Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service

STEPHAN WIEFLING, H-BRS University of Applied Sciences, Germany and Ruhr University Bochum, Germany
PAUL RENÉ JØRGENSEN and SIGURD THUNEM, Telenor Digital, Norway
LUIGI LO IACONO, H-BRS University of Applied Sciences, Germany

Risk-based authentication (RBA) aims to protect users against attacks involving stolen passwords. RBA monitors features during login, and requests re-authentication when feature values widely differ from those previously observed. It is recommended by various national security organizations, and users perceive it more usable than and equally secure to equivalent two-factor authentication. Despite that, RBA is still used by very few online services. Reasons for this include a lack of validated open resources on RBA properties, implementation, and configuration. This effectively hinders the RBA research, development, and adoption progress.

To close this gap, we provide the first long-term RBA analysis on a real-world large-scale online service. We collected feature data of 3.3 million users and 31.3 million login attempts over more than 1 year. Based on the data, we provide (i) studies on RBA's real-world characteristics plus its configurations and enhancements to balance usability, security, and privacy; (ii) a machine learning–based RBA parameter optimization method to support administrators finding an optimal configuration for their own use case scenario; (iii) an evaluation of the round-trip time feature's potential to replace the IP address for enhanced user privacy; and (iv) a synthesized RBA dataset to reproduce this research and to foster future RBA research. Our results provide insights on selecting an optimized RBA configuration so that users profit from RBA after just a few logins. The open dataset enables researchers to study, test, and improve RBA for widespread deployment in the wild.

CCS Concepts: • **Security and privacy** → **Authentication**; **Access control**; **Web application security**; **Privacy protections**; **Usability in security and privacy**;

Additional Key Words and Phrases: Risk-based authentication, large-scale online services, big data analysis

1 INTRODUCTION

Despite decades of efforts to replace passwords with more secure alternatives [9], they are still the main authentication method for the majority of online services [47]. The recent rise of data breaches further increased the threats to password-based authentication [29, 35, 63]. Credential stuffing and password spraying attacks automatically enter leaked login credentials (username and password) on other websites in the hope that users have reused them [16, 28]. Such attacks can be scaled with little effort and promise financial returns, which is why they are very popular today [4, 43]. Targeted password guessing methods using machine learning (ML) are less scalable but also very effective [44, 68]. This increases the need for responsible online services to protect their users. Two-factor authentication (2FA) [46] is a commonly offered measure to increase account security, but users tend to refuse it outside of use cases involving sensitive and financial data, such as online banking [20, 37, 41, 48, 49, 65, 70]. The 2FA adoption rates are low on popular online services, for example, 2.5% on Twitter [65] and ≈4% on Facebook in 2021 [41]. Google also has a low 2FA adoption rate [37, 46], which is why they tried to force 2FA on 150M of their users, that is, less than 10% [25], by the end of 2021 [26]. The outcome of this experiment regarding user acceptance, however, is unclear to date. To still protect users without enabled 2FA and to increase the cost for attackers, risk-based authentication (RBA) is a solution that is preferred by users over equivalent 2FA in many use case scenarios [70]. It can be considered a scalable interim solution to strengthen password-based authentication until more secure authentication methods are in place. Therefore, government agencies such as the National Institute of Standards and Technology (NIST, United States), the National Cyber Security Centre (NCSC, United Kingdom), and the Australian Cyber Security Centre (ACSC, Australia) recommend RBA to protect users against attacks involving stolen passwords [7, 27, 40]. US federal agencies even have to establish RBA across their enterprise by presidential executive order [8].

1.1 Risk-Based Authentication (RBA)

RBA estimates whether a login is legitimate or an account takeover attempt. This is done by monitoring and recording a set of features that are available in the login context. Potential features range from network (e.g., IP address), device or client (e.g., user agent string), to behavioral biometric information (e.g., login time) [69]. Based on the feature values of the current login attempt and those of previous ones stored in a login history, RBA calculates a risk score related to the login attempt. Optionally, the risk score can also consider known attack data to identify suspicious feature values or potential users at risk (e.g., VIPs) [24]. An access threshold typically classifies this risk score into low, medium, and high risk [71]. Based on the risk classification, the online service performs different actions [24, 71]. On a low risk (e.g., same device and location as always), access is granted without further service intervention. On a medium or higher risk (e.g., unusual device or location), RBA can ask for additional information to prove the claimed identity (e.g., verifying the email address), or even block access. The choice of the access threshold impacts RBA's overall usability and security [69]. When the threshold is set too low, RBA requests re-authentication every login attempt, making the look and feel identical to classical 2FA. When set too high, RBA never requests re-authentication but also does not provide any security benefit, as all attacks are classified as legitimate login attempts. Sensible RBA implementations range between these extremes. They only request re-authentication for a fraction of legitimate login attempts but also protect users against a high number of attacks.

Some popular online services started using RBA [43, 69, 71]. However, there is still a lack of open resources to support research and development. Only a few research papers scientifically evaluated RBA algorithms and configurations, and these used only small datasets [24, 69, 73]. In

addition, there are no open-source RBA solutions available beyond a very basic and weak algorithm provided by the single sign-on (SSO) solution OpenAM [42, 69]. More open resources can support online services to fully understand RBA and, therefore, enable a broad RBA deployment in practice.

1.2 Research Questions and Contributions

Following governmental recommendations and obligations, increasing numbers of online services will have to deploy RBA in a few months [8]. For real-world large-scale online services, however, RBA has not been widely understood to date. This is crucial, as even small configuration errors may largely impact usability and security for many users [69]. Previous studies already investigated RBA on a small online service [69, 73], but they have limitations: they are limited to a small online service with a user base mostly located in one city. Therefore, it is unclear whether the results regarding the RBA behavior hold for a globally distributed large user base.

We provide the first in-depth analysis of RBA on a large-scale online service, including 3.3M users and 31.3M login attempts collected over more than 1 year. We evaluated RBA with four attacker models related to real-world attacks. The service, belonging to the multinational telecommunications provider Telenor, provided access to sensitive data, for example, cloud storage and billing data. This made it a suitable RBA use case [70]. To investigate RBA on a large online service, we formulated and answered a set of research questions, which we outline here with our contributions. We answered the questions with the collected data. Two of the following research questions (RQ1a and RQ1c) replicate a related study [69], while the other research questions aim at enhancing our understanding of RBA on a large online service.

RBA's characteristics had previously been analyzed in a related study made on a small online service [69]. A clear study limitation was that the dataset was rather small and specific, and did not contain attack data. However, to investigate the broad applicability of the study's statement, it needs to be validated with a large user base. Therefore, we replicated this study in order to validate whether the results also hold for a large online service. We also analyzed how the RBA behavior depends on different login frequencies (e.g., daily to less frequent use).

RQ1: **(a)** How often does RBA request re-authentication on a real-world large-scale online service?
    (b) How does the number of re-authentication requests differ between users' login frequencies?
    (c) How many user sessions need to be captured on a large-scale online service to achieve a stable and reliable setup?
    (d) How do RBA characteristics differ between a large-scale and a small online service?

> **Main Findings**
>
> The results confirm previous findings [69] that RBA rarely requests re-authentication in practice, even when blocking more than 99% of attacks that used the victim's credentials and mimicked some of the legitimate login contexts. Going beyond the previous study, the results also show that the login frequencies significantly impact the number of re-authentication requests.

Other aspects of RBA have not yet been studied due to a lack of available real-world, large-scale user login data [51, 73]. These aspects included, for example, how attack data [24] and login history minimization [73] influence RBA's usability and security properties. Since our dataset is large and contains attack data, we are able to analyze these aspects.

RQ2: **(a)** How does RBA perform when taking user-based attack data into account?
    (b) How does RBA perform when also taking feature-based attack data into account?

RQ3: How long do we need to store login history data to maintain good RBA security and
usability?

Main Findings

The results suggest that attack data should be used with caution or not at all. Login history
minimization proved to be useful to increase privacy while balancing usability and security.

The current way of setting up RBA appears to be complicated in practice, as administrators need
to set and test all parameters manually, which is error prone and can largely impact usability and
security [24, 51, 69]. Therefore, we introduce and tested an ML-based RBA parameter optimization
that assists administrators in finding an optimal RBA setup for their online service.

RQ4: (a) In what way can a classical RBA algorithm be enhanced with ML mechanisms to optimize
the usability and security parameters?
(b) How do ML-based RBA optimizations perform compared with a classical algorithm?

Main Findings

The results show that the proposed optimization can provide an RBA configuration baseline
to achieve good usability and security in a short period of time.

The RBA performance of current algorithms likely degrades substantially with large global login
history sizes, as is the case with large-scale online services [69]. To tackle this problem, we also
developed an algorithm optimization to speed up RBA for large-scale online services.

RQ5: What are performance implications of RBA on a large-scale online service and how can they
be improved?

Main Findings

The results identified non-optimized RBA as vulnerable to denial of service when users log in
at similar times. Our optimizations using hash tables were able to shorten the authentication
times with a 28x speedup.

Due to its hard-to-spoof properties, a server-originated round-trip time (RTT) feature [69] showed
potential to replace IP addresses as a privacy-enhancing alternative [73]. As this claim has not been
investigated and validated on a real user base so far, we evaluated the RTT's potential to verify
countries and regions and as an IP address replacement with our dataset.

RQ6: To what extent can the server-originated RTT be used as a privacy-enhancing alternative
to the IP address feature?

Main Findings

The results indicate that the RTT has strong potential as a drop-in replacement for the IP ad-
dress feature. It can identify legitimate users with good usability, security, and privacy while
increasing the costs for attackers trying to spoof the IP geolocation, that is, they need access
to a device physically located in the target area instead of just using a VPN connection.

To address the current lack of high-quality open data for RBA research, we further provide a
synthesized RBA dataset[1] (see Appendix A for details). The synthesized dataset resembles the
statistical properties of our original dataset that we had to delete for privacy reasons (see Section 4).

---

[1]Available at https://github.com/das-group/rba-dataset.

The results of our studies show that RBA's state of practice can significantly be improved. Its usability and security properties strongly depend on implementation and configuration details. Our research advances RBA knowledge and development, further supporting the latter with the provided dataset. It allows service owners to decide on RBA configurations for their online services. Administrators and developers get advice on selecting suitable RBA settings and optimizations to balance usability and security. Researchers get insights on RBA's behavior on a large online service.

