enable attackers to carry out more login attempts before the attacker is blocked or additional authentication measures are triggered. Websites of *QQ* and *LinkedIn* (highlighted with red in Table 1 in the Guesses column) grant more login attempts, making the security of their websites weaker than their apps. Contrarily, the apps of *Bilibili*, *Snapchat*, and *Spotify* are less secure than their websites for the same reason, even allowing up to 25 login attempts before a successful login. Notably, *Meituan* exhibits a unique case where anomalies about mobile phone numbers trigger login blocks after a single incorrect password entry within the app, but the website still allows login attempts. This suggests that the app's security system may be more sensitive to changes in device and network environments, triggering warnings more easily than the website's system.

Takeaway 4: More complex throttling mechanisms in Chinese services. Interestingly, we have observed that all Chinese online services implement login throttling mechanisms, with a preference for CAPTCHA, often in conjunction with other throttling mechanisms. In contrast, for the 10 non-Chinese services, 2 out of them do not employ login throttling mechanisms, and 6 out of them solely set locking mechanisms. However, attackers can more easily bypass those locking mechanisms (see below). In this case, non-Chinese services are more susceptible than Chinese services to online password attacks.

Takeaway 5: Weaker blocking mechanisms in non-Chinese services. Non-Chinese services offer the least effective blocking mechanisms. By simply changing the IP address, a blocked attacker can resume five additional login attempts on *X*. Worse, while Microsoft or Snapchat (website version) appear to block further attempts, an attacker can still try to log in with the correct password from a fresh browser state. This suggests that no blocking measures are effective other than cookie-based, otherwise, even the correct password would be rejected. However, Chinese services do not suffer from these issues.

Takeaway 6: Potential DoS attacks. In our analysis of four Chinese services, *QQ*, *iQIYI*, *JD*, and *Meitu*, and three non-Chinese services, *Facebook*, *LinkedIn*, and *Zoom*, we observed that these platforms (7/20) implement a security measure where accounts are locked out after several consecutive incorrect password attempts. This strategy is designed to enhance the security of user accounts by preventing unauthorized access through brute-force attacks. However, while this approach effectively safeguards user accounts, it also introduces a vulnerability to intentional Denial of Service (DoS) attacks. Specifically, this security measure can be exploited by attackers aiming to disrupt service for legitimate users. By intentionally entering incorrect passwords, attackers can trigger the account lockout mechanism, thereby blocking real users from accessing their own accounts. This is particularly crucial for services like *JD*, a shopping website. Attackers could strategically target legitimate users during peak promotional events, deliberately triggering account lockouts and thus preventing these users from taking advantage of time-sensitive deals. This not only frustrates customers but also poses a tangible threat to the service's revenue, as blocked users are unable to complete their purchases. Similarly, a DoS attack on legitimate users on *Zoom* could create critical consequences during this remote-work era, as it relies heavily on the availability and reliability of digital communication tools.

Takeaway 7: Difference during registration. During account registration, we noticed that almost all Chinese services require a real (e.g., non-virtual, Chinese) phone number and verification code for registration without setting a password, while most of the non-Chinese services prefer to take an email address and allow for setting the password during the registration. Additionally, in most Chinese apps, the login and registration processes are integrated, i.e., upon successful verification of an unregistered mobile phone number, the user is automatically registered and logged in without providing a password or username. Subsequently, users have the option to update their account information at any

time after logging in. We conjecture the main reason for the popularity of SMS verification in China is real-name authentication for each phone number [9,19].

Takeaway 8: More CAPTCHAs in Chinese services. Chinese online services often utilize a wide variety of CAPTCHA mechanisms. Notably, some of these CAPTCHAs are specifically tailored to Chinese users, as they necessitate the understanding of Chinese characters or context to be solved. This language-specific approach adds an extra layer of security but also limits accessibility to users who can understand Chinese. In contrast, we only encountered two types of CAPTCHA in the experiments with non-Chinese services (e.g., orientation selection from *LinkedIn* and distorted text CAPTCHA from *Google*). While stronger CAPTCHA-based mechanisms appear more effective than none, complex and challenging CAPTCHAs can frustrate users, especially those with disabilities, leading to disengagement [40,10]. Implementing CAPTCHAs may result in indirect costs for the website, as it may lose users and potential revenue due to diminished user satisfaction and retention.

