A Study of Multi-Factor and Risk-Based Authentication Availability

Anthony Gavazzi
Northeastern University

Ryan Williams
Northeastern University

Engin Kirda
Northeastern University

Long Lu
Northeastern University

Andre King
MIT Lincoln Laboratory

Andy Davis
MIT Lincoln Laboratory

Tim Leek
MIT Lincoln Laboratory

Abstract

Password-based authentication (PBA) remains the most popular form of user authentication on the web despite its long-understood insecurity. Given the deficiencies of PBA, many online services support multi-factor authentication (MFA) and/or risk-based authentication (RBA) to better secure user accounts. The security, usability, and implementations of MFA and RBA have been studied extensively, but attempts to measure their availability among popular web services have lacked breadth. Additionally, no study has analyzed MFA and RBA prevalence together or how the presence of Single-Sign-On (SSO) providers affects the availability of MFA and RBA on the web.

In this paper, we present a study of 208 popular sites in the Tranco top 5K to understand the availability of MFA and RBA on the web, the additional authentication factors that can be used for MFA and RBA, and how logging into sites through more secure SSO providers changes the landscape of user authentication security. We find that only 42.31% of sites support any form of MFA, and only 22.12% of sites block an obvious account hijacking attempt. Though most sites do not offer MFA or RBA, SSO completely changes the picture. If one were to create an account for each site through an SSO provider that offers MFA and/or RBA, whenever available, 80.29% of sites would have access to MFA and 72.60% of sites would stop an obvious account hijacking attempt. However, this proliferation through SSO comes with a privacy trade-off, as nearly all SSO providers that support MFA and RBA are major third-party trackers.

1 Introduction

User authentication is essential for a wide range of online applications and services. From blogging to banking, users expect to be able to secure their personal accounts from public access. The most common mechanism [4,5,29] of user authentication is password-based authentication (PBA), in which a user identifies themselves using a string of characters known only to them. However, the insecurity of passwords has been known for decades. For one thing, users are prone to choosing weak, guessable passwords for their accounts [3,27,33]. Even when users choose nearly unguessable passwords, malware, phishing, and data leaks provide attackers with huge corpuses of account credentials which they can use to access accounts anyway [31]. Further, given the tendency for users to reuse passwords across accounts [7], access to one password via any means can result in multiple accounts being compromised.

Many online services offer multi-factor authentication (MFA) in addition to PBA so that users may further secure their accounts to address PBA deficiencies. Typically, MFA involves unique one-time passcodes (OTPs) generated by a trusted source and sent to a device under the control of the account owner. By disclosing this OTP, the user demonstrates possession of something the account owner *has*. By combining this factor with the original password (something the user *knows*), users successfully authenticate under MFA. MFA implementations still present privacy issues [32], concern over third-party trust [4], and their own security flaws [26], but overall, MFA substantially increases the difficulty of account hijacking.

Despite these potential account security benefits, however, usability issues with MFA lead to pitifully low adoption rates across a number of services [25, 28]. To address this, NIST recommends the use of risk-based authentication (RBA) to secure user accounts [15]. RBA is an adaptive security technique that observes various features available at login and uses those to infer whether the individual logging in is the one who has access to the account. Based on how similar the observed features are to what is typically observed at login, a risk score is calculated, and if the risk score is high enough, additional information may be requested to authenticate successfully. Typically, this information is either personal identifying information or an OTP sent to a device or account under the control of the account owner. For especially high risk-scores, the login attempt may be blocked altogether [25]. Though it still incurs the same security flaws as MFA and is susceptible to false negatives, RBA offers a balance between

security and usability not offered by PBA or MFA.

Although the security and usability of MFA and RBA have been studied extensively [6, 8, 11, 22, 26, 35], any work to characterize the availability of MFA and RBA on the web has been limited to samples of just a handful of sites, and no study has analyzed the impact Single-Sign-On (SSO – an authentication scheme that allows a user to sign into one site using credentials for another) has on their availability. In this paper, we present a study of 208 popular sites in the top 5K of the Tranco list to understand 1) what percentage of sites use RBA, 2) what additional information is requested when a login is deemed suspicious, 3) what percentage of sites offer MFA, 4) what devices can be used by sites for MFA, and 5) how inheriting login defenses from SSO providers changes the landscape of login security on the web.

This paper makes the following contributions:

- We present the most extensive study to date on the availability and characteristics of MFA and RBA on the web.
- We show that although its potential account security benefits are well known, most sites still do not support MFA.
- We show that among sites that support MFA, the majority do not automatically block a highly-suspicious login attempt, but most do alert the user to suspicious activity on their account.
- We show that although most sites do not support MFA or RBA, the vast majority of sites have at least one SSO provider that does, providing users a means to securing accounts on sites that otherwise have poor login security.
- We show that nearly all SSO providers that support MFA and/or RBA are major third-party trackers. Hence, improving login security through SSO providers often comes with a privacy trade-off.

2 Related Work

MFA and RBA Measurements. Few studies or datasets exist which measure the prevalence of MFA or RBA on the web, and those that do are not comprehensive. Quermann et al. [29] manually characterized the user authentication schemes of 48 services across several categories such as popular sites, top universities, banks, and IoT devices. In their analysis, they identified which services offered MFA, what additional authentication factors they supported for MFA, and what SSO providers were available for each service. However, their analysis was primarily concerned with the fine details of a select few services, not on the widespread availability of MFA across the web, so their study covered just nine popular sites and did not test for RBA at all.

2fa.directory [1] is an open-source database of popular services, whether they support MFA, and, if so, the additional authentication factors that they support. As of October 2021, the database lists 1887 services across 37 categories, painting a much broader picture of MFA availability on the web than existing published studies. However, the database lacks entries for a substantial number of popular sites. Of the 208 sites in our study, 114 were not documented in *2fa.directory*. Additionally, the dataset has no information about SSO providers or RBA.

To assess RBA availability, Wiefling et al. [37] developed an automated framework to log into and interact with eight popular sites over the course of two months in order to train any underlying RBA implementations on a specific set of features. By then attempting to log into these accounts from a machine with different features, they identified whether each site used RBA, which features were used to identify a user, and the relative weights assigned to each feature. Their aim, however, was to understand a small set of sites in depth, so their results do not describe the prevalence of RBA across a larger number of popular sites or consider how SSO impacts that prevalence.

In a concurrent study on bypassing RBA via fingerprint spoofing, Lin et al. [24] tested 300 sites in the Alexa top 20K for RBA. However, as their aim was simply to identify as many sites with RBA as possible in order to evaluate their tool, their RBA-detection methodology depended on a "remember this device" option on the login page being the sole trigger for enabling RBA protections. In total, they identified just 16 sites with RBA, most of which were banking and tax-preparation sites, so their results do not report the prevalence of RBA at large on the web.

SSO Relationships. Several studies have analyzed SSO relationships at or above the scale of our study. [30], [39], and [40], for example, analyzed upwards of thousands of relying parties for vulnerabilities in their OAuth implementations, and [13] demonstrated how flaws in the most popular SSO providers can result in thousands of accounts for relying parties being compromised. Though these studies measure SSO relationships similarly to our study, they all focus on authentication flaws and how relying parties inherit these flaws, rather than on the benefits sites may gain from their SSO providers.

3 Methodology

3.1 Challenges to Large-Scale Studies

To effectively study MFA and RBA availability across the web, it is necessary to create new accounts for a large number of services and manually inspect those accounts for MFA and RBA. Manual inspection is necessary because we cannot rely on support documents being available that tell us whether a site supports MFA and RBA. This is especially true in the case of RBA, which does not need supporting documents because a user has no way to control it directly like they

can MFA. Should support documents exist, we also cannot count on them being accurate, as we found several sites with documentation that claimed to support different factors for MFA than were actually available.

As RBA cannot be directly controlled by a user and is unlikely to have supporting documentation, the only way to tell whether a site uses RBA is to observe it as the response to a "suspicious login attempt" – that is, an attempt to log into an account from a machine with a substantially different set of features than are usually seen during login for that account.

In order for us to accurately attempt suspicious logins, we must know the features of the machine used during account creation and during all successful logins to that account. If we do not, we have to guess what features to use in our suspicious login attempt and run the risk of choosing features that are too similar to what is typically observed, resulting in false negatives. Thus, obtaining credentials for accounts created by somebody else on a machine we do not have control of, for example by crowdsourcing account creation for a number of sites or by crawling account-sharing sites like *BugMeNot.com*, is not suitable for our analysis.

Though black-box techniques for automatically creating accounts on sites exist, the state of the art succeeds for only 1.59% of all sites and 11.83% of sites where the signup page is known [10]. The primary reason for this is that signup forms have unpredictable structures and can require specific formatting for each form field. Developing a single automated tool to handle every possible signup form would be practically impossible.