2 RBA MODEL SELECTION

Different RBA models exist to derive a risk score for a login attempt. The first approach was published by Freeman et al. [24] in 2016 and is based on statistical relations. Other models, some more related to the area of implicit authentication, are based on more complex ML schemes such as Siamese neural networks [2, 17, 51] and one-class support vector machines (OCSVM) [38]. Other relevant ML schemes for outlier detection include isolation forest, local outlier factor, and robust covariance [54]. They have not been seen in RBA literature yet, but we found them relevant for our model selection. We used the RBA model of Freeman et al. since a preliminary evaluation showed it to be the most suitable for our dataset. We outline the model selection in the following.

We aimed to focus on scalable and practical RBA solutions that can be used with large-scale online services. Like other popular online services observed in practice [71], the online service collected categorical feature data, for example, IP address and user agent. Two types of RBA models exist in literature that focus on categorical data [69]. One type, the most published one (SIMPLE model), is very simple and checks only exact matches in the user's login history [18, 30, 42, 61]. The other type, published by Freeman et al. [24] and probably used by popular online services [71], compares statistical feature relations between all users. This model outperformed the other models in a previous evaluation [69]. To provide a baseline comparison, we also repeated this evaluation on the large-scale online service (see Appendix E). Both results confirm that the SIMPLE model cannot block the same high amount of attackers as the model by Freeman et al. The latter also allowed more fine-grained security configurations than the SIMPLE model. For instance, varying a tiny fraction of the access threshold greatly lowered VPN attacker protection from 96% to 53% with the SIMPLE model on the large-scale online service. In contrast, attacker protection lowered for only a tiny fraction with the model by Freeman et al. On a geographically close user base, it also blocked significantly more targeted attackers while making significantly fewer requests to re-authenticate [69].

We also evaluated ML schemes for outlier detection on a subset of our collected data to assess their capabilities in terms of RBA risk estimation. However, they did not qualify for our use case for the following reasons: (i) Our feature data are categorical with a nominal scale, that is, they have no ranking. We did not find a rational reason to rank the categorical feature values in descending order, with each having an individual rank (e.g., is Firefox Mobile a better feature than Firefox Desktop?). Most of these ML schemes, however, currently showed good performance only with numerical data having an interval scale, for example, time or sensor measurements [2, 17, 51]. We acknowledge that nominal categorical data without a ranking can also be transformed into binary numerical data with one-hot encoding [53]. Due to their properties, however, unknown values—such as a new user agent—can be transformed into only one value, that is, the "unknown" category [10]. As this one-hot encoded value has a low scalar distance to the known values, the classification produced inaccurate results in our tests. (ii) The outlier detection of these mechanisms proved to be suitable only when including the user's own login history in the training [38]. Otherwise, as the global feature values in our data were largely distributed, the mechanisms would not detect any outliers in most cases. The data per average user, however, were very sparse at the online service, that is, four categorical data points per feature (see Section 4). Based on our tests,

this was not enough for sensible outlier classification using most of these models, that is, almost every login risk is high unless all feature values are the same. (iii) In contrast to the selected model, these ML- and non-ML–based mechanisms did not allow for a relation between local and global feature value distributions. This led to inaccurate results with respect to newly established legitimate feature values, for example, new browser versions across multiple users. (iv) In addition, the ML mechanisms need to be trained with new feature values for each login to optimize the outlier detection for each user. Otherwise, legitimate users would potentially be prompted for re-authentication more often, as new legitimate feature values would be lacking in the trained model. This, in combination with the classification, was time-consuming compared with the Freeman et al. model (e.g., 2.5 s for isolation forest and 270 ms for OCSVM, compared with 0.003 ms for the Freeman et al. model). Therefore, we did not consider them practical for large-scale online services in their current form.

As a result, to focus on a practical and scalable solution for large-scale online services, we chose the RBA model of Freeman et al. [24] for further evaluation. Nevertheless, as its risk score has a numerical interval scale, we still found ML mechanisms useful to optimize the selected RBA model parameters (see Section 10). Parameter optimization can be used as a tool for administrators to set up RBA as effectively as possible in their individual environments.

3   FREEMAN ET AL. RBA MODEL

The selected RBA model is comparable to a multi-features model derived from observations on the behavior of popular online services [71]. According to the observations, the online services Google, Amazon, and LinkedIn used this model and probably still use it in some form [71]. The model calculates the risk score S for user u and a given feature set $(FV^1, \ldots, FV^d)$ with d features as

$$S_u(FV) = \left(\prod_{k=1}^{d} \frac{p(FV^k)}{p(FV^k|u, legit)} \right) \frac{p(u|attack)}{p(u|legit)}. \tag{1}$$

The calculation uses the probabilities $p(FV^k)$ of a feature value in the global login history, and $p(FV^k|u, legit)$ of a feature value in the legitimate user's login history. The user login probability is based on the proportion of the user logging in, that is, $p(u|legit) = \frac{Number\ of\ user\ logins}{Number\ of\ all\ logins}$. Attack data can optionally be included to protect users at risk. As in related work [69], we considered all users equally likely being attacked when not having attack data. Thus, we set $p(u|attack) = \frac{1}{|U|}$ with U being the set of users with $u \in U$. We did not use attack data in most of our studies, unless otherwise noted, to be able to compare our results to a related study [69]. We also assumed that use cases without attack data are more common in practice, especially for medium and small websites that have limited storage and computing capacity.

When considering attack data, we estimated $p(u|attack)$ based on the number of failed login attempts per user. We expect that users with a high number of failed login attempts are likely being targeted in credential stuffing or password spraying attacks. If the user is not present in the attack data, we set the probability to the minimum value: $p(u|attack) = \frac{1}{Total\ number\ of\ logins}$. We did this as it is still possible that other users are being attacked.

Furthermore, the equation can be extended to include an attack probability for the used feature values:

$$S_u(FV) = \left(\prod_{k=1}^{d} p(attack|FV^k) \frac{p(FV^k)}{p(FV^k|u, legit)} \right) \frac{p(u|attack)}{p(u|legit)} \tag{2}$$

In our use case, $p(attack|FV^k)$ is the number of feature value occurrences in the failed login attempts. Similarly, we returned the minimum probability when the feature value was not present in the attack data, that is, $p(attack|FV^k) = \frac{1}{Total\ number\ of\ attacks}$. Returning a minimum probability is necessary, as we assume that there is always a residual risk of other feature values being used in an attack.

4 LOGIN DATASET

We collected login feature data of 3.3M users between February 2020 and February 2021 at an SSO service belonging to the Norwegian telecommunications company Telenor Their customers used the SSO service to access other services provided by the company, for example, cloud storage or billing information services. The users could authenticate with their username in two ways: either by entering a password or by letting the online service's mobile network verify a mobile phone number (see Section 12). In some cases, users also had to verify their mobile phone number or email address by entering a one-time code (OTP) that was sent to the corresponding number or address.

4.1 Login Sessions

We collected feature data of 31.3M login attempts; 12.5M were successful and 18.8M failed. Of the failed login attempts, 25% had a correct username. Out of the failed ones with a correct username who proceeded with the login attempt, 90% failed at the password entry, 9.9% at the OTP entry, and 0.1% at the mobile network verification step. Furthermore, 87 successful logins were detected as account takeover by the security incident response team of the company. The users logged in between 1 and 5,972 times (mean: 3.8, median: 2, SD: 9.35, see Figure 1). The majority of users logged in less than a month (last login was more than a month ago, 0-1 times a month on average) (48.3%) or daily (22.4%). They used mobile (65.3%) and desktop devices (34.6%). The other device types were bots and non-identifiable devices. The operating systems (OS) of desktop devices were Windows (79.2%), macOS (19.4%), and Linux (1.4%) based. Most mobile devices were Android (64.9%) and iOS (35.1%). The majority of online browsers were Chrome (59.8%), Safari (27.4%), Edge (5.9%), and Firefox (3.0%). The order of the browser percentages corresponds to the global and country's browser market share [59, 67].[2]

The online service was used by customers residing in Norway. We did not collect demographical data for privacy reasons. However, both the online service and company target a general audience. Also, the dataset's user count is around 2/3 of the country's population. For these reasons, we expect that the data likely represents the country's demographics.

4.2 Features

Due to storage limitations and privacy reasons at the large-scale online service, we had to limit our feature scope for data collection. Previous work identified a subset of useful features out of 247 collected client- and server-originated features [69]. We used this work to focus on a subset of the most relevant RBA features that provide good security and usability. As a result, the dataset contained the features *IP address*, *user agent string*, *server-originated RTT*, and *login timestamp*. The online service derived subfeatures from the IP address (ASN *[autonomous system number]*, country, region, and city) and the user agent string (browser/OS name and version, and device type). The full plain-text IP address was dropped after this step for privacy reasons. The dataset used by the researchers included two privacy-enhanced versions of the IP address, following suggestions from related work [73]: (i) The full IP address was salted and hashed by the online service to

---

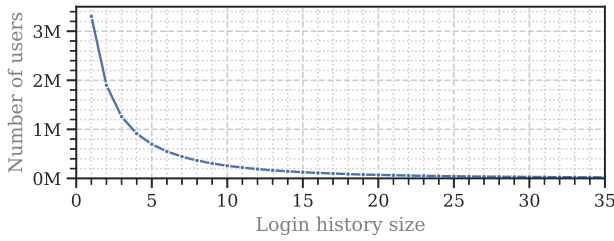[2]Citation omitted for blind review.

Fig. 1. Available login history sizes per user count. We cropped the plot on the x-axis for readability (the maximum size was 5,972).

mitigate potential re-identification. We used this variant in our RBA model. The hashing only transformed the data representation, which is why it did not affect the results produced by the model [73]. (ii) The IP address was also included in a truncated form with the last octet removed. We used this version for further analysis to extract a subset of hashed full IP addresses that are potential attack IP addresses. As the company used a very long random salt, the truncated IP still did not allow us to reconstruct the full IP address. However, for data protection and ethical reasons, we did not even attempt this.