Takeaway 9: Different CAPTCHA implementations between website and app. In Chinese services, distorted text CAPTCHAs are often implemented on websites, and slider-based CAPTCHAs are implemented on apps. The frequency and the time at which CAPTCHA appears also differ between the website and app. For example, *Bilibili* CAPTCHAs start to appear after 11 attempts on the app but every time on the website. Among the 20 services we examined, 12 of them deploy CAPTCHAs, with 8 utilizing different CAPTCHA implementations between their website and app, and 4 having CAPTCHAs on only one platform. An attacker may exploit this difference to bypass a platform with weaker CAPTCHA settings, allowing them to gain unauthorized access and conduct malicious activities.

6 Related Work

In an online guessing attack, attackers often gather lists of popular passwords from previous breaches and attempt to log in impersonating the legitimate user. Implementing login throttling mechanisms to mitigate such attacks has emerged as an important strategy for online services. However, research indicates that many online services lack this protective mechanism. Lu et al. [23] proposed a black-box approach to model and validate the implementation of authentication throttling mechanisms for 182 popular websites in the U.S. Their research revealed that 131 out of the 182 websites did not properly implement throttling mechanisms. Among the remaining 51 websites, 28 could block legitimate users with correct passwords. This means overly restrictive throttling strategies may also degrade user experience. Golla et al. [11] investigated differences in throttling mechanisms across 12 non-Chinese website services leveraging the Tor network, often successfully attempting 25 incorrect passwords and logging into half. By contrast, we compare Chinese and non-Chinese services as well as the corresponding Android apps (which is sometimes the main platform for a service, especially in China). We report notable new differences and takeaways.

Furthermore, Risk-based Authentication (RBA) [8] base on risk factors like IP address, device, cookies, login time, and failed attempts [14], assigning different risk levels (i.e., VPN connections are low-risk, unfamiliar devices medium-risk, and different locations high-risk, requiring additional verification) [3], protecting accounts from strong attackers guessing the correct password within a low number of attempts [41]. Wiefling et al. [41] found the IP address most critical in assessing login risk. In our work, in addition to IP addresses, we have also considered different devices to assess whether such changes in the website and app accesses affect login attempts.

7 Limitations

For practical reasons, we excluded services requiring real-name authentication, which might involve more stringent identity verification mechanisms that are worth exploring. Our experiments do not test hundreds or thousands of passwords. Such extensive activities could be blocked differently. We only performed one set of experiments per service. Results might vary due to factors beyond our control and could change with time or external conditions. Our tests were conducted from one country; multi-country failed login attempts (as could be achieved via a botnet) may trigger throttling in different ways. Despite resetting the incognito window and changing IP addresses, services might use browser fingerprinting to detect the same user [18]. Note that we leveraged a different device when entering a correct password on the alternative platform than the device used to perform all the tests.

8 Conclusion

In this paper, we analyze throttling authentication mechanisms employed to mitigate online guessing attacks, focusing on CAPTCHA, blocking/account lockout, and SMS/Email verification. We propose a procedure for exploring such mechanisms in the entire login process and analyze the discrepancy across platforms (i.e., between websites and apps) and across regions (i.e., between Chinese and non-Chinese services). Our results indicate that the same service may set different login throttling mechanisms (especially CAPTCHAs) on different platforms. Additionally, Chinese services tend to set complex CAPTCHAs and SMS verification, while there is a higher chance of bypassing throttling and successfully logging in on non-Chinese services. In summary, our research provides valuable takeaways regarding cross-platform and cross-region implementations of login throttling, highlighting both unexpected and flawed discrepancies as well as interesting variations in the user experience in and out of China. Different strategies should be further evaluated to advance user safety on online services.