Further, defenses such as CAPTCHAs and web-bot detection tools [16,19,38] hamper automated account creation even when the signup form's structure is perfectly understood. Sites with more sophisticated anti-bot protections would likely be the same sites that use adaptive security measures like RBA, so only analyzing sites that an automated tool can handle would present a significant weakness in our results.

Crowdsourcing our analysis through, for example, Amazon Mechanical Turk is also not ideal for our analysis. If we have each participant perform the entire audit of a site, we run into the problem of ensuring that they performed the audit correctly. If we assign each participant a control site to audit for which we know the correct audit results, then we burden each site in our control set with a potentially substantial number of useless accounts, raising ethical concerns. Providing each participant with a means to control their browser fingerprint to attempt the suspicious login presents additional challenges.

3.2 Site Selection and Account Creation

As performing a large study to characterize the entire web is impractical, we sought to report the state of user authentication security as a typical Internet user would see it. It has long been observed that web traffic corresponds with a Zipf distribution [23], so a substantial amount of Internet traffic occurs on the most popular sites. We therefore focused on a reasonably large number of popular sites that offer account creation and audited them for MFA and RBA.

To obtain a set of sites to audit, we started with the set of sites studied by Innocenti et al. [17], which contains 366 sites of varying popularity that support account creation. By using this dataset, we saved ourselves the trouble of finding sites that support account creation. We selected the 161 sites in the dataset that were in the top 1K of the Tranco list[1] [23] generated on June 21, 2021, and then chose 50 random sites from the dataset between rank 1K and 5K to give us more breadth. Next, for each site in our set, we selected all of its SSO providers and, recursively, the SSO providers for those providers until no new sites were found. Each SSO provider we identified was in the Tranco top 5K, so our analysis still extends only to sites in that range. In total, our set consisted of 235 unique sites.

Most sites have one basic type of user account, but some offer multiple. For example, *indeed.com* allows users to sign up as either a job seeker or an employer. In such cases, we created an account of the type that matched what we perceived to be the most common type for the site. In the case of *indeed.com*, we created an account as a job seeker, as we expected more people to be looking for jobs than to be hiring.

Additionally, some sites allow anybody to sign up for free, but also allow users to pay for premium accounts to access exclusive site features. In cases like this, we created a free account only, as we expected most users would not pay for premium accounts. In all cases where sites support multiple account types, we report the MFA availability, RBA behavior, and SSO providers for our chosen account type only.

Lastly, some sites only allow account creation through other domains, effectively forcing the use of SSO. For example, attempting to create a new account for *slideshare.net* redirected to *linkedin.com*, and all subsequent logins had to use SSO through Linkedin. In the three cases where we saw this, we audited the SSO provider and copied its audit data for the relying party.

3.3 Login Scripting

Though automating account creation is riddled with challenges, programmatically logging into a site given a set of valid credentials is comparably less difficult. For one thing, most login forms do not ask for anything more than a username and password, a structure that is much easier to interpret by an automated tool. More importantly, sites use CAPTCHAs less frequently at login than they do at account creation. Drakonakis et al. found that 13.8% of sites they attempted to sign up for employed CAPTCHAs to protect themselves [10], but Jonker et al. found only 3.9% of sites using CAPTCHAs at the login page [18]. Thus, though creating accounts for the sites in our set was a necessarily manual task,

[1] Available at https://tranco-list.eu/list/42XX.

we hoped to automate logging into the sites in order to make our analysis faster and more repeatable.

To ensure that we could always automatically log into a site successfully (wherever possible in our experimentation), and to allow us to experiment with different browser automation tools, we developed a simple JSON file structure detailing how to log into and out of every site in our set. For each site, we specify a username and password, the URL of the login page, and a series of "actions" that specify the CSS selector of some element on the page and how to interact with that element. Examples of supported actions include typing text into an input element, clicking an element, hovering over an element, and waiting for page navigation.

We then developed a tool that parses these JSON files and executes each action in order, using the HOSIT browser automation framework [36] to complete each action. HOSIT is an extension of Puppeteer [14] that modifies certain library functions to mimic human behavior. Most notably, it clicks at random points in the middle of elements, rather than at the dead center, and adds random delays between pressing and releasing keys and mouse-clicks. These slight modifications are designed to avoid obvious bot-like behavior that some sites may detect and block. Additionally, we always ran HOSIT in a headful mode to further avoid being detected as a bot.

We hoped that by using HOSIT, we could decrease the number of sites we would have to log into manually. We modified HOSIT to handle errors more gracefully and added better support for simultaneously clicking elements and waiting for page loads.

3.4 Black-Box Testing for RBA

One of the primary goals of our study is to measure what portion of sites use some form of RBA, but unlike MFA, RBA is not measurable by reviewing a site's account settings. To identify a site that uses RBA, we have to train the underlying model that identifies a user (referred to as the "RBA model") to associate a particular set of features with our account, then attempt to log into the site from a machine with a completely different set of features, and then observe that the site requests additional authentication factors or responds in some other noticeable way to the login attempt.

Unlike Wiefling et al.'s study [37], our goal is not to determine which features sites look at to decide when to request additional factors or to understand the weight assigned to each feature, but to test whether RBA is used at all. Thus, we do not have to wait for any underlying RBA models to completely stabilize, and need to train the models enough that a login attempt which has been carefully crafted to set off as many alarms as possible will cause a site to request additional authentication factors. The questions therefore become: which features should we explicitly modify in our suspicious login attempt, and how much site interaction is necessary to sufficiently train any underlying RBA models?

Feature	Training	Suspicious
IP Address	Boston, USA	Sofia, Bulgaria
Operating System	Ubuntu 20.04	Windows 10
Browser	Chrome 89.0	Firefox 91.0
Display Resolution	1920 x 1080	1488 x 878

Table 1: Features when training RBA models and during the first suspicious login attempt.

3.4.1 Feature Selection

We based our choice of features on the results of [37]. For all five sites the authors studied that used RBA, training their accounts from one IP address and attempting to log in from an IP address in a different country was enough to cause these five sites to request additional authentication factors. Additionally, the authors found that the user agent string, which comprises the browser version and operating system, and screen resolution were features used by Google, Facebook, and Linkedin, and were highly weighted by Google and Facebook.

Thus, for our analysis, we chose the IP address, operating system, browser, and screen resolution to be our feature set. We trained each account on Chrome 89.0 on an Ubuntu 20.04 virtual machine from a personal IP address in Boston, Massachusetts without the use of a VPN, and attempted our suspicious logins on Firefox 91.0 on Windows 10 from an IP address in Bulgaria using VPN tunnels through NordVPN. We chose Bulgaria due to its far physical distance from our training IP address and due to the presence of sophisticated hackers from eastern Europe [21]. Table 1 shows the values of our chosen features during training and during each suspicious login attempt.

Our choice of operating system and browser was also based on the results of [37]. We based our experimental setup so closely on theirs for two reasons: first, we wanted to use the same environment for HOSIT to ensure that nothing would break, and second, they were able to trigger RBA on five sites using their configuration, so we used it to ensure that we could trigger RBA as well.

Missing from both their study and ours is a formal evaluation of the environment parameters to understand exactly what may factor into enabling and triggering RBA. However, this should be unnecessary. That is, every site *should* detect our suspicious login attempt, as it is designed to be an obvious case of someone logging in using stolen credentials. If some aspect of our experiment setup causes a site that does use RBA not to respond to the suspicious login attempt, then that site's RBA implementation should be considered insecure, as not all user accounts are protected from hijacking. By incorrectly classifying such sites as not using RBA, our results would capture their insecure implementations by reporting only the number of sites with *effective* RBA implementations, which is a more important metric to users given the granularity of our study.

10 sites in our set blocked access from IP addresses outside of the United States. For these sites, rather than attempt the suspicious login from Bulgaria, we would attempt it from San Francisco, California, as it is farthest location in the United States from our original training IP address available to connect to via NordVPN.

3.4.2 Inferring Minimum Necessary Site Interaction

To train RBA models on a specific set of features and to avoid being detected as a bot, [37] performed 20 interactive sessions on each site over the course of two months before attempting their suspicious logins. Automating such interaction for every site in our dataset would be a prohibitively large amount of manual work, but as RBA availability is understudied and the results of [37] are relatively old, there is no available ground truth to help infer the minimum site interaction necessary to enable RBA and train RBA models.

We thus determined a reasonable ground truth and show that regular, lengthy interactions with a site should not be necessary to enable and train RBA – simply logging in from the same machine multiple times is sufficient.

We made two assumptions about RBA implementations that we then evaluated. First, we assumed RBA models are trained at successful logins only, as it is unlikely that sites are performing extensive fingerprinting after this due to the overhead this would incur. Second, we assume that completing account verification should be the only necessary step to enable RBA, as waiting a certain amount of time or for some minimum amount of account interaction to enable it would create a window during which user accounts are more vulnerable to unauthorized access.