4.3 Feature Optimization

Service owners need to optimize the collected features to improve the expected RBA performance. Based on optimizations found in related RBA and browser fingerprinting literature [5, 24, 30, 61, 69], the service extracted subfeatures from the IP address and user agent string. Administrators also need to weigh the different features to optimize performance. The weighting ensures that certain features are treated with higher priority, for example, those that are harder to spoof. We calculated the weightings with the method described in Freeman et al. [24]. The calculated weightings for IP address (IP address: 0.6, ASN: 0.3, country: 0.1) and user agent (full string: 0.55, browser: 0.29, OS: 0.15, device type: 0.01) were very similar to those of a related study [69]. Thus, for comparability, we set the same weightings for the IP address (IP address: 0.6, ASN: 0.3, country: 0.1) and user agent (full string: 0.53, browser: 0.27, OS: 0.19, device type: 0.01).[3]

4.4 Ethical and Legal Considerations

We do not have a formal institutional review board process at our university. However, in addition to our ethical considerations discussed here, we made sure to minimize potential harm by complying with the ethics code of the German Sociological Association (DGS) and the standards of good scientific practice of the German Research Foundation (DFG) We also made sure to comply with the European General Data Protection Regulation (GDPR) [22].

By using the SSO service, the users agreed to data collection and evaluation for research purposes. To minimize potential harm, we worked on a privacy-preserving dataset. All features and the processes of data pseudonymization and processing were carefully reviewed over more than 2 months and approved by the company's legal department. The data were provided by the company for research purposes only and deleted after research completion. The data were stored on an encrypted external hard drive. Only the researchers had access to the drive and the decryption

---

[3]To assess our results' general validity, we recalculated RQ1's results using the weightings calculated for the dataset. As these were nearly identical to those using the related study's weightings, this encouraged us to use the latter for a fair comparison.
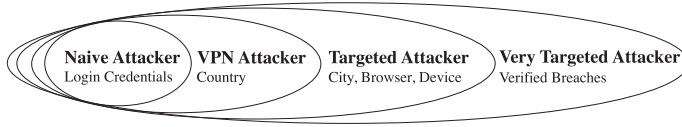
Fig. 2. Overview of the attacker models used in the study, with their increasing capabilities to mimic a legitimate user (extended from Wiefling et al. [69]).

password. For study reproduction and fostering RBA research, we entered into an agreement with the data owner to create a synthesized dataset that does not allow re-identification of customers.

5   ATTACKER MODELS

We evaluated the RBA model with four attacker models, with three of them based on known models in literature [24, 69, 72]. Due to verified account takeover data and the relevance in real-world scenarios [12, 63], we introduce the additional *very targeted attacker*. We describe the attacker models in this section. All attackers possess the login credentials of the victim (see Figure 2).

The **naïve attacker** signs in from a random IP address from somewhere in the world. We simulated this attacker with IP addresses of failed login attempts that match a dataset identifying real-world attacks [23]. We expect that a large number of failed login attempts for a user likely indicates that the user was targeted in an attack. However, the majority of identified attacks came from a particular ASN and a country that was far from the online service's main country. Therefore, detecting those would be trivial. To better diversify the sample, and to simulate global attackers, we included only the most occurring IP address per ASN. Then, we extracted two types of naïve attackers to simulate a wide range of them. Half of the attacks resembled a very naïve attacker type comparable to one that randomly tries to enter a login credential. These attacks involved random IP addresses from all 180 countries identified in the dataset.[4] The user agents were also randomly selected. The other half of attacks resembled naïve attackers that were able to buy additional resources, for example, servers or botnets. These attacks involved the top IP addresses and user agents having most occurrences in the dataset.

The **VPN attacker** knows the country of the victim. Therefore, the attacker spoofs the IP geolocation using a VPN connection and popular user agent strings. The IP address does not necessarily have to belong to a VPN service provider. We simulated this attacker with identified attack IP addresses located in the users' main country [23]. As this attacker does not target specific users, we randomly sampled these attack IP addresses from the failed login attempts. The attacker used the most popular user agents from legitimate login attempts.

The **targeted attacker** knows the location, device, and browser of the victim. Therefore, the attacker uses IP addresses and user agent strings that the victim would likely choose. We simulated this attacker with identified attack IP addresses located in the users' main country [23]. As the attacker targets specific user groups, we sampled the IP addresses and user agents of the failed login attempts with most occurrences.

The **very targeted attacker** is the strongest of our attacker models, which attacks a victim only once, but with all information gathered about it. This could, but not necessarily, include accessing the device or the network of the victim by malware. We simulated this attacker by replaying the 87 identified successful account takeovers. We did this to check how many of these very targeted attacks would be detected by the RBA system in practice.

---

[4]We could not fill in missing countries because we were limited by the pseudonymized IP addresses, that is, we could use only those that were provided to us.

6   METHODOLOGY

We did the calculations for our analysis on a high-performance computing (HPC) cluster with more than 2,400 CPU cores and 7,500 GB RAM. This was crucial, as calculating the risk scores was computational intensive, especially with the large dataset used. Computing the results on an eight-core desktop PC would have taken more than a year. The HPC cluster reduced this computation time to approximately 1.5 weeks.

Feature Set. Unless otherwise noted, our RBA model used IP address and user agent string as features. The subfeatures were ASN and country for the IP address, and browser, OS, and device type for the user agent string. We did this to allow a fair comparison to the related study and since this feature set can be considered the state of RBA practice [69, 71].

Optimizations. The huge dataset size largely affected computation time and limited computing capacity. Thus, we optimized the RBA algorithm to speed up calculation (see Section 11.1). The optimizations allowed us to calculate the risk scores for all legitimate logins. However, we still had to limit the attacker calculations to a data subset to be able to obtain the results after a reasonable period of time.

Stratification. Due to the increased storage and computational load caused by the attacker simulation, we calculated the attackers' risk scores for naïve, VPN, and targeted attackers on a stratified dataset containing 10% of users. We stratified the user set based on the number of logins to retain the dataset properties regarding login histories as efficiently as possible. Stratifying datasets is common practice in ML solutions [55]. To focus our attack detection on real-world attacks, the stratified set included those users who had the highest number of failed login attempts, that is, likely targeted users.

   To make sure that the stratification did not influence the general validity of our results, we compared the results for RQ1 using all legitimate logins to those using a stratified dataset. The results were identical for the very large majority of login history sizes lower than 100. Thus, we considered the stratified sample appropriate for our analysis.

Significance Testing. For statistical tests, we used Kruskal-Wallis (K-W) for the omnibus cases and Dunn's multiple comparison test with Bonferroni correction (Dunn-Bonferroni) for post-hoc analysis. We considered p values lower than 0.05 as significant.

7   EVALUATING RBA IN PRACTICE (RQ1)

To estimate RBA's performance on a large online service, we repeated a related study [69] with our dataset. The related study estimated RBA characteristics on a small online service. In addition, since the dataset was large enough to do this, we further analyzed the number of re-authentication requests depending on login frequency. We used the same methodology, as it has already been used for RBA analysis. This allowed us to compare the results.

7.1   Study Procedure

We first calibrated the risk scores regarding the percentage of blocked attackers for each attacker model, that is, the true positive rate (TPR) [24, 69, 73]. We then replayed all legitimate login attempts as they had happened on the online service. For each login attempt, we (i) restored the state of the dataset at the time of login, (ii) calculated the risk score, (iii) applied the access threshold for the given TPR and attacker model, and (iv) stored the access decision.
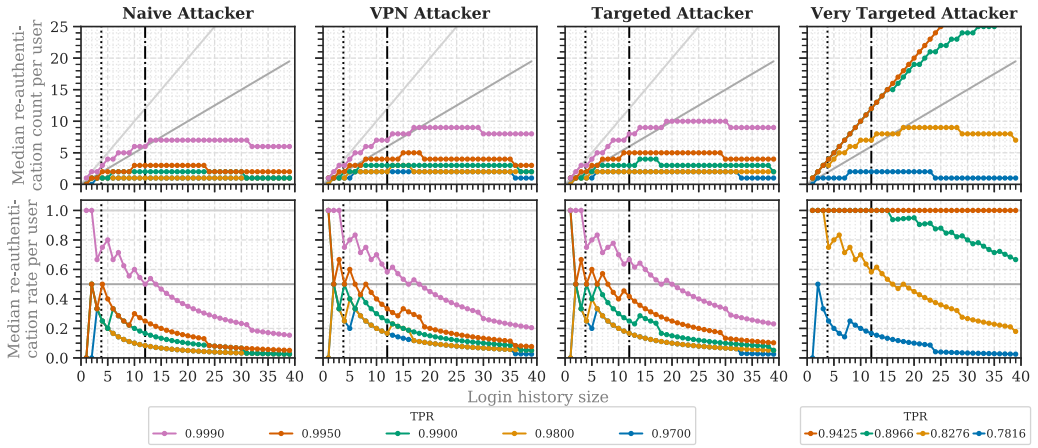
Fig. 3. Median re-authentication counts and rates for the four attacker models. The true positive rate relates to each of the attacker models. For orientation, we added the baseline for 2FA (light-gray line), the stable setup threshold (gray line), and the mean login counts for our study (dotted black line) and the study of Wiefling et al. [69] (dash-dotted black line).

7.2 Results

We now present the results in order of the research questions, followed by a discussion.

Number of Re-authentication Requests (RQ1a). The users logged in 3.8 times on mean average. Therefore, we considered a user's login history size of 4 to estimate the re-authentication count for the average user. The median re-authentication count decreased with an increasing login history size (Figure 3).

When configuring RBA to block 99.5% of **naïve attackers**, the average legitimate user was asked for re-authentication every second login. When blocking a lower percentage, users were prompted every fourth login (TPR 0.99–0.95) or not at all (TPR <0.95). When blocking **VPN** or **targeted attackers**, legitimate users were prompted every second (TPR 0.995–0.99) or fourth time (0.98–0.90). When blocking at least 82.76% of **very targeted attackers**, legitimate users were asked every login attempt.

For a fair comparison to the related study [69], we also considered a login history size of 12 to estimate the median login count until re-authentication. In this case, users were asked every fourth login or even less often for TPRs lower than 0.995 (Table 1). This is similar to the related study. Summarizing our findings, RBA rarely asks for re-authentication in practice, even when blocking a high number of targeted attackers.

Re-authentication Requests per Login Frequency (RQ1b). We classified the users of our dataset into five groups having different login frequencies: daily, several days per week, once a week (weekly), once in several weeks, and once in more than 30 days (less than monthly). For each user, we calculated the time differences between successive logins. We then classified the user to the group to which the majority of time differences belonged (e.g., the user used the online service mainly on a daily basis). To ensure a sufficient data basis for the classification, we then dropped all users who logged in on less than two different months, two different weeks, and four different days, depending on the class. We based the limit on the median login count and our own observations on the different user groups in the dataset. Based on the labels, we calculated the re-authentication counts per login frequency.