References

1. Acien, A., Morales, A., Monaco, J.V., Vera-Rodriguez, R., Fierrez, J.: Typenet: Deep learning keystroke biometrics. IEEE Transactions on Biometrics, Behavior, and Identity Science **4**(1), 57–70 (2022)
2. AndroidRank.com: List of Android most popular Google Play apps, https://www.androidrank.org/android-most-popular-google-play-apps?start=1&sort=4&price=all&category=all, accessed 2024-01-05
3. Awati, R.: TechTarget: risk-based authentication (RBA), https://www.techtarget.com/searchsecurity/definition/risk-based-authentication-RBA
4. AZcaptchas: Auto Captcha Solver Service and Cheap Captcha Bypass Service Provider - AZcaptchas, https://azcaptcha.com/, last accessed 2024-01-08
5. Bonneau, J., Preibusch, S.: The password thicket: Technical and market failures in human authentication on the web. In: Workshop on the Economics of Information Security (2010)
6. Cloudflare: What is rate limiting?, https://www.cloudflare.com/en-gb/learning/bots/what-is-rate-limiting/, accessed 2024-01-05
7. Florêncio, D., Herley, C., van Oorschot, P.C.: An administrator's guide to internet password research. In: Large Installation System Administration Conference (LISA) (2014)
8. Freeman, D., Jain, S., Dürmuth, M., Biggio, B., Giacinto, G.: Who are you? A statistical approach to measuring user authenticity. In: Network and Distributed System Security Symposium. The Internet Society, San Diego, California (2016)
9. Fu, K., Chan, C., Chau, M.: Assessing censorship on microblogs in China: Discriminatory keyword analysis and the real-name registration policy. IEEE Internet Computing **17**(3), 42–50 (2013)
10. Gafni, R., Nagar, I.: Captcha: Impact on user experience of users with learning disabilities. Interdisciplinary Journal of e-Skills and Lifelong Learning **12**, 207–223 (2016)
11. Golla, M., Schnitzler, T., Dürmuth, M., Görtz, H.: "Will any password do?" Exploring rate-limiting on the web. In: Who Are You?! Adventures in Authentication (WAY) (2016)
12. Han, W., Li, Z., Yuan, L., Xu, W.: Regional patterns and vulnerability analysis of Chinese web passwords. IEEE Transactions on Information Forensics and Security **11**(2), 258–272 (2016)
13. Hunt, T.: Pwned passwords, version 6, https://www.troyhunt.com/pwned-passwords-version-6/, last accessed 2024-01-06
14. Hurkała, A., Hurkała, J.: Architecture of context-risk-aware authentication system for web environments. In: The Third International Conference on Informatics Engineering and Information Science (2014)
15. Iliou, C., Kostoulas, T., Tsikrika, T., Katos, V., Vrochidis, S., Kompatsiaris, I.: Detection of advanced web bots by combining web logs with mouse behavioural biometrics. Digital Threats **2**(3) (2021)
16. Khattak, S., Fifield, D., Afroz, S., Javed, M., Sundaresan, S., McCoy, D., Paxson, V., Murdoch, S.J.: Do you see what I see? Differential treatment of anonymous users. In: Annual Network and Distributed System Security Symposium (2016)
17. Kheshaifaty, N., Gutub, A.A.A.: Preventing multiple accessing attacks via efficient integration of CAPTCHA crypto hash functions. International Journal of Computer Science and Network Security **20**(9), 16–28 (2020)