To develop our ground truth and assess these assumptions, we randomly selected 50 sites from our set of 235, created two accounts for each site, and completed all necessary account verification steps for both accounts. We attempted to automate logins for all 50 sites using our aforementioned tool, and succeeded for 35 sites.

For one account on each site, we logged in 10 times in a row, using our tool when possible, with no interaction with the site other than logging in and then immediately closing the browser window. 10 logins was chosen based on a separate study by Wiefling et al. [34], in which they found that observing 10 successful logins would be enough to train an RBA model to block 99.92% of attacks. Following the methodology in [37], each login was performed with an empty browser cache and cleared cookies to avoid cookies being used to identify the user instead of browser features.

For the other account on each site, we completed 10 manual sessions lasting 15 to 30 minutes each over the course of a week. In each session, we interacted with each site as we expected a typical user would, for example by watching and liking videos on streaming services, liking and sharing posts on social media sites, and browsing for products on e-commerce sites. Additionally, we created a detailed profile on each account by adding fake personal identifying information. We believe that if some sites enable RBA only for accounts that meet some minimum criteria, these accounts should meet those criteria, as they have profile information and an account history that are worth protecting from unauthorized access.

The day after completing the tenth session, we attempted a suspicious login and recorded the site's response, noting whether additional authentication factors were requested and whether an alert was sent to the email provided at account creation.

For all 50 sites, the response to the suspicious login attempt for the two accounts was exactly the same. When a site requested additional authentication factors for one account, it requested the same factors for the other. When a site allowed the login but sent the user an email alert, it did so for both accounts. If a site did nothing in response to the login attempt, it did nothing for both accounts. Though we can never prove the absence of RBA, our results suggest that logging into an account 10 times in quick succession is as good at enabling and training RBA as interacting with the site multiple times manually over a full week.

3.4.3 Audit Methodology

Thus, in our RBA training, we logged into and then immediately out of each account 10 times before attempting a suspicious login. As with our initial experiment, each login was performed with an empty browser cache and cleared cookies. No attention was paid to the timing of these logins aside from that they never took place the same day an account was created and only after completing account verification, when required. We note that when a site would request authentication factors in response to the suspicious login, we never provided them. Instead, we noted what was requested and then closed the browser window without successfully authenticating.

10 sites in our set required additional authentication factors on every login, even on the same machine that the account was created from. Such sites provide an option at login to "remember this device," which sets a cookie in the browser so that future logins can succeed with just a username and password. [**?**] We consider this a simple RBA implementation, where the only feature analyzed at login is the presence of that cookie, and with the risk score being high whenever the cookie is absent. Thus, for these sites, we did not proceed with further training or even attempt a suspicious login.

We consider any site that responds to the suspicious login attempt as using RBA, and classify these sites as one of two categories. "Blocking" sites request additional authentication features from the user or block the login attempt altogether, for example by displaying a generic error message. "Alerting" sites allow the login to proceed without requesting additional information, but send the user an email alerting them of the login. Sites that do not block the login attempt or alert the

Feature	Value
IP Address	Auckland, New Zealand
Operating System	macOS Big Sur
Browser	Safari 15.0
Display Resolution	1024 x 768

Table 2: Features during the second suspicious login attempt.

user to it simply allow the login to proceed with no apparent action taken, and are classified as not using RBA.

In our experiment, we assumed that a week's worth of activity and 10 full sessions on an account would be enough for it to be considered valid and worthy of adaptive security protections, but we consider the possibility that sites only enable RBA on long-standing accounts or after suspicious activity has been observed before. We thus attempted an additional suspicious login two months after the initial one.

This time, we attempted to log in on Safari 15.0 on macOS Big Sur from an IP address in New Zealand, again using VPN tunnels through NordVPN. We chose New Zealand due to its extreme physical distance from our training location and Bulgaria. For the sites that blocked traffic from Bulgaria, we attempted the suspicious login from Dallas, Texas, as it is the farthest possible location in the United States from the other two United States locations.

Table 2 shows the values of our chosen features during the second suspicious login attempt. We report all numbers based on the behavior observed in this second suspicious login attempt and discuss the different behaviors between the two attempts later.

As a final means to confirming our RBA findings, we used bug bounty programs, security contacts, and customer support contacts to notify each site that did not block our second suspicious login attempt of our findings. In addition to notifying them, we asked them 1) to confirm our findings, 2) why they did not support RBA, and 3) whether they would consider adding it in the future.

After determining which sites had RBA, we randomly selected 50 sites that did not block the suspicious login attempt and investigated whether they would block suspicious behavior post-login. After completing the login from the suspicious machine, we traversed every page in the account settings, noting whether we could see and modify the personal information there. Then, we attempted to change the account password, noting whether we succeeded and, if so, whether an email alert was sent to the email associated with the account.

As an additional case study, for each site that blocked the suspicious login attempt, we logged in from the training machine, copied all of the cookies for that domain to the suspicious machine, and noted whether we were logged in as the user and could access the account settings from the suspicious machine.

3.5 Additional Data Collected

In addition to our RBA measurements, we also took note of which SSO providers could be used to log into the site, whether the site supported MFA, and what devices could be used for MFA. A site's SSO providers are listed in plain sight on the site's signup page and/or login page. As the data is collected manually, we did not need to use any heuristics beyond what we could plainly see on the page, and noted the domains we were redirected to when selecting each SSO provider.

To determine MFA support, we first searched through the entire account settings for the site, typically looking for pages related to account security. If we found the option to enable MFA, we made note of all supported devices. Regardless of whether we found MFA support here, we referenced *2fa.directory* to double check our findings. If *2fa.directory* did not have an entry for the site, then we would search for any self-attestation of MFA support by that site. If at this point we still could not find any evidence that the site supported MFA, then we concluded that it did not support it.

4 Limitations and Scope

A limitation inherent to our study is that we can never prove the absence of RBA on a site, only the existence. Facebook has been known, at least in the past, to enable RBA only for certain accounts based on received friend requests and exchanged messages [37]. Although we expect that this is an uncommon and insecure practice, it demonstrates that our results are strictly a lower bound on RBA availability. Further, some sites may have reserved MFA and/or RBA for specific account types or for users who pay for premium memberships, but our study is unable to report on such possibilities.

Additionally, our study does not try to characterize the underlying RBA models or study the correctness or security of their implementations. Other login defenses, too, such as rate limiting or locking accounts after a certain number of incorrect password guesses may also be present, but are outside the scope of our study.

Lastly, although our work is the largest study of MFA and RBA to date, we are aware that our dataset still comprises a relatively small number of sites. By focusing on popular sites and in light of trends in our results, we believe that we have characterized MFA and RBA availability across popular and freely-accessible sites, but we do not claim that our results reflect user authentication security across the entire Internet. In particular, our results do not capture the availability of MFA or RBA for other services such as online banking, IoT devices, or mobile apps.

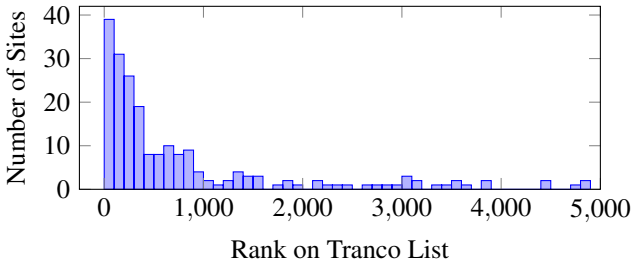

Figure 1: Histogram of Tranco ranks of sites we audited.

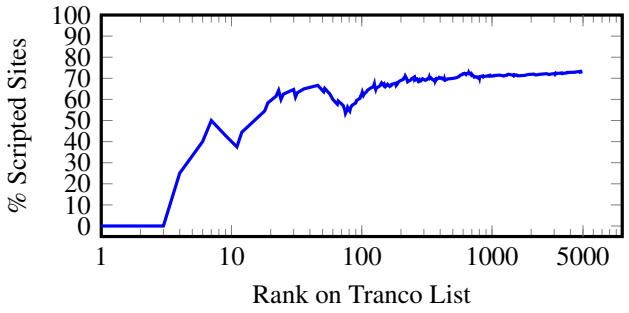

Figure 2: Percentage of sites at or below a given Tranco rank that we could log into using our framework.

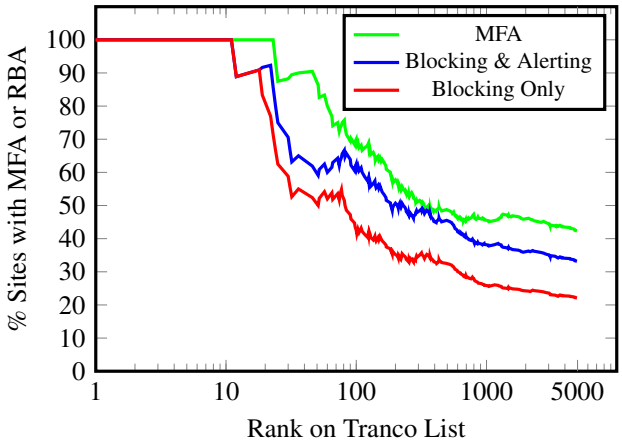

Figure 3: Percentage of sites at or below a given Tranco rank that support MFA or use RBA.