Table 1. Median Login Count until Re-authentication at a
Login History Size of 12

| TPR | Median logins until re-authentication | | | TPR | |
	Naïve Attacker	VPN Attacker	Targeted Attacker		Very targeted attacker
0.9990	2	1.71	1.5	0.9425	1
0.9950	4	3	2.4	0.8966	1
0.9900	6	4	4	0.8276	1.71
0.9800	12	6	6	0.7816	6
0.9500	12	6	6		
0.9000	∞	12	12		

Table 2. Dunn-Bonferroni p values for the Different Login Frequency Types
(TPR 0.995, Targeted Attacker)

	Daily	Several Days	Weekly	Several Weeks	>30 Days
Daily	-	**0.0229**	-	**0.0108**	**<0.0001**
Several Days	**0.0229**	-	-	-	0.0924
Weekly	-	-	-	0.1881	**<0.0001**
Several Weeks	**0.0108**	-	0.1881	-	**0.0131**
>30 Days	**<0.0001**	0.0924	**<0.0001**	**0.0131**	-

The users logged in daily (<1 day), once in several days (1–7 days), weekly (7–14 days), once in several weeks (14–30 days), and less-than-monthly (>30 days). We omitted p values greater than 0.2 for readability. Bold: Significant.

The median number of re-authentication requests varied significantly between the different login frequency groups (Table 2). Daily users were asked significantly less for re-authentication than those who logged in once in several days and less than several weeks. For instance, when aiming to block 99% of VPN and targeted attackers, the average daily user had to re-authenticate every fourth login, whereas weekly or less frequently logging in users had to do it almost every time (Figure 4). Less-than-monthly users were also prompted for re-authentication significantly more than those who logged in daily, weekly, or once in several weeks. Concluding the results, daily users were mostly less asked for re-authentication than those who logged in less frequently.

Required Login History Size (RQ1c). In order to achieve RBA's usability gain, users need to notice a difference to 2FA [70]. A baseline for this can be to request re-authentication less than every second time [69]. Thus, we considered the required login history size as the point after which the median re-authentication rate remained below 0.5.

Following this definition, most TPRs lower than 0.99 required one or no entry in the login history for a stable setup (see Figure 3). The other TPRs required 4 (naïve attacker, TPR 0.995; VPN attacker, TPR 0.99), six (targeted attacker, TPR 0.99), 8 (VPN and targeted attacker, TPR 0.995), and 18 entries (very targeted attacker, TPR 0.8276). This would achieve a baseline setup that, on median, requests re-authentication less than every second time.

However, the required login history size differed based on the login frequency of the users. Even when blocking 99.5% of all targeted attackers, the median re-authentication rate for daily
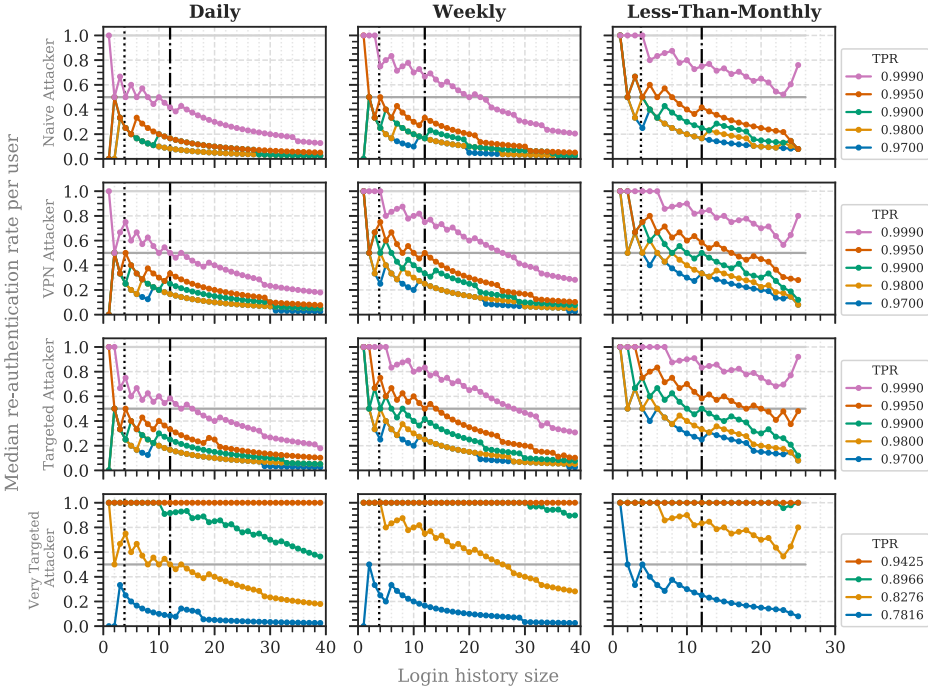
Fig. 4. Median re-authentication rates for the attacker models, filtered by login frequency. We added the baseline for 2FA (light-gray line), the stable setup threshold (gray line), and the online service's mean login count (dotted black line) for orientation.

users remained below the baseline after a login history size of 4 (see Figure 4). Thus, in this case, 4 entries were required for a stable setup targeted at daily users. This was different from the other user types, which required at least 8 entries for targeted attackers in this case. The results for the very targeted attacker did not pass the baseline for TPRs greater than 0.89 until a login history size of at least 40.

In general, the login history of each user does not have to contain a large amount of entries for a stable setup. Based on the results, we conclude that storing 8 entries is sufficient for a stable RBA setup that blocks 99.5% of targeted attackers at this online service. When blocking naïve or VPN attackers, 4 to 6 entries are required in that case.

Comparison with Small Online Service (RQ1d). Wiefling et al. [69] evaluated RBA on a small online service having 780 users and 9,555 legitimate logins. Due to the same setup, we can compare both studies. Our results mainly reflect their findings (Figure 5). Users were less requested for re-authentication with an increasing login history size. Although our re-authentication rates mostly declined slower, the main tendency was the same. Therefore, the RBA characteristics did not differ greatly between a large and small online service.

7.3 Discussion

The vast majority of identified attacks on the online service came from a different country than the victim (2.2M incidents, 97%), representing naïve attackers. VPN, targeted, and very targeted attacks using the same country as the victim were less common (58K incidents, 3%). Therefore, blocking
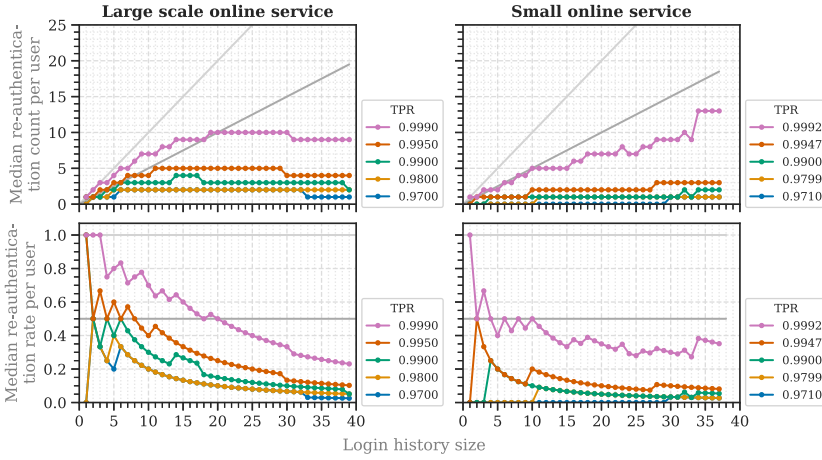
Fig. 5. Median re-authentication rates and counts for targeted attackers compared with a large-scale (our results) and a small online service (Wiefling et al. [69]).

naïve attackers already helps to protect against most attacks in practice. Freeman et al. [24] reported similar findings regarding botnet attacks, that is, only 1% involved the same country.

The data for very targeted attackers were very sparse, which is why the TPRs were very coarse grained. Still, the results show that RBA was able to detect the majority of successful account takeovers (78.16%) with low re-authentication rates.

The results confirm that RBA can achieve good usability and security after only a few login attempts. However, this varies depending on the user type. Less-than-monthly users had to re-authenticate significantly more than daily users. This is due to the fact that feature values are more likely to differ after a longer period of time because, for example, the device or browser was updated or a new IP address was set [6].

Daily to weekly users received less frequent re-authentication requests, even when RBA was configured for high security. Therefore, RBA can achieve a high acceptance among these users in practice [70]. Less frequent users, however, would be prompted for re-authentication almost every time in this case. The burden of re-authentication is limited here, as users are prompted once per month. Thus, we expect that they will tend to accept it [15, 32, 70]. However, when being surprised by a re-authentication request after some months, there is a risk of users not being able to solve the re-authentication challenge [70]. This could result in users being annoyed by RBA [15, 32, 70]. To mitigate this effect, we recommend to set the targeted TPR based on the login frequencies of the general user base. The TPR can be set high for a mostly daily user base, while it has to be lowered for less frequent users. For instance, to achieve a user experience for less-than-monthly users that is comparable to daily users, the TPR needs to be lowered from 0.995 to 0.97 for targeted attackers at this online service (see Figure 4).

8 EVALUATING RBA WITH ATTACK DATA (RQ2)

The RBA model offered two possibilities to include attack data (see Section 3) that had not been investigated to date. Thus, we evaluate the attack data variations in this section. We simulated the login behavior as in the previous study (see Section 7.1). However, we added the two different attack data variations in our risk score calculation. We outline the procedures together with the results in the following.
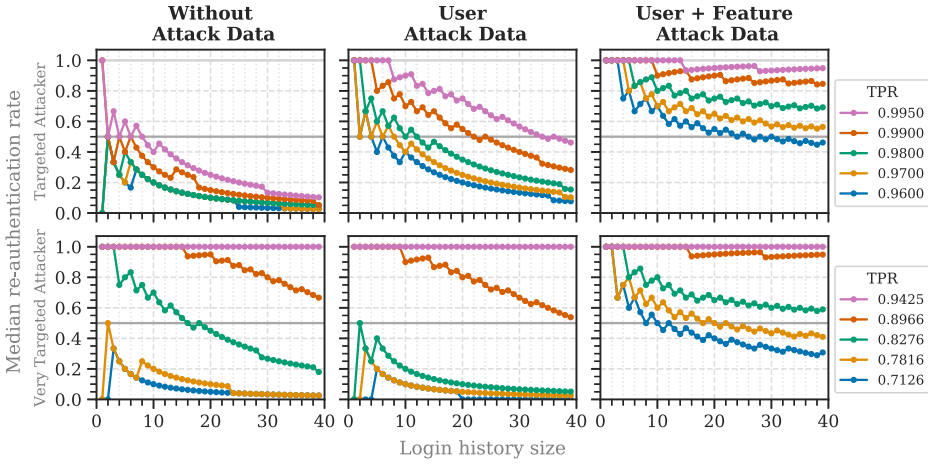
Fig. 6. Median re-authentication rates per user for targeted and very targeted attackers when adding user attack data (RQ2a) and both user and feature value attack data (RQ2b).