18. Laperdrix, P., Bielova, N., Baudry, B., Avoine, G.: Browser fingerprinting: A survey. ACM Transactions on the Web **14**(2), 8:1–8:33 (2020)
19. Lee, J.A., Liu, C.Y.: Real-name registration rules and the fading digital anonymity in China. Washington International Law Journal **25**, 1 (2016)
20. Lee, K., Kaiser, B., Mayer, J.R., Narayanan, A.: An empirical study of wireless carrier authentication for SIM swaps. In: Symposium on Usable Privacy and Security (SOUPS). pp. 61–79 (2020)
21. Li, Z., Han, W., Xu, W.: A large-scale empirical analysis of Chinese web passwords. In: USENIX Security (2014)
22. Liu, X.: Jifeng Forum was exposed to have leaked the information of 23 million users (translated), Beijing News article (Jan. 6, 2015). `https://www.bjnews.com.cn/detail/155148659914920.html`, last accessed 2024-01-06
23. Lu, B., Zhang, X., Ling, Z., Zhang, Y., Lin, Z.: A measurement study of authentication rate-limiting mechanisms of modern websites. In: Annual Computer Security Applications Conference (ACSAC) (2018)
24. Mao, S., Dewan, S., Ho, Y.I.: Personalized ranking at a mobile app distribution platform. Information Systems Research **34**(3), 811–827 (2023)
25. Markert, P., Schnitzler, T., Golla, M., Dürmuth, M.: "As soon as it's a risk, I want to require MFA": How administrators configure risk-based authentication. In: Symposium on Usable Privacy and Security (SOUPS) (2022)
26. National Institute of Standards and Technology: Digital identity guidelines: Authentication and lifecycle management, nIST Special Publication 800-63B
27. OpenWall.com: John the Ripper password cracker, `https://www.openwall.com/john/`, accessed on 2024-01-05
28. Oracle: Oracle: Java card technology, `https://www.oracle.com/java/java-card/`
29. Pal, B., Daniel, T., Chatterjee, R., Ristenpart, T.: Beyond credential stuffing: Password similarity models using neural networks. In: IEEE Symposium on Security and Privacy (S&P) (2019)
30. Rescorla, E.: The transport layer security (TLS) protocol version 1.3. RFC **8446**, 1–160 (2018)
31. Sami Laine: SMS two-factor authentication – worse than just a good password?, `https://sec.okta.com/articles/2020/05/sms-two-factor-authentication-worse-just-good-password`
32. Searles, A., Nakatsuka, Y., Ozturk, E., Paverd, A., Tsudik, G., Enkoji, A.: An empirical study & evaluation of modern captchas. In: USENIX Security (2023)
33. Shahin, M., Zahedi, M., Khalajzadeh, H., Nasab, A.R.: A study of gender discussions in mobile apps. In: International Conference on Mining Software Repositories (2023)
34. Tencent: Tencent official website, `https://sj.qq.com/`, last accessed 2024-01-06
35. Thanh, D.V., Jørstad, I., Jønvik, T.E., van Thuan, D.: Strong authentication with mobile phone as security token. In: International Conference on Mobile Adhoc and Sensor Systems (MASS) (2009)
36. Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., et al.: Data breaches, phishing, or malware? understanding the risks of stolen credentials. In: ACM Conference on Computer and Communications Security (2017)
37. Wang, D., Wang, P., He, D., Tian, Y.: Birthday, name and bifacial-security: Understanding passwords of Chinese web users. In: USENIX Security (2019)
38. Wang, D., Zhang, Z., Wang, P., Yan, J., Huang, X.: Targeted online password guessing: An underestimated threat. In: ACM Conference on Computer and Communications Security (2016)

39. Wang, X., Markert, C., Sasangohar, F.: Investigating popular mental health mobile application downloads and activity during the COVID-19 pandemic. Human Factors **65**(1), 50–61 (2023)
40. Wentz, B., Pham, D.J., Tressler, K.: Exploring the accessibility of banking and finance systems for blind users. First Monday **22**(3) (2017)
41. Wiefling, S., Iacono, L.L., Dürmuth, M.: Is this really you? An empirical study on risk-based authentication applied in the wild. CoRR **abs/2003.07622** (2020)
42. Xinyi Chen and Yuxuan (Tammy) Zhou: Mobile login methods help chinese users avoid password roadblocks, `https://www.nngroup.com/articles/mobile-login-china/`