5 Results

Of the 235 sites in our set, we successfully audited 208. We note that every site we audited supported PBA. The 27 remaining sites could not be audited for various reasons such as sites requiring paid subscriptions to sign up, requiring region-specific identification numbers we did not have access to, having removed account creation entirely since Innocenti et al.'s study, and giving errors at the account creation page we could not diagnose. Figure 1 shows a histogram of the Tranco ranks of sites we successfully audited.

Of the 208 sites we audited, we were able to log into 152 sites (73.08%) using our tool, and refer to these as "scripted sites." We refer to the remaining 56 sites (26.92%) as "manual sites." Of these manual sites, 43 could not be scripted because they used CAPTCHAs at the login page, 10 required MFA on every login, and 3 detected and blocked our web driver.

To understand how a site's popularity relates to whether its login can be scripted, Figure 2 plots the percentage of sites at or below a given Tranco rank that could be scripted. Note that the x-axis is in a log scale because of the long tailed distribution of site ranks. For the most part, Tranco rank does not appear to have a major impact on login automatability, with the exception of only the absolute most popular sites blocking bots. We could script logging into 66.67% of sites we audited in the Tranco top 50 and 62.5% of those in the Tranco top 100. By rank 200, the percentage of scripted sites begins to oscillate about 70%, and by rank 500, it never drops below 70% again.

5.1 MFA Results

We find that the majority of sites in our dataset do not support any form of MFA. Only 88 sites (42.3%) support some form of MFA, demonstrating that although the insecurity of PBA has been known for decades, the majority of popular sites still do not allow users to secure their own accounts through MFA.

To understand how a site's popularity factors into MFA availability, we plot the percentage of sites at or below a given Tranco rank that support MFA in Figure 3 (shown in green). The average rank of a site with MFA in our set was 527.75, compared to the average site rank of 811.09 across our entire set. Among the most popular sites, MFA is almost universally supported. 90.48% of sites we audited in the Tranco top 50 and 70% of sites in the Tranco top 100 support MFA. By rank 334, however, the percentage of sites supporting MFA drops below 50% and continues to decrease down to 42.31%.

Table 3 breaks down all the devices available for MFA that we observed and how many sites offered them. We note that the factors listed are exhaustive, and that we did not see any sites supporting biometric or passwordless authentication. We consider proprietary apps (e.g., eBay's mobile app) separate from other authenticator apps (e.g., Google Authenticator) because using a proprietary app to demonstrate possession of an additional authentication factor circumvents the third-party trust issue that is faced when using more common authenticator apps.

By far the most popular devices available for MFA are SMS and third-party authenticator apps, at 61 sites each. Notably, we found 10 sites that support *only* SMS-based MFA, which is known to be less secure than other methods [20]. We also found four sites that support only email-based MFA, which may be insecure as it does not necessarily demonstrate pos-

Authentication Factor	# Sites
SMS	61
Authenticator App	61
Email	16
Security Key	12
Phone Call	11
Proprietary App	5
Proprietary Device	3

Table 3: Authentication factors supported for MFA among sites we audited and how many sites supported them.

session of something the user has, but instead demonstrates knowledge of the user's email account credentials. Using an email-based second authentication factor is actually two-step verification instead of true multi-factor authentication.

We found 18 sites that support only SMS-based and/or email-based MFA ("unsafe" MFA), and just as MFA availability in general declines among less-popular sites, sites with unsafe MFA also tend to be less popular. The most popular site in our set with unsafe MFA is *canva.com*, at rank 116, and the average rank of a site with unsafe MFA is 859.33, compared to 527.75 for sites that support MFA in general.

Different site categories (classified using McAfee's URL Ticketing System [2]) tend to support MFA more often than others. Among categories with at least three sites, Auctions/Classifieds, Games, Social Networking, and Personal Network Storage support MFA the most, with over 75% of sites in those categories supporting MFA. Conversely, none of the sites in the categories General News, Sports, and Fashion/Beauty support MFA.

Some categories tend to be better than others at offering MFA through secure devices as well. For example, 50% of the sites the category Auctions/Classifieds only support unsafe MFA. However, all sites in the categories Personal Network Storage and Finance/Banking support MFA through secure means such as authenticator apps, security keys, and proprietary devices.

Table 9 in Appendix A.1 provides a complete breakdown of how many sites in each category support MFA, support each authentication factor, and support MFA only through less-secure factors.

Table 4 breaks down the availability of MFA for scripted and manual sites. One might predict that sites that protect their login forms against automation tools would be the same security-conscious sites that offer MFA to users, and these results show that indeed, 35.53% of scripted sites support MFA compared to 60.71% of manual sites.

To assess the statistical significance of these results, we conduct a two-proportion z-test to test whether the proportion of scripted sites that support MFA (p_A) is less than the proportion of manual sites that support MFA (p_M). We construct the null hypothesis (H_0) that $p_A = p_M$, the alternative hypothesis

	Has MFA	Does Not Have MFA	Total
Scripted	54	98	152
Manual	34	22	56
Total	88	120	208

Table 4: Number of scripted and manual sites that support or do not support MFA.

Authentication Factor	# Sites
OTP from Email	24
OTP from SMS	12
Click Email Link	4
Phone Number	2
OTP from Email or SMS	2
None (Login Forbidden)	2

Table 5: Authentication factors requested by sites that blocked the suspicious login attempt and how many sites requested them.

(H_a) that $p_A \neq p_M$, and choose a significance level (α) of 0.01. We report a z-score of -3.261 with a corresponding p-value of 0.00111, which is well below our chosen α. Thus, we reject H_0 and conclude that sites that protect their login pages from bots are more likely to support MFA.

5.2 RBA Results

We found that 46 sites (22.1%) blocked the suspicious login attempt. Table 5 shows all the additional authentication factors requested by these sites and how many sites requested them. Email OTPs being by far the most popular authentication factor is likely due to the fact that, for most sites, an email address is the only identifier we provided at account creation, as we did not provide a phone number unless we were required to. 23 sites did not block the suspicious login attempt, but did send an email to the user alerting them of the login. The remaining 139 sites did not respond in any noticeable way to the suspicious login attempt, and are considered not to use RBA.

To understand how a site's popularity factors into RBA use, we plot the percentage of sites at or below a given Tranco rank that use RBA in Figure 3. The percentages of just blocking sites are shown in red and the percentages of blocking and alerting sites together are shown in blue. Both plots follow similar trends. The absolute most popular sites almost universally use RBA, but the percentage drops off very suddenly around rank 20, stays consistent until about rank 80, then decreases steadily as Tranco rank increases.

Certain site categories tend to use RBA more frequently than others. Among categories with at least three sites, Games, Personal Network Storage, Social Networking, and Finance/Banking use RBA the most, with over 40% of sites in those categories blocking a suspicious login and 60% of sites

either blocking or alerting the user to the suspicious login. On the other hand, none of the sites in the categories General News, Education/Reference, and Fashion/Beauty respond in any way to the suspicious login attempt. Additionally, though some sites in the categories Pornography and Sports alerted the user to the suspicious login, none of them blocked it.

These groups of more- and less-secure site categories are nearly identical to those identified in Section 5.1. Indeed, certain categories tend to be better or worse at supporting both MFA and RBA together. Over 50% of sites in the categories Personal Network Storage, Games, Social Networking, and Interactive Web Applications support both MFA and RBA, whereas none of the sites in the categories General News and Fashion/Beauty support either.

These trends likely reflect the fact that certain site categories handle more sensitive and valuable information than others, and both have a higher need to protect this information and often have more resources to implement authentication security mechanisms like MFA and RBA. Table 10 in Appendix A.2 provides a complete breakdown of the RBA behavior we observed for each site category.

Table 6 shows the RBA responses we observed for sites that support MFA and those that do not. Just 38 of the 88 sites that support MFA blocked the suspicious login attempt, and 34 did not respond in any noticeable way to it, showing that even among sites that support MFA, better security measures could still be implemented.

One may expect that all sites that use RBA would also support MFA, and that RBA is used to protect users who do not opt into MFA. However, we found 15 sites that use RBA but do not support MFA. Of those, eight blocked the suspicious login attempt while the remaining seven simply alerted the user to it. Sites that block the login attempt but do not support MFA are especially curious because these sites are capable of generating OTPs as a secondary authentication factor, but do not allow users to use this functionality for every login. Using RBA without offering MFA prevents users from securing their own accounts to their liking, and means that each account is only as secure as the site's RBA implementation, which may be weak to impersonation attacks [6].