User Attack Data (RQ2a). In the first variation, we used the user attack probability $p(u|attack)$ calculated from the number of failed login attempts in the attack data. Figure 6 shows the results. When adding the user attack probability, the re-authentication rate significantly increased for naïve, VPN, and targeted attackers ($p \ll 0.0001$). For very targeted attackers, however, the re-authentication rate significantly decreased for TPRs lower than 0.94 ($p \ll 0.0001$). Thus, very targeted attackers could be better distinguished from legitimate users than before, whereas the opposite was true for the other attackers.

User and Feature Value Attack Data (RQ2b). In the second variation, we further added probabilities for feature values being used in an attack. Therefore, we used Equation (2) (see Section 3) with both the user attack $p(u|attack)$ and the feature attack probability $p(attack|FV^k)$ calculated from the failed login attempt data.

When also including the feature attack probability, the re-authentication rates significantly increased for all attacker models ($p \ll 0.0001$, see Figure 6). Thus, attackers could not be distinguished from legitimate users as well as before.

Discussion. Using the user attack data in the RBA model reduced re-authentication requests when blocking *very targeted attackers*. As the targeted users were likely attacked before, the risk scores increased for them. However, the risk scores decreased for attacks on those users who were not attacked before. Thus, the usability performed worse for the other attacker models that targeted a wide range of users. Therefore, the decision to include attack data should be made with great caution. Related work suggests that other metrics, such as social media followers [24] or frequently guessed passwords [64], could help to identify accounts that should receive higher protection. This, however, requires further research.

Adding feature attack data made it harder for the RBA system to distinguish between legitimate users and attackers. Attackers identified in the dataset likely chose popular feature values, for example, the user agent. Thus, legitimate users having these feature values received a high risk score. Attackers trying unpopular feature values, however, likely received risk scores in the range of legitimate users using popular feature values. As a result, legitimate users could not be distinguished from attackers as well as before. With this is mind, we do not recommend using the tested feature attack data variant in the RBA model to calculate the risk score.

9 LOGIN HISTORY MINIMIZATION (RQ3)

Over the last few years, an increasing number of data protection regulations aimed to protect users from massive data collection by online services. These regulations, such as the CCPA [60] and GDPR [22], suggest limiting data storage as much as possible. In terms of RBA, this means that the feature data have to be disposed of as soon as they no longer serve the purpose of RBA [73]. One measure to minimize data is to delete global login history entries after a fixed amount of months [73]. Such a data-limiting procedure is also useful to maintain an acceptable authentication speed [69]. To estimate the potential of the minimization approach, we evaluated it in the coming sections.

9.1 Study Procedure

We simulated the login behavior as in the first study (see Section 7.1). For each login, however, we kept only the global login history within three different monthly ranges (1, 3, and 6 months). We selected these ranges to estimate the scope of possible effects when minimizing the login history. We further included the full dataset results (12 months) for comparison. To enable RBA protection for all users, including less-than-monthly ones, we kept at least one entry in the user's login history.

  We attacked each user at the point at which the user last logged in to observe the possible RBA protection with login history minimization. We had to select a specific point in time for the attacks due to the limited computing and storing capacity. Therefore, we selected the point with the highest login history size distribution to observe a wide range of effects. Otherwise, when selecting later points in time, the login history sizes of all users would approach one. Thus, we would not likely be able to observe daily to several weeks users in the results.

9.2 Results

For all attacker models, the median re-authentication rate per user increased when minimizing the global login history (Figure 7). The differences between the monthly ranges were significant for all attackers ($p \ll 0.0001$), except for the naïve attacker between 3 and both 6 and 12 months (TPR 0.995; see Figure 7). When keeping less than 6 months, the median re-authentication rate for daily users increased after a few logins (VPN: 6 months; very targeted/targeted: ≤3 months; naive: 1 month).

9.3 Discussion

The results show that, in terms of naïve attackers, reducing the global login history to 3 months can be possible without significantly influencing the median re-authentication count. However, this is different when blocking more intelligent attackers. For targeted attackers and a high TPR (≥0.99), daily users were prompted for re-authentication at almost every login when keeping less than 6 months. Users would not accept such frequent re-authentication [15, 32, 70]. Therefore, when aiming to block a high number of attackers, keeping between 6 and 12 months of login history seems reasonable for a stable setup on the online service.

  Our results showed the login history minimization's potential to increase user privacy while maintaining usability and security. Further research should investigate the effects of limiting both the user's login history and the global login history to a maximum size. These approaches could also potentially reduce authentication time and required storage.

10 ML-BASED RBA PARAMETER OPTIMIZATION (RQ4)

In order to optimize the usability and security properties of RBA, administrators need to analyze the risk scores of legitimate users and attackers to set a suitable access threshold. This is a
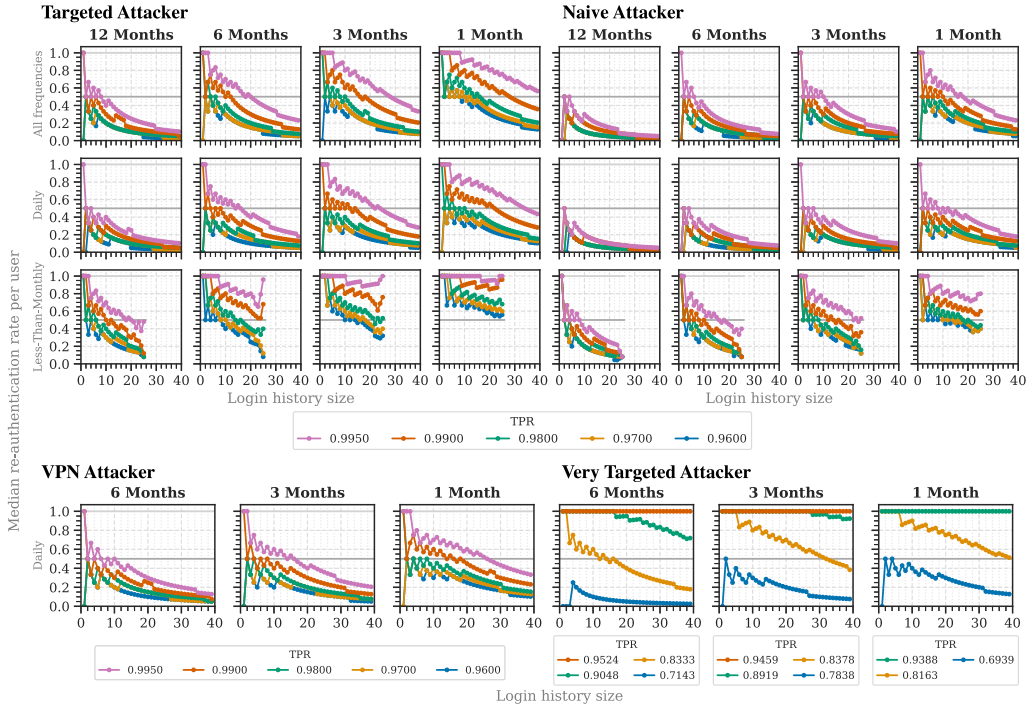
Fig. 7. RQ3: Median re-authentication rates by login frequencies and attacker model when minimizing the login history.

time-consuming and complex process, however. For this reason, we analyzed whether the RBA configuration process can be automated and improved with ML mechanisms to achieve a good RBA setup in a short time.

10.1 Dynamic Access Threshold

Current RBA literature considered the access threshold as a static, one-value component to distinguish between two risk categories [24, 51, 61, 69]. However, when analyzing the risk scores for legitimate users and attackers, we discovered that they declined with an increasing login history size. Therefore, for optimized usability and security, the access threshold can be set higher at the beginning and follow this decline with an increasing login history size. Based on our findings, we introduce a dynamic access threshold, which uses different access thresholds based on the user's login history size. To select appropriate threshold values, an ML-based approach can assist administrators in doing so.

Overview. Our ML-based approach works as follows (Figure 8). We first calculate the risk scores and the user's corresponding login history size for a number of legitimate login attempts. The ML model then learns the risk scores based on the login history size. As a result, the ML model can generate a risk score that is closest to most legitimate users, that is, an optimized access threshold. Assuming that legitimate users have lower risk scores than attackers, this can reduce the re-authentication count while not decreasing security to a large extent.

After training the ML model, the RBA algorithm extracts the ML-generated dynamic access threshold based on the login history of the user and uses it for the access decision.
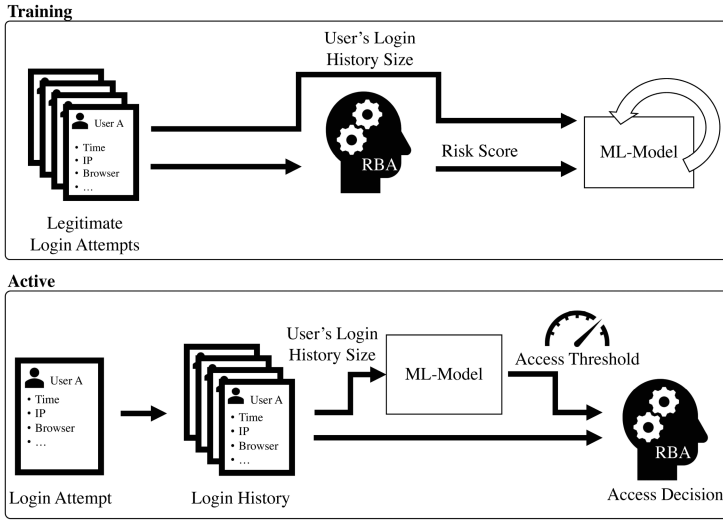
Fig. 8. Workflow of the ML-based RBA parameter optimization using the dynamic access threshold.

Access Threshold Calculation. The risk scores for legitimate users declined fast for the first few login history sizes and then followed a near-linear decline. To consider this progression over the login history size, we trained the dynamic access thresholds with regression models. We created and tested three methods to calculate the dynamic access thresholds. These were (i) Linear, a linear regression model to follow a linear progression; (ii) Poly, a polynomial regression model to follow a fast decline; and (iii) Hybrid, a hybrid of both models to follow the expected risk score development over the login history size.

The risk scores might contain outliers that increase the generated access thresholds and, thus, decrease the security properties. To counteract this, we cleaned the data from the highest 5% of scores. We based this threshold on the observed risk scores. RBA risk scores generated by our tested model cannot go lower than zero. To make sure that the generated dynamic access thresholds were valid, we used only those that were greater than zero. If this was not the case, we kept the access threshold of the previous login history size. To achieve the best possible security and adoption to the expected risk scores, the hybrid model took the lowest access threshold of both models.

10.2 Studies

We designed two studies to estimate the performance of the ML-enhanced RBA model in practice. We trained the model on sample slices of 100K consecutive legitimate logins, which were less than 1% of our dataset. As the login history size ranged between 1 and 5,972, we considered 100K as a reasonable size to include multiple samples of all sizes. We acknowledge that random sampling a training set from the complete dataset is often used in ML applications [31]. However, we assume that using consecutive logins is more realistic for RBA in practice, as we can sample logins that happened only in the past. Also, the available historical login data is often sparse in practice; thus, random sampling is likely not possible.

Study 1: TPR Stability. We first estimated the range of TPRs that each of the three ML models achieved in practice. We trained the models with multiple sample slices and calculated the TPRs for each of the attackers. We extracted the 100K sample slices in consecutive 500K steps from the dataset to approximate the performance over the whole dataset.

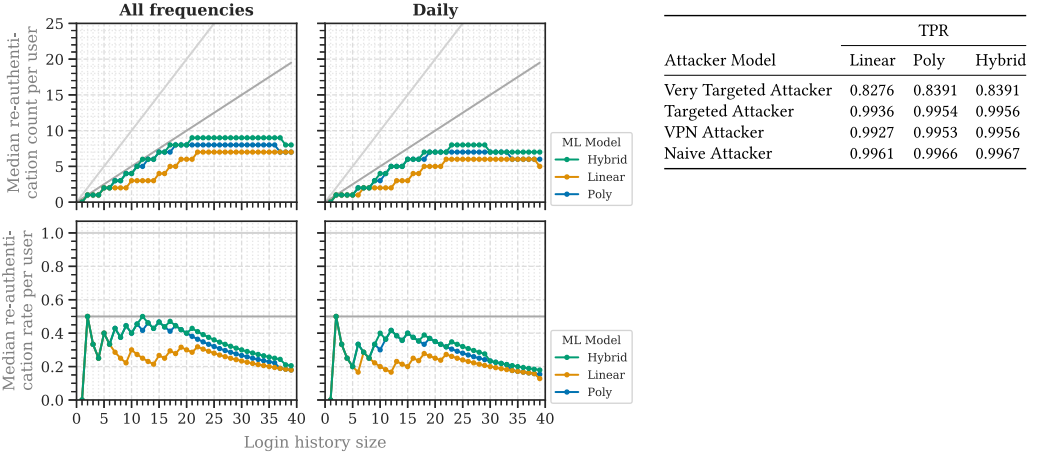|	TPR		
Attacker Model	Linear	Poly	Hybrid
Very Targeted Attacker	0.8276	0.8391	0.8391
Targeted Attacker	0.9936	0.9954	0.9956
VPN Attacker	0.9927	0.9953	0.9956
Naive Attacker	0.9961	0.9966	0.9967

Fig. 9. RQ4, Study 2: Median re-authentication counts and rates by login frequency (left) and achieved TPRs (right) for the dynamic access thresholds generated by three different ML models. The models were trained on the first 100K legitimate logins. The differences between all three models were significant.

After training, the Hybrid model achieved higher TPRs than the Linear and Poly model (see Table 7 in Appendix D). The TPRs of all models varied between 0.93 and 0.997 for naïve attackers, 0.86 and 0.996 for VPN and targeted attackers, and 0.65 and 0.87 for very targeted attackers. Therefore, the Hybrid model achieved a stronger security than the other two models.

Study 2: Re-authentication Rates. To identify how often the dynamic access threshold requests re-authentication in a practical example, we trained the ML models on the first 100K legitimate logins and calculated the re-authentication counts and rates. We selected the first logins to estimate the performance over the course of 1 year.

In contrast to the static access threshold, the dynamic access threshold never exceeded a 0.5 median re-authentication rate per user at TPRs higher than 0.995 (Figure 9). Therefore, the dynamic access threshold reduced re-authentication requests compared with the static threshold. However, the median re-authentication rate varied significantly between the different ML-models generating the dynamic access threshold. The Linear model requested significantly less re-authentication than the Poly and Hybrid model ($p \ll 0.0001$). The Hybrid model requested significantly more re-authentication than the Linear and Poly method ($p \ll 0.0001$).

10.3 Discussion

The dynamic access threshold performed better than the static threshold, as it achieved lower re-authentication rates at the same TPRs. For high usability, we recommend using the linear model, as its TPRs were similar to those of the other models but with significantly lower re-authentication rates.

The results show that ML-based parameter optimization can serve administrators as a basic setup to balance RBA's usability and security in little time. However, the achieved TPR varied based on the trained input, which is probably not desired by online service operators. This is why we consider the generated dynamic access thresholds more as a starting point for administrators. They can take these thresholds and adjust them to their needs, for example, reducing them to increase security and re-authentication count. They can also adjust the training set to drop fewer high-risk score outliers to increase the generated access thresholds in general.

11 PERFORMANCE (RQ5)

Using RBA with large global login history sizes can greatly increase risk score calculation time [69]. To be able to compute the results in reasonable time, we optimized the RBA algorithm. We describe our optimizations, including an analysis, in this section. The optimizations are also applicable to productive environments.

11.1 Algorithm Optimization

To calculate the probabilities in the risk score (see Section 3), the RBA model needs to execute multiple database queries to count the number of feature occurrences. The time for each query increases with the global login history size [69]. Therefore, calculating the risk score for one login attempt took around 5.6 s when using all 12.5M login history entries. Since such a delay is not feasible in a real-world scenario [58], we optimized the algorithm with hash tables [34], one for each feature. The hash table stored the number of occurrences per feature value as key value pairs: *feature value → number of occurrences*. After each successful login attempt, the hash table values were updated by increasing the feature value counts of those features belonging to the login attempt. Since this replaced the computation-intensive query with a simple addition, *current value* + 1, it reduced the computation time to a large degree. Calculating the risk score using hash tables took around 0.2 s, which was a 28x speedup.

We verified through partial calculations that the optimized risk score calculations were identical to the unoptimized version. We calculated both risk score types on multiple slices of 5K login attempts at random positions of the dataset and compared the results.

11.2 Analysis

The online service processed a median of more than 74K login attempts per day, with peaks over 150K around Christmas and New Year's (Figure 10). The peaks resulted from an increase in failed login attempts, also from identified attack IP addresses. Such high numbers may influence the RBA-dependent authentication time per user, especially with multiple login attempts in parallel.

To get an estimate of the expected authentication time per login attempt, we measured the risk score calculation time based on global login history size [69]. Due to a large calculation time, measuring both unoptimized and optimized RBA algorithms for all 12.5M login history sizes multiple times was not feasible. To consider the increasing calculation times, we measured the risk score calculation multiple times with global login histories that increased in 500K steps and calculated the linear regression line. As the authentication time increases linearly with the global login history size [69], a linear regression was feasible for our analysis. We measured on a server with Intel Xeon Gold 6130 processor (2.1 GHz, 64 cores), 480 GB SSD storage, and 64 GB RAM. We determined the effect sizes based on Cohen [14].

Without optimization, the regression yielded to $y = 58.7327 + 0.00044 \cdot x$ with a large effect size ($R^2 = 0.95$; f = 4.40; $p \ll 0.0001$), where y is the time in ms and x the global login history size. With hash table optimization, the regression line resulted in $y = 69.1459 + 0.00002 \cdot x$ and a large effect size ($R^2 = 0.99$; f = 9.25; $p \ll 0.0001$).

Authentication Time in Practice. We used the resulting regression lines and the login timestamps to calculate the estimated risk score calculation time ranges, that is, start and end of calculation. Based on their intersections, we identified the number of risk scores that were calculated at the same time (Figure 11). The number of concurrent risk score calculations reduced significantly when using the hash table ($p \ll 0.0001$).

We assume that one processor core can calculate one risk score at at time; thus, the authentication time might slow down with multiple login attempts at similar times. To estimate a realistic
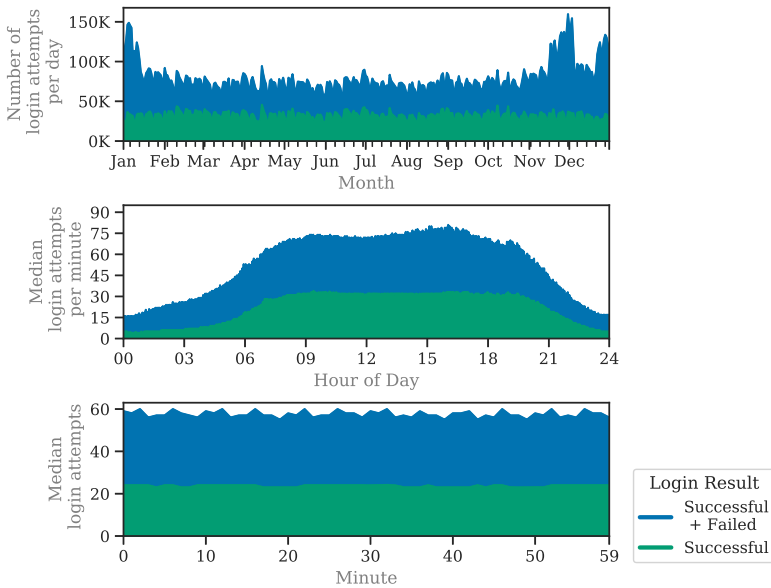
Fig. 10. Login attempts on the online service over the year (top), and on a daily (middle) and hourly basis (bottom). The service processed a median of 74.8K login attempts per day (SD: 18.6K), and 58 logins per minute (SD: 28).