Interestingly, among sites that support MFA, 17 sites that request an OTP from the user's email in response to the suspicious login attempt do not support email OTPs for MFA, suggesting that such sites reserve email OTPs for protecting against login attempts that they deem suspicious.

Table 7 breaks down the different RBA responses we observed for our scripted and manual sites. As with MFA, one might predict that sites that protect their login forms against automation tools would be the same sites that would likely use an adaptive security mechanism such as RBA to protect user accounts. Just 26.97% of scripted sites respond to the suspicious login attempt by blocking or alerting compared to 50% of manual sites. To assess the statistical significance of these results, we conduct a two-proportion z-test to test whether the

	Block	Alert	None	Total
Has MFA	38	16	34	88
Does Not Have MFA	8	7	105	120
Total	46	23	139	208

Table 6: Number of sites with various responses to the suspicious login attempt among sites that support and do not support MFA.

	Block	Alert	None	Total
Scripted	21	20	111	152
Manual	25	3	28	56
Total	46	23	139	208

Table 7: Number of sites with various responses to the suspicious login attempt among scripted and manual sites.

proportion of scripted sites that use RBA (p_A) is less than the proportion of manual sites that use RBA (p_M). We construct the null hypothesis (H_0) that $p_A = p_M$ and the alternative hypothesis (H_a) that $p_A \neq p_M$, and choose a significance level (α) of 0.01.

We report a z-score of -3.129 with a corresponding p-value of 0.00176, which is well below our chosen α. Thus, we reject H_0 and conclude that sites that protect their login pages from bots are more likely to use RBA. This result backs up our earlier assumption that scaling up a study such as ours using state-of-the-art techniques for automated account creation and login will substantially under-report the number of sites that use RBA.

Just 52.38% of sites we audited in the Tranco top 50 and 42.50% of sites in the Tranco top 100 blocked the suspicious login attempt, and the percentage drops down to just 22.12% of our entire dataset. 61.90% of sites we audited in the Tranco top 50 and 62.50% of sites in the Tranco top 100 either block the suspicious login attempt or alert the user to it, with the percentage dropping down to 33.17% of our entire dataset.

As noted in Section 3.4, we randomly chose 50 sites that did not block the suspicious login attempt to inspect for any kind of post-login RBA. After logging in from the suspicious machine, all 50 sites allowed us to view all account settings, which included personal information such as the user's first and last name, home address, gender, date of birth, and sexuality. 48 sites allowed us to modify this personal information, with the remaining two sites requiring us to enter an email OTP to continue.

Only 10 sites blocked us from changing the account password by requiring email verification, and among the remaining 40 sites that allowed the password change, just 19 sent an email or SMS alert notifying us that the password was changed. Of the 50 sites in this case study, seven had alerted the user to the suspicious login, and interestingly, all seven of these sites allowed the password change, yet six of them alerted the user to it.

Suspicious Login Attempt	Block	Alert	None
First	37	27	144
Second	46	23	139

Table 8: Number of sites with various responses to the first and second suspicious login attempts.

All of the sites that blocked us from modifying personal information and changing the password also did so when we tried to change them from the machine typically associated with the account, so although this exemplifies a secure practice, it does not indicate that any of the sites use RBA post-login because the behavior is not influenced by implicit browser features. Though just a sample of our overall dataset, our results here indicate that among sites that do not block suspicious login attempts, detecting and blocking suspicious behavior post-login is uncommon.

Also as noted in Section 3.4, for the 46 sites that blocked the suspicious login attempt, we copied each domain's cookies to the suspicious machine to see whether an obvious cookie-stealing attack would be blocked. 10 of these sites blocked us from accessing the account, which, although just 21.74% of blocking sites and a meager 4.81% of our overall dataset, highlights that some sites do protect user accounts post-login.

In this section, we reported the RBA behavior we saw after attempting a second suspicious login two months after the first. Table 8 shows the behavior we saw on the first and second attempts. Although the majority of sites behaved the same between the two attempts, thirteen behaved differently. Twelve sites responded in a more secure manner in the second attempt, and one site that alerted the user to the first suspicious login attempt did not on the second.

We cannot know why these sites reacted differently, but four distinct possibilities occur to us. First, these sites may only enable some defenses for accounts that have existed for a certain amount of time. Second, the first suspicious login attempt may have been detected, but not blocked, and additional security measures were applied to the accounts as a result. Third, the features observed in the second suspicious login attempt may have been seen as more suspicious than those in the first. Fourth, the sites may have changed their RBA behavior for all accounts between the first and second attempts.

We cannot accurately test these possibilities without new accounts for each site and many months to explore how time parameters influence RBA, but we consider the first three possibilities to reflect weaknesses in the sites' RBA implementations, as they would fail to protect new accounts, fail to protect accounts on which suspicious activity has never been detected, or allow attackers to hijack accounts by simply choosing the right browser and VPN location.

To confirm our RBA findings, we reached out to all 162 sites that did not block the suspicious login attempt to ask whether our findings were correct and, if so, why they did not block it. We received responses from 25 sites, none of which contested our findings. Seven sites gave no information aside from confirming the results, but five sites said that they were actively working on adding RBA and nine sites said that they would consider it.

The remaining four sites explained why they would not consider using RBA. Two of them pointed to other defenses such as supporting MFA and locking accounts after a certain number of incorrect password guesses as being sufficient to secure accounts. One site stated that there was little user demand for such defenses, and the last site said that their authentication was handled through a third party, so they had no control over whether RBA could be used or not.

5.3 The Impact of SSO

Though the majority of sites we audited did not support MFA or use RBA, many of these sites have SSO providers that do. In such cases, a user could sign in through the SSO provider and effectively inherit its login security. Then, if one were to obtain the user's credentials and attempt to compromise their account, they would have to go through the SSO provider's login portal and may be thwarted by MFA or RBA. We thus sought to understand how MFA and RBA availability changes when sites inherit them from SSO providers.

For each site we audited that did not support MFA or use RBA, we checked whether one of its direct SSO providers supports it or, recursively, whether one of that provider's SSO providers supports it. For alerting sites, we performed the same recursive analysis to test whether they could inherit an RBA mechanism that blocks the suspicious login attempt, as such a response is more secure.

We find that SSO changes the landscape of user authentication security completely. 167 sites (80.29%) in our set either have MFA or could inherit it through SSO providers. Of the 19 sites that supported only SMS-based and/or email-based MFA, 14 have a direct SSO provider that offers a more secure authentication factor for MFA, leaving just five without access to more secure MFA options. Additionally, 161 sites (77.40%) in our set either use RBA or could inherit it through SSO providers. 151 sites (72.60%) have or inherit an RBA mechanism that blocks the suspicious login attempt, leaving just 10 sites that only alert the user.

Notably, every site that cannot inherit MFA or RBA through SSO has no SSO providers at all. That is, every site with at least one SSO provider has a provider from which they could inherit both MFA and an RBA mechanism that blocks the suspicious login attempt. However, these results should not be understood as saying that using SSO strictly improves user authentication security. Of the 43 unique SSO providers we identified in our set, three did not support MFA and four did not use any kind of RBA. Additionally, we found two sites offering MFA that have at least one SSO provider that does

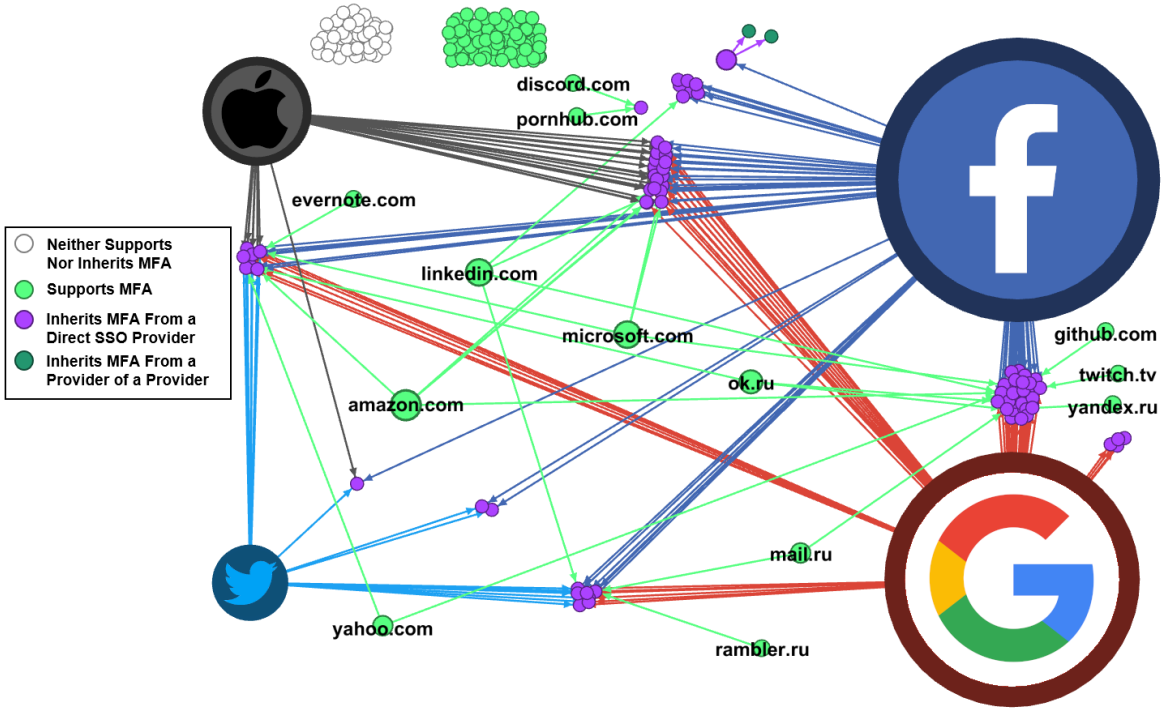

Figure 4: Visualization of MFA inheritance through SSO. Each node is a site we audited. The four most common SSO providers (shown with their own logos) support MFA. Edges point from the SSO provider to the relying party. Only edges where a site gains MFA through a provider are shown.