Table 3. Memory usage for the Hash Table with All 12.5M Legitimate Logins

Feature	Storage	Feature	Storage
User ID	50.42 MB	IP Address	35.08 MB
ASN	0.12 MB	Country	0.003 MB
User Agent String	3.89 MB	Browser	0.05 MB
OS	0.01 MB	Device Type	0.0001 MB
Total	89.56 MB		

The hash table required 1% of the login history memory (8.28 GB).

authentication delay for our dataset, we calculated the authentication time for a server having 64 processor cores. In this case, the median authentication time was 193 ms (SD: 64 ms) with a hash table, and 3.2 s (SD: 2.77 s) without a hash table (see Figure 11).

Required Storage. The memory storage for our hash tables depends on the number of unique feature values in the dataset. In practice, users had an intersection of some feature values; thus, the required storage was much smaller than the full dataset. Also, the feature values were hashed in the table; thus, they required much less memory than the raw string values. As a result, the required memory for the hash table was very small, even after all recorded logins were added (Table 3).

11.3 Discussion

Not using a hash table greatly increased the authentication time. It increased even more with a large amount of login attempts. In the extreme case, when the concurrent login attempts exceeded
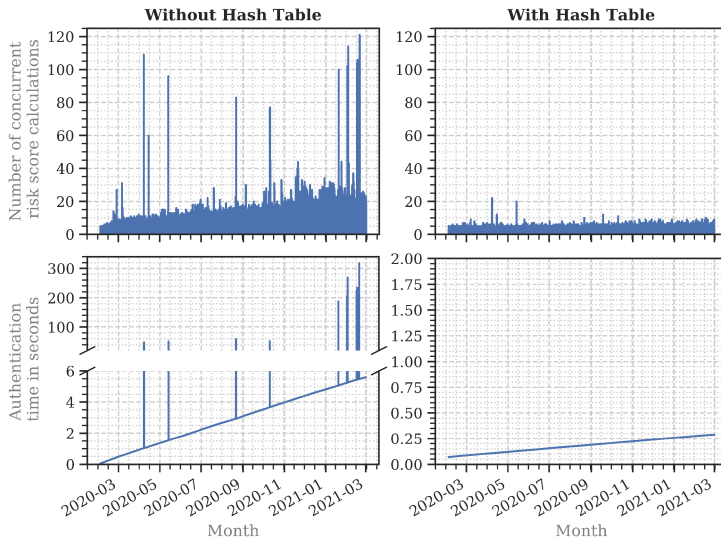
Fig. 11. Concurrent risk score calculations and expected authentication times within the observation period at the online service. We calculated the times for a 64 core processor using the regression lines representing the average authentication times per global login history size. Using a hash table significantly decreased the authentication time.

the processor core count, the authentication times were similar to a denial of service attack. This also shows that unoptimized RBA can be vulnerable to such attacks with little effort. Therefore, to significantly improve authentication performance on large-scale online services, we recommend using the hash table. Due to the low memory footprint, the hash table can be stored in RAM for fast performance (mean: 0.003 ms with hash table, 35 ms without hash table).[5] Using a relational database, for example, MariaDB, for storage would slow down the overall query time, but the hash table would still increase the performance (mean: 3.5 ms with, 8.9 s without).[4]

12 ROUND-TRIP TIME FEATURE (RQ6)

Wiefling et al. [69] proposed a server-originated RTT feature based on the WebSocket technology [36] to estimate device location. The server requests a data packet (ping frame) from the client and measures the time until the response (pong frame). Only devices near the server location can achieve low RTTs, which is why this is hard to spoof for attackers without high effort — they need access to a device physically located in the target area. Thus, this feature can help to determine whether the client's device location is really in the indicated region or spoofed by VPNs or proxies [1, 12]. Content Delivery Networks (CDNs) can even improve the reliability of RTT, as edge nodes close to the client's device are also considered for the measurement [69]. Concurrent and independent work confirmed this potential [33]. For most re-identification attacks with leaked data, the RTT is useless because it depends on the server location, which is widely distributed in practice. Therefore, the RTT also has potential to be a privacy-enhancing alternative to the sensitive IP address feature [73]. To investigate its potential as a location verifier and as a drop-in replacement for the IP address feature, we evaluated the RTT feature's abilities.

---

[5]Time to get a userid's entry count on the full dataset, that is, the largest delay possible in our use case scenario.

Table 4. Dunn-Bonferroni p values for the RTTs

(a) Countries

	BD	DE	MM	MY	NO	PK	SE	US
BD	-	-	-	**<0.0001**	**<0.0001**	**0.0074**	0.1686	-
DE	-	-	-	0.1381	-	-	-	-
MM	-	-	-	**0.0038**	-	-	0.1082	-
MY	**<0.0001**	0.1381	**0.0038**	-	**0.0410**	**<0.0001**	-	**<0.0001**
NO	**<0.0001**	-	-	**0.0410**	-	**<0.0001**	-	**<0.0001**
PK	**0.0074**	-	-	**<0.0001**	**<0.0001**	-	**0.0002**	-
SE	0.1686	-	0.1082	-	-	**0.0002**	-	**0.0005**
US	-	-	-	**<0.0001**	**<0.0001**	-	**0.0005**	-

(b) Counties of main country

	Innlandet	Nordland	Troms og Finnmark	Trøndelag	Vestfold og Telemark	Vestland	Viken
Innlandet	-	**0.0023**	**0.0052**	**0.0196**	-	0.1161	-
Nordland	**0.0023**	-	-	-	**0.0334**	-	**<0.0001**
Troms og Finnmark	**0.0052**	-	-	-	**0.0474**	-	**0.0003**
Trøndelag	**0.0196**	-	-	-	-	-	**0.0007**
Vestfold og Telemark	-	**0.0334**	**0.0474**	-	-	-	-
Vestland	0.1161	-	-	-	-	-	**0.0024**
Viken	-	**<0.0001**	**0.0003**	**0.0007**	-	**0.0024**	-

We omitted p values greater than 0.2 for readability. Bold: Significant.

The online service did not collect RTTs for all users during login. However, the online service offered mobile users the option to verify their mobile phone number when connected to the service's network provider. In this case, the online service sent a request to the mobile phone and the phone responded. The service measured the time (in ms) for the response, which can be considered an RTT variant. The RTT was recorded in 5.1M login attempts (2.5M successful, 2.6M failed), which were 17.6% of all login attempts.

12.1 RTT as a Location Verifier

We investigated the RTT's potential to verify locations that were given by the IP-based geolocation. In contrast to Wiefling et al. [69], the RTT was not measured five times during each login, so the RTTs may largely vary due to mobile connectivity. To mitigate this effect, we considered only those cities with at least 10 measurements. We describe our studies to measure the RTT's ability to determine countries and regions next.

RTT per Country. We first analyzed the RTT differences between countries, calculating the median RTT per city to determine a stable average value for each of them. We then calculated the median RTT of each country's cities to determine their main tendencies. We assume that attackers may try to spoof the location via a VPN connection; thus, the measured RTT might not match their IP-indicated region. Therefore, we considered successful logins separately to compare the results at a higher level of trust, as it would be the case with RBA, that is, only successful logins are stored.

Figure 12 shows the results. Most countries had a higher median RTT than the online service's main country. Furthermore, the differences in the RTTs were significant for some countries (Table 4(a)). All RTTs had a median standard deviation of 290 ms per country. Following that, the RTTs can be narrowed down to a number of countries.

RTT per Region. Our dataset contained a large number of RTTs with an IP geolocation inside the online service's home country. Thus, we further analyzed whether it is possible for the RTT
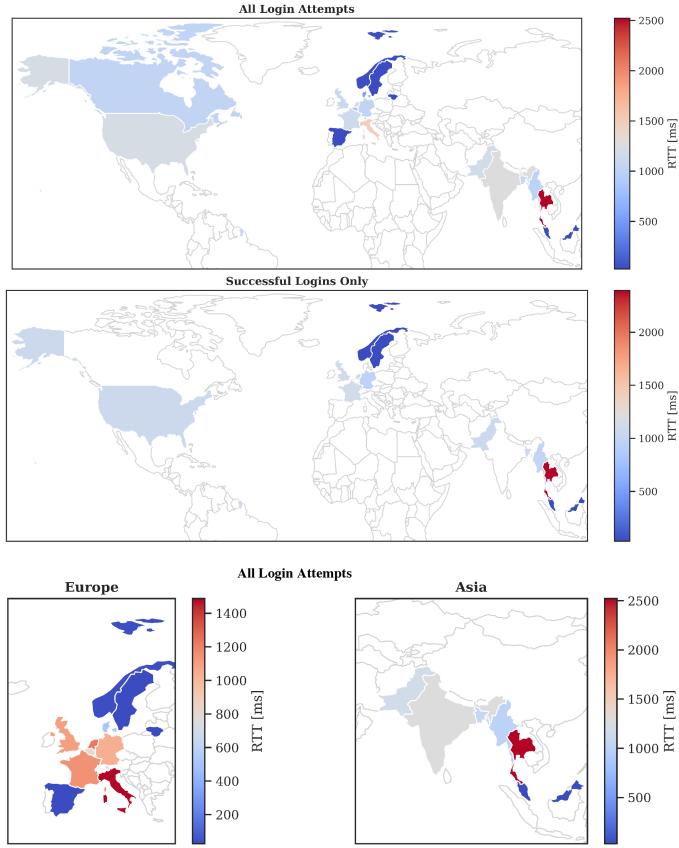
Fig. 12. Median RTTs by country. The RTTs were measured on mobile devices.

to verify regions in this country. To consider a high level of trust, we included only successful logins. As in the previous study, we first calculated the median RTT per city. We then clustered the RTTs to each of the country's counties. The results show significant differences in the RTTs of the counties (Table 4(b)). Therefore, the RTT can also narrow down regions of a country.

12.2   RTT as an IP Address Replacement

Our previous results showed RTT's potential to verify IP-based geolocations. One step further would be to remove the full IP address from the feature set completely. Replacing the IP address with the RTT would protect users' privacy in terms of data breaches [73].

Based on the ideas of Wiefling et al. [73], we derived the ASN and country only from the IP address and then replaced the IP address with the RTT. The RTT had the identical weighting as the IP address for a fair comparison. We kept and evaluated the RTT values as they were recorded from each login attempt to test the RBA behavior under real-world conditions.