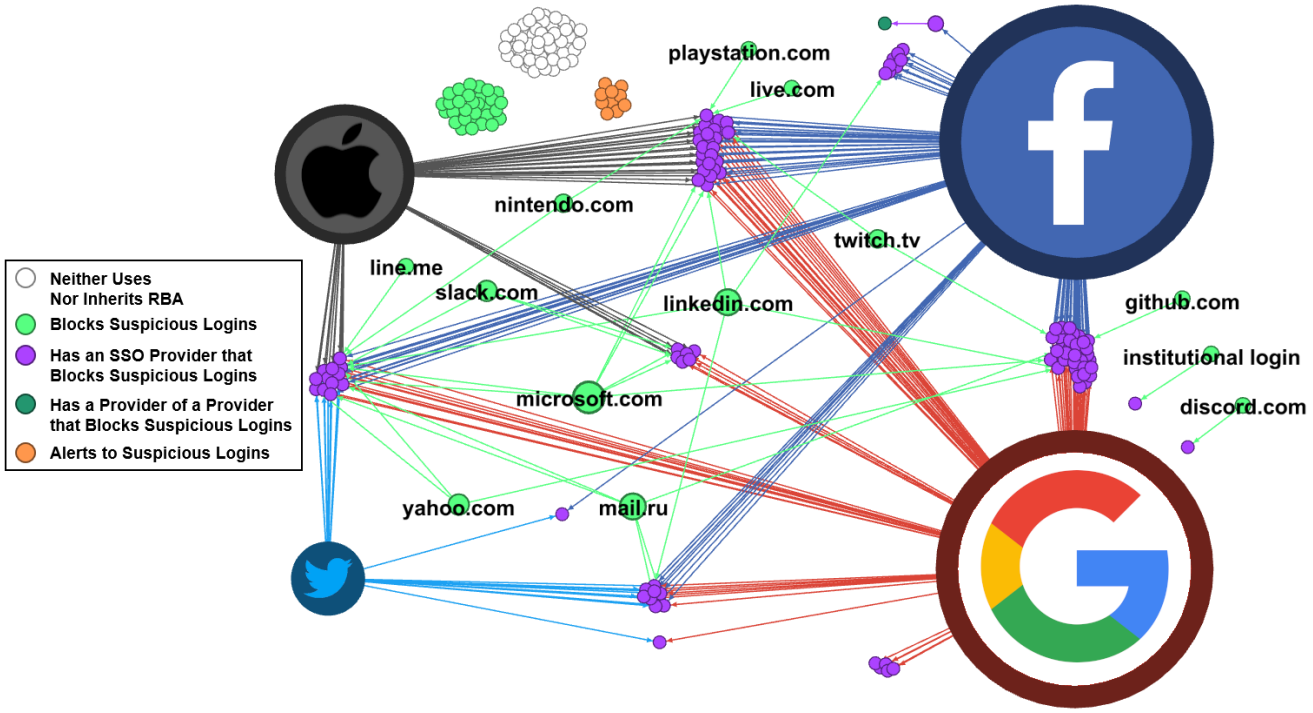

Figure 5: Visualization of RBA inheritance through SSO. Each node is a site we audited. The four most common SSO providers (shown with their own logos) block a suspicious login. Edges point from the SSO provider to the relying party. Only edges where a site gains RBA through a provider are shown.

not, and found three sites with RBA that have at least one SSO provider without it.

The profusion of MFA and RBA through SSO is due largely to the extreme prevalence of Google and Facebook as SSO providers, both of which block suspicious login attempts and offer MFA through secure mechanisms like authenticator apps and hardware security keys. Google and Facebook are providers for 107 and 106 sites in our set, respectively, and of the 131 sites in our set that have at least one SSO provider, only seven have neither Google nor Facebook as a provider.

Figures 4 and 5 depict the impact of SSO graphically. Each node represents a site we audited, edges point from an SSO provider to the relying party, and a node's size is proportional to its out-degree. The four most prevalent SSO providers (indicated by their own logos) and all light green nodes support MFA or blocked the suspicious login attempt. The ubiquity of Facebook and Google as SSO providers is further highlighted here, as although Apple and Twitter are SSO providers for many sites, every relying party to Apple and Twitter is also a relying party to Facebook and/or Google.

The profusion of MFA and RBA through SSO is not entirely thanks to Google and Facebook, however. Many relying parties to Facebook and Google are relying parties to other providers that support MFA and RBA as well, so if Google and Facebook were not available as SSO providers, 132 sites (63.49%) in our set would still either support or inherit MFA, and 109 sites (52.43%) would still either block a suspicious login attempt or inherit an RBA mechanism that would.

It is notable that nearly all of the SSO providers that could confer MFA or RBA to their relying parties are major trackers. Google, Facebook, and Twitter, for example, are the most prevalent third-party trackers on the Internet, according to past measurements [12]. Though using SSO through such providers could improve user authentication security, it comes with a privacy trade-off, as the SSO provider could use the request from the relying party to track a user's browsing activity without the use of third-party tracking cookies.

If none of the domains present on the list of trackers provided by Disconnect.me [9] were used as SSO providers, 118 sites (56.76%) in our set would either support or inherit MFA, and 95 sites (45.70%) would either block a suspicious login attempt or inherit an RBA mechanism that would. Table 11 in Appendix A.3 provides a more thorough breakdown of how MFA and RBA availability changes when only certain SSO providers are allowed.

For MFA, this amounts to a 23.57% decrease in availability over using all domains for SSO, and a 14.14% increase in availability over not using SSO at all. For RBA, this amounts to a 23.96% decrease over using all domains and a 23.56% increase over not using SSO. Though still an improvement over not using SSO at all, this level of MFA and RBA availability is a long way from the superior numbers seen when any domain can be used for SSO.

When tracking domains are omitted, nearly all of the sites that can inherit MFA and RBA inherit it from Apple, which confers MFA to 26 sites and RBA to 44. Of the 30 sites that could inherit MFA from a non-tracking domain, 24 can *only* inherit it from Apple, and of the 49 sites that could inherit blocking-RBA from a non-tracking domain, 39 can only inherit it from Apple.

6 Discussion

Despite the long-understood insecurity of relying on passwords alone for user authentication, only 42.31% of sites we audited supported MFA and only 22.12% of sites blocked a highly suspicious login attempt. Popular sites tend to be the most secure, with 90.48% of sites we audited in the Tranco top 50 supporting MFA and 52.38% blocking a suspicious login attempt. However, we found that both MFA and RBA availability plummet beyond rank 50, and the percentage of sites we saw that supported them continued to decrease as we considered less and less popular sites.

This trend suggests that though our sample of sites is relatively small, our results are likely representative of freely-accessible sites across the Internet. We focused our study on popular sites, covering 40 sites in the Tranco top 100, and observed the results described above. Scaling up our study would primarily entail auditing sites past rank 100, which, given our results and the fact that our set was randomly sampled, we would not expect to be significantly more secure than the other sites we audited. We thus argue that MFA and RBA are uncommon outside of the absolute most popular sites.

Despite this bleak picture, however, the security of popular sites can be leveraged via SSO to protect accounts on less-secure sites. Thanks mostly to the near ubiquity of Google and Facebook as SSO providers, if each account that does not support MFA and/or RBA were to be made through an SSO provider that does, whenever available, 80.29% of sites would have access to MFA and 72.60% of sites would block a suspicious login attempt.

The ones best able to make use of our results are end users, who benefit from the awareness that most sites do not protect their accounts from unauthorized access by someone who knows their account credentials, and that they can leverage SSO to protect their accounts. Site owners may also benefit from knowing that offering SSO through secure providers could be an alternative to adding MFA and RBA themselves.

However, this is not the end of the story. Although we found that SSO is a useful means to securing accounts, nearly all of the providers that can confer MFA and RBA to their relying parties are major trackers. If no trackers were used for SSO, only 56.76% of sites would have access to MFA and only 45.70% would block a suspicious login attempt, with nearly all sites inheriting MFA and RBA from Apple.

The best trade-off between security and privacy in user authentication would seem, therefore, to be to use SSO through Apple whenever possible, but this is far from a satisfactory

solution. Though Apple is the most common SSO provider among non-tracking domains, its lacks the ubiquity needed to confer MFA and RBA to a supermajority of sites. Trusting a single private entity to provide login security to the web at large is also problematic. Thus, at the time of collecting our data, we conclude that MFA and RBA are uncommon on the web, and there is no suitable privacy-preserving means to expanding their availability to the majority of sites.

It would be preferable if sites would support MFA and RBA themselves, but clearly, most are lagging in their adoption. Though our sample size is small, the four sites that explained why they would not support RBA provide some insight into what needs to change. Notably, no sites made any reference to any technical or financial challenges of implementing RBA, so the solution to improving authentication security is likely not to make integrating MFA and RBA easier. Instead, some sites still believe that users are the ones who should protect their own accounts, for example by choosing strong passwords and using MFA, when available. Additionally, we found that some sites simply do not see a strong enough user demand for MFA and RBA to implement them.

The message this sends to the security community is that work still must be done to change prevailing attitudes towards PBA. Site owners need to understand the limitations of relying on passwords and the hesitancy of users to use MFA even when it is available. They must be willing to offer security features like MFA to security-conscious users and implement automatic defenses like RBA to protect the rest.

On the other hand, end users' attitudes toward security must also change. A small, but vocal, group of security-conscious users requesting MFA and RBA may not be enough to convince site owners to support them, but a preponderance of users requesting them could, which would help protect even those remaining users who do not see security as a priority.

7 Ethical Considerations

In this study, we created a substantial number of accounts across 208 sites that we never intended to use as a typical user. To avoid spamming or otherwise abusing each site in our set, we took care to minimize the number of accounts per site.

For each site, we would typically create just two accounts: one for guiding and testing our automation and another for collecting the final audit data. In some cases, we reused existing personal accounts for guiding the automation, and would create just one account for collecting the final data. For each of the 50 sites used in the experiment in Section 3.4.2, we created three accounts total. Considering accounts created during a brief experimentation phase, the upper bound for accounts we created for any site is five, but we reiterate that for the majority of sites, we created just two accounts.

We could have created considerably more accounts to test in detail how factors such as time since account creation and total number of logins factor into enabling and training RBA, but decided against it to avoid spamming the sites.

8 Conclusion

In this paper, we presented a study of 208 sites in the Tranco top 5K to understand how prevalent MFA and RBA are among popular sites. Only 42.31% of sites we audited offered any form of MFA and only 22.12% of sites blocked a highly suspicious login attempt. However, if each account that does not support MFA and/or RBA were to be made through an SSO provider that does, whenever available, 80.29% of sites would have access to MFA and 72.60% of sites would block a suspicious login attempt. This improved security comes with a privacy trade-off, though, as nearly all SSO providers with MFA and RBA are third-party trackers. User authentication security on the web thus has a long way to go, and the primary barrier seems to be users' and site owners' prevailing attitudes and misconceptions towards login security in general.

9 Acknowledgements

We would like to thank the anonymous reviewers for their valuable feedback. This project has partially-been supported by AFOR-PT/MITLL-7000489508 and NSF/CNS-2127200.

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

References

[1] 2fa.directory. https://2fa.directory/.

[2] Mcafee - customer url ticketing system. https://trustedsource.org/en/feedback/url.

[3] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE Symposium on Security and Privacy*, pages 538–552. IEEE, 2012.

[4] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*, pages 553–567. IEEE, 2012.

[5] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. Passwords and the evolution of imperfect authentication. *Communications of the ACM*, 58(7):78–87, 2015.

[6] Michele Campobasso and Luca Allodi. Impersonation-as-a-service: Characterizing the emerging criminal infrastructure for user impersonation at scale. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1665–1680, 2020.

[7] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *NDSS*, volume 14, pages 23–26, 2014.

[8] Sanchari Das, Andrew Dingman, and L Jean Camp. Why johnny doesn't use two factor a two-phase usability study of the fido u2f security key. In *International Conference on Financial Cryptography and Data Security*, pages 160–179. Springer, 2018.

[9] Disconnect.me. Disconnect tracking protection. https://github.com/disconnectme/disconnect-tracking-protection/blob/056d0f19c211b5a6f7a456a36238a12b7198be3b/entities.json, 2022.

[10] Kostas Drakonakis, Sotiris Ioannidis, and Jason Polakis. The cookie hunter: Automated black-box auditing for web authentication and authorization flaws. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1953–1970, 2020.

[11] Jonathan Dutson, Danny Allen, Dennis Eggett, and Kent Seamons. Don't punish all of us: measuring user attitudes about two-factor authentication. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 119–128. IEEE, 2019.

[12] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1388–1401, 2016.

[13] Mohammad Ghasemisharif, Amrutha Ramesh, Stephen Checkoway, Chris Kanich, and Jason Polakis. O single sign-off, where art thou? an empirical analysis of single sign-on account hijacking and session management on the web. In *27th USENIX Security Symposium (USENIX Security 18)*, Baltimore, MD, 2018. USENIX Association.

[14] Google. Puppeteer. https://github.com/googlechrome/puppeteer, 2021.

[15] Paul A Grassi, James L Fenton, Elaine M Newton, Ray Perlner, Andrew Regenscheid, William E Burr, Justin P Richer, Naomi Lefkovitz, Jamie M Danker, Yee-Yin Choong, et al. Digital identity guidelines: Authentication and lifecycle management [includes updates as of 03-02-2020]. 2020.

[16] Christos Iliou, Theodoros Kostoulas, Theodora Tsikrika, Vasilis Katos, Stefanos Vrochidis, and Ioannis Kompatsiaris. Detection of advanced web bots by combining web logs with mouse behavioural biometrics. *Digital Threats: Research and Practice*, 2(3):1–26, 2021.

[17] Tommaso Innocenti, Seyed Ali Mirheidari, Amin Kharraz, Bruno Crispo, and Engin Kirda. You've got (a reset) mail: A security analysis of email-based password reset procedures. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 1–20. Springer, 2021.

[18] Hugo Jonker, Stefan Karsch, Benjamin Krumnow, and Marc Sleegers. Shepherd: a generic approach to automating website login. 2020.

[19] Hugo Jonker, Benjamin Krumnow, and Gabry Vlot. Fingerprint surface-based detection of web bot detectors. In *European Symposium on Research in Computer Security*, pages 586–605. Springer, 2019.

[20] Roger Piqueras Jover. Security analysis of sms as a second factor of authentication: The challenges of multifactor authentication based on sms, including cellular security deficiencies, ss7 exploits, and sim swapping. *Queue*, 18(4):37–60, 2020.

[21] Tom Kellermann. Peter the great versus sun tzu. *Trend Micro Incorporated Opinion Piece*, 2012.

[22] Kat Krol, Eleni Philippou, Emiliano De Cristofaro, and M Angela Sasse. "they brought in the horrible key ring thing!" analysing the usability of two-factor authentication in uk online banking. *arXiv preprint arXiv:1501.04434*, 2015.

[23] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened

against manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, NDSS 2019, February 2019.

[24] Xu Lin, Panagiotis Ilia, Saumya Solanki, and Jason Polakis. Phish in sheep's clothing: Exploring the authentication pitfalls of browser fingerprinting. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1651–1668, 2022.

[25] Grzergor Milka. Anatomy of account takeover. In *Enigma 2018 (Enigma 2018)*, 2018.

[26] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin, and Jean-Pierre Seifert. Sms-based one-time passwords: attacks and defense. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 150–159. Springer, 2013.

[27] Bijeeta Pal, Tal Daniel, Rahul Chatterjee, and Thomas Ristenpart. Beyond credential stuffing: Password similarity models using neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 417–434. IEEE, 2019.

[28] Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, and Sotiris Ioannidis. Two-factor authentication: is the world ready? quantifying 2fa adoption. In *Proceedings of the eighth european workshop on system security*, pages 1–7, 2015.

[29] Nils Quermann, Marian Harbach, and Markus Dürmuth. The state of user authentication in the wild. *Who are you*, 2018.

[30] San-Tsai Sun and Konstantin Beznosov. The devil is in the (implementation) details: an empirical analysis of oauth sso systems. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 378–390, 2012.

[31] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, et al. Data breaches, phishing, or malware? understanding the risks of stolen credentials. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 1421–1434, 2017.

[32] Giridhari Venkatadri, Elena Lucherini, Piotr Sapiezynski, and Alan Mislove. Investigating sources of pii used in facebook's targeted advertising. *Proc. Priv. Enhancing Technol.*, 2019(1):227–244, 2019.

[33] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1242–1254, 2016.

[34] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. What's in score for website users: A data-driven long-term study on risk-based authentication characteristics. *arXiv preprint arXiv:2101.10681*, 2021.

[35] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. More than just good passwords? a study on usability and security perceptions of risk-based authentication. In *Annual Computer Security Applications Conference*, pages 203–218, 2020.

[36] Stephan Wiefling, Nils Gruschka, and Luigi Lo Iacono. Even Turing Should Sometimes Not Be Able To Tell: Mimicking Humanoid Usage Behavior for Exploratory Studies of Online Services. In *24th Nordic Conference on Secure IT Systems (NordSec 2019)*, volume 11875 of *Lecture Notes in Computer Science*, pages 188–203. Springer Nature, November 2019.

[37] Stephan Wiefling, Luigi Lo Iacono, and Markus Dürmuth. Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild. In *34th IFIP TC-11 International Conference on Information Security and Privacy Protection (IFIP SEC 2019)*, volume 562 of *IFIP Advances in Information and Communication Technology*, pages 134–148. Springer International Publishing, June 2019.

[38] Haitao Xu, Zhao Li, Chen Chu, Yuanmi Chen, Yifan Yang, Haifeng Lu, Haining Wang, and Angelos Stavrou. Detecting and characterizing web bot traffic in a large e-commerce marketplace. In *European Symposium on Research in Computer Security*, pages 143–163. Springer, 2018.

[39] Ronghai Yang, Guanchen Li, Wing Cheong Lau, Kehuan Zhang, and Pili Hu. Model-based security testing: An empirical study on oauth 2.0 implementations. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 651–662, 2016.

[40] Yuchen Zhou and David Evans. Ssoscan: Automated testing of web applications for single sign-on vulnerabilities. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 495–510, 2014.

A Appendix

A.1 MFA Factors by Category

Category	# Sites	# MFA	# Unsafe	SMS	3rd Party App	Email	Security Key	Phone Call	1st Party App	1st Party Device
Business	24	12	2	8	8	3	0	2	1	1
Software/ Hardware	20	12	1	6	8	3	0	2	1	2
Blogs/Wiki	19	7	1	4	6	1	1	0	0	0
Internet Services	19	8	2	7	6	1	3	1	0	0
General News	17	0	0	0	0	0	0	0	0	0
Online Shopping	15	4	1	3	3	1	0	2	0	0
Streaming Media	11	3	0	2	3	1	1	1	0	0
Media Sharing	11	1	0	1	1	0	1	1	0	0
Interactive Web Applications	11	7	1	6	5	0	1	2	0	1
Education/ Reference	10	2	0	0	2	1	0	0	0	0
Portal Sites	10	7	2	4	2	1	2	0	2	0
Games	10	9	2	3	6	4	0	0	1	0
Entertainment	10	2	0	0	2	1	0	0	0	0
Pornography	9	5	1	3	4	0	1	1	0	0
Personal Network Storage	8	6	0	6	5	0	1	2	0	1
Social Networking	7	6	1	6	5	0	2	0	0	0
Sports	6	0	0	0	0	0	0	0	0	0
Auctions/ Classifieds	5	4	2	4	1	1	0	0	1	0
Finance/Banking	4	2	0	2	2	0	0	0	0	0
Fashion/Beauty	4	0	0	0	0	0	0	0	0	0
Forum/Bulletin Boards	3	1	0	0	1	0	0	0	0	0
Technical/Business Forums	3	1	0	1	1	0	0	0	0	0
Search Engines	2	2	0	2	2	1	1	2	0	0
Professional Networking	2	1	0	1	1	0	0	0	0	0
Web Meetings	2	2	0	1	2	0	0	0	0	0
Personal Pages	2	1	0	1	1	0	0	0	0	0
Job Search	2	1	0	1	1	0	0	0	0	0
Travel	2	1	1	1	0	1	0	0	0	0
Public Information	2	1	1	0	0	1	0	0	0	0
Art/Culture/ Heritage	2	0	0	0	0	0	0	0	0	0
Stock Trading	2	1	0	1	1	0	0	0	0	0
Potential Illegal Software	2	0	0	0	0	0	0	0	0	0

Table 9: Categories of sites we audited and how many sites in each category support various factors for MFA. Only categories with two or more sites are shown. The "# Unsafe" column counts the number of sites that only support MFA through SMS or email.

A.2 RBA Behavior by Category

Category	# Sites	# RBA	Email OTP	SMS OTP	Email Link	Login Forbidden	Phone Number	Email or SMS OTP	Email Alert
Business	24	6	1	2	1	0	0	0	2
Software/ Hardware	20	6	2	1	0	0	0	0	3
Blogs/Wiki	19	5	1	0	0	0	1	0	3
Internet Services	19	8	4	0	1	0	0	1	2
General News	17	0	0	0	0	0	0	0	0
Online Shopping	15	4	3	0	0	0	0	0	1
Streaming Media	11	4	1	1	0	0	0	0	2
Media Sharing	11	3	0	1	0	1	0	0	1
Interactive Web Applications	11	6	2	1	0	0	0	0	3
Education/ Reference	10	0	0	0	0	0	0	0	0
Portal Sites	10	4	0	1	0	0	1	1	1
Games	10	6	5	0	0	0	0	0	1
Entertainment	10	3	0	1	0	0	0	0	2
Pornography	9	2	0	0	0	0	0	0	2
Personal Network Storage	8	6	3	1	1	0	0	0	1
Social Networking	7	5	1	0	0	1	1	0	2
Sports	6	1	0	0	0	0	0	0	1
Auctions/ Classifieds	5	2	0	1	0	0	0	0	1
Finance/Banking	4	3	0	2	0	0	0	0	1
Fashion/Beauty	4	0	0	0	0	0	0	0	0
Forum/Bulletin Boards	3	1	0	0	1	0	0	0	0
Technical/Business Forums	3	1	1	0	0	0	0	0	0
Search Engines	2	2	1	1	0	0	0	0	0
Professional Networking	2	1	1	0	0	0	0	0	0
Web Meetings	2	1	1	0	0	0	0	0	0
Personal Pages	2	1	0	0	0	0	0	0	1
Job Search	2	1	1	0	0	0	0	0	0
Travel	2	0	0	0	0	0	0	0	0
Public Information	2	2	1	0	0	0	0	0	1
Art/Culture/ Heritage	2	0	0	0	0	0	0	0	0
Stock Trading	2	1	0	1	0	0	0	0	0
Potential Illegal Software	2	0	0	0	0	0	0	0	0

Table 10: Categories of sites we audited and how many sites in each category exhibited certain RBA behavior. Only categories with two or more sites are shown.

A.3 MFA and RBA Availability with Restricted SSO Providers

Restriction on SSO Providers	# MFA	# RBA	# Block	# Alert
No Sites Allowed	88 (42.33%)	69 (33.19%)	46 (22.13%)	23 (11.06%)
Any Sites Allowed	167 (80.33%)	161 (77.44%)	151 (72.63%)	10 (4.81%)
Only Google	153 (73.59%)	149 (71.67%)	138 (66.38%)	11 (5.29%)
Only Facebook	159 (76.48%)	149 (71.67%)	136 (65.42%)	13 (6.25%)
Only Apple	114 (54.83%)	105 (50.50%)	90 (43.29%)	15 (7.21%)
Only Twitter	105 (50.50%)	87 (41.85%)	67 (32.23%)	20 (9.62%)
Only Apple or Twitter	123 (59.16%)	113 (54.35%)	100 (48.10%)	13 (6.25%)
No Google	163 (78.40%)	156 (75.04%)	146 (70.23%)	10 (4.81%)
No Facebook	159 (76.48%)	154 (74.07%)	143 (68.78%)	11 (5.29%)
No Google or Facebook	132 (63.49%)	121 (58.20%)	109 (52.43%)	12 (5.77%)
No Google, Facebook, or Apple	119 (57.24%)	99 (47.62%)	83 (39.92%)	16 (7.70%)
No Google, Facebook, or Twitter	125 (60.12%)	114 (54.83%)	102 (49.06%)	12 (5.77%)
No Google, Facebook, Apple, or Twitter	108 (51.95%)	87 (41.85%)	71 (34.15%)	16 (7.70%)
No Tracking Domains	118 (56.76%)	107 (51.47%)	95 (45.70%)	12 (5.77%)
No Tracking Domains or Apple	94 (45.21%)	70 (33.67%)	53 (25.49%)	17 (8.18%)

Table 11: Number of sites that have or could inherit MFA and various RBA behaviors through SSO when restrictions are placed on which sites can be used for SSO.