To compare the risk scores between IP address and RTT, we included only the login history data that had both features: 2.5M legitimate logins. Based on the dataset, we calculated the different risk scores for RBA when using IP address and RTT features. Similar to related work [69], we also rounded the RTT to the nearest 5, 10, and 50 ms to observe the RBA behavior at different levels of RTT granularity. We tested the naïve, VPN, and targeted attackers. The available RTT data for the very targeted attacker was sparse, which is why we were not able to test this attacker here.
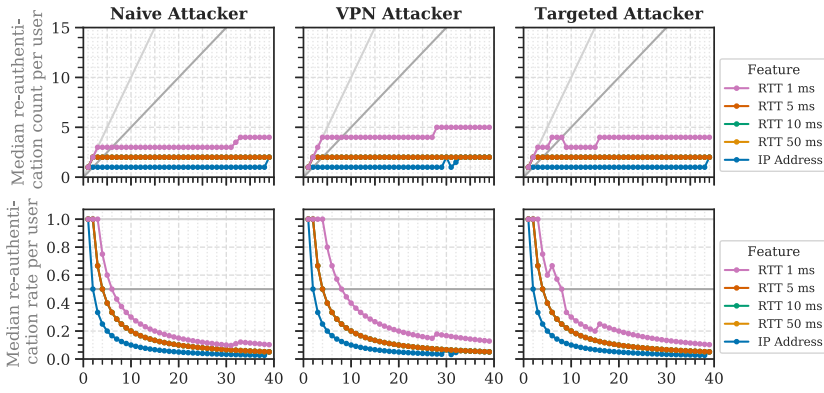
Fig. 13. Direct comparison of the RTT and IP address features regarding their re-authentication counts and rates (TPR: 0.999). The results for 5 ms, 10 ms, and 50 ms were identical.

Table 5. Risk Score Relations for the IP Address and
RTT Features Based on the Three Attacker Models (TPR: 0.999)

		RTT			
Attacker Model	IP Address	1 ms	5 ms	10 ms	50 ms
Targeted Attacker	70.25	68.53	68.22	67.87	65.64
VPN Attacker	61.90	60.39	60.12	59.81	57.83
Naïve Attacker	54.43	53.10	52.87	52.61	50.88

The median re-authentication rates of all features were zero until TPRs of 0.995. At a TPR of 0.999, the median re-authentication rates of the RTT slightly overtook those of the IP address for all attacker models (Figure 13).

To estimate the features' ability to distinguish between attackers and legitimate users, we also calculated the risk score relation (RSR) [69, 73] for them. The RSR is defined as the relation between the mean risk scores for attackers and legitimate users:

$$RSR = \frac{mean\ attacker\ risk\ score}{mean\ legitimate\ risk\ score} \qquad (3)$$

The higher the RSR, the better attackers can be distinguished from legitimate users. Table 5 shows the results for the features and attacker models. In general, the IP address achieved a slightly higher RSR than the RTT variations.

12.3 Discussion

The results, especially those of successful logins, highlight the tendency that most RTTs were higher than those of the online service's main region. Therefore, the RTT can be considered a strong factor to verify locations indicated by the IP address, reflecting and extending previous findings [1, 51, 69]. Beyond that, the RTT can also identify users when used as a replacement for the IP address, which is a new finding. The RTT was measured only on mobile devices; thus, they contain larger variations due to mobile connectivity. We expect that these variations would be smaller on desktop devices connected to home or company networks. Therefore, it is possible that the study results are more negative than under real-world conditions involving desktop devices.

The RTTs for Malaysia were very low. This is due to the fact that the online service provider operated mobile networks inside this country. Therefore, they were likely authenticated via their local mobile network. An optimized mobile connection was also likely true for Spain, where the RTTs were very low only in popular travel destinations for people of the online service's home country. In practice, this corresponds to behavior similar to CDNs; thus, users inside these regions can be verified with high confidence.

The IP address had higher RSRs, but the RSR differences to the RTT were low. The RSR differences between the 1-ms and 5-ms RTT variations were small. The latter, however, achieved a lower re-authentication rate. This rate was identical to those of 10 and 50 ms, which had lower RSRs. Therefore, an RTT of 5 ms provides good usability and privacy, and is preferable to 10 and 50 ms in terms of security in this dataset and TPR context.

Using the RTT as a feature is becoming more relevant, as some tech companies have started testing VPN and proxy solutions as a privacy enhancement for their users [39, 45]. Considering a relatively stable delay between VPN server and online service, we assume that the RTT can also be used to determine the distance between a VPN server and the client device. This, however, requires further study.

For the RTT feature comparison, we had to recalculate the attacker models based on the reduced RTT dataset. The majority of failed login attempts did not include an RTT. We expect that most attackers did not expect the mobile number authentication method to be successful and, thus, did not try it. We assume that this is why the re-authentication counts per TPR were lower than when using the full dataset. Nevertheless, our results can still give indications on the effectiveness of using the RTT feature.

13   LIMITATIONS

The study results presented in this article are representative of a certain country (Norway) and online service type (SSO of telecommunication services). They do not represent all online services worldwide; rather, they show an example of a typical large-scale online service with sensitive data involved.

Since the dataset corresponds to the global browser market share (see Section 4.1), our results can help to apply RBA globally. Nevertheless, cultural differences in some countries may lead to different login patterns. For example, people in rural Myanmar rely on mobile shop staff to set up online accounts for them [21]. In this case, online services can notice that many different user accounts were created over the same Internet connection. From a Western perspective, this behavior may seem suspicious, but it is normal in these areas. Therefore, it makes sense for service owners to tailor their RBA systems to cultural conditions in the target locations to achieve a high user acceptance.

The RTT was measured only once per login attempt. We assume that the RSRs for the RTT would improve with multiple RTT measurements per login attempt, as shown in Wiefling et al. [69] with five RTT measurements during the login process.

14   RELATED WORK

With our contributions, we extend the related work in various aspects, which we discuss in this section.

Freeman et al. [24] tested their RBA model in a case study using 300K legitimate logins collected during 6 months on the popular large-scale online service LinkedIn. However, they did not provide an analysis of RBA characteristics in practice. Also, for such a large-scale online service, the sample was rather small. To provide a more realistic estimate, we analyzed RBA characteristics using 12.5M legitimate logins collected in more than one year. Wiefling et al. [69] tested the model by

Freeman et al. on a small online service with 9,555 legitimate logins collected over 1.8 years. Their and our results confirmed that IP address, user agent string, and RTT are useful RBA features. Our work also extended the analysis for different login frequencies. Wiefling et al. [73] proposed and tested privacy enhancements for RBA. Due to their small dataset, they were not able to test their proposed login history minimization approach. We filled this research gap and tested this approach on a large-scale online service.

Alaca and van Oorschot [5] evaluated a list of features regarding their ability to identify users. Regarding distinguishing information, they rated the IP address feature higher than the RTT. They noted, however, that the latter required further study. There were different RTT implementations for RBA proposed in concurrent and independent work. Rivera et al. [51] proposed a client-originated version. Wiefling et al. [69] proposed a server-initiated solution that they considered harder to spoof for attackers. The studies in both works showed the RTT's potential to verify countries [51] and users [69]. Our work further showed RTT's potential to verify regions and to replace the IP address as a feature. Campobasso and Allodi [12] studied a criminal infrastructure that used malware-infected devices to bypass RBA. The RTT feature can detect such attempts, as the infrastructure used SOCKS5 proxies.

Doerfler et al. [19] evaluated RBA re-authentication challenges on a sample of 1.2M legitimate Google users. They stated that their RBA implementation was able to block 99.99% of automated account takeover attempts and 92% of phishing attacks. Our study results with 3.3M legitimate users reflect their findings.

Besides RBA, there are other related methods that aim to protect user accounts against automated attacks. CAPTCHAs [66] are a widely used measure to block bot traffic by presenting a challenge that, in contrast to bots, is solvable by humans with a lower error rate. Some RBA systems also present CAPTCHAs on some occasions to increase friction only for bot-like traffic [71]. However, solving CAPTCHAs is still error prone for humans [13, 50]; thus, achieving usable CAPTCHAs can be considered a challenge. Bots can also bypass them by proxying the CAPTCHA challenge to real humans [71] or machine learning algorithms [62]. These bypass methods are scalable in practice. RBA, in contrast, depends on the specific user. Thus, proxying cannot scale when RBA uses appropriate features, for example, the RTT.

Another method is attribute-based authentication using attribute-based credentials (ABCs) [11]. In this case, users authenticate to an online service by providing a verified attribute, for example, their age, to prove that they are over 18 years old. The verification is done by a trusted authority and the information presented to the online service can be done using pseudonyms. Therefore, online services receive only the information that is required for their service, for example, that the user is over 18, without getting personal details. ABCs can increase user privacy but are independent from RBA, as the latter do not require verified personal details. However, ABCs can be considered to be a re-authentication challenge in high-risk environments in which users expect increased security, for example, online shopping or online banking [52, 70].

15 CONCLUSION AND SUGGESTIONS

Online services are a popular attack target [4]. Stolen passwords used, for example, in credential stuffing attacks pose new security challenges to service developers, operators, and owners [19]. As these attacks have increased over the years [3, 4], RBA has become an important measure for online services to protect their users. NIST, NCSC, and ACSC recommend RBA [7, 27, 40]; some have recommended it for more than 3 years. Still, current literature does not provide any insights on implementing, configuring, and operating RBA on these types of online services. These insights are important, though, to achieve RBA's security gain while balancing usability and privacy. Only

a fraction of work exists that focuses on RBA algorithms or studies, all of which is on small online services. We studied RBA characteristics on a large-scale online service to close this research gap.

Our results, based on our dataset of a large online service, confirm previous results, based on a small online service, that RBA rarely requests re-authentication in practice, even when blocking more than 99% of targeted attackers. As new results, our evaluation furthermore shows that RBA can even detect a high number of very targeted account takeover attempts. The RBA behavior also significantly depends on the users' login frequencies. For daily-use websites, RBA can achieve very high security with few re-authentication requests. For monthly-use websites, the percentage of blocked attackers needs to be lowered down to around 2% to 3% to achieve a similar usability for targeted attackers. Still, a high percentage of attackers would be blocked for all user types. Therefore, we consider RBA an effective measure to protect users from the majority of attacks involving stolen passwords. In addition, RTT proved to be a promising RBA feature to verify regions and to identify users while increasing user privacy.

RBA needs to be carefully evaluated for and adapted to each individual use case scenario. Based on our findings, RBA system designers should consider the following to find the best configuration for their environment: (i) The amount of minimizing the login history needs to be carefully adjusted to users' login frequencies. Online services with a daily user base should keep at least 6 months of login history for a stable setup that blocks targeted attackers. When blocking naïve attackers only, which represented 97% of all identified incidents at the online service, reducing to 3 months is possible. (ii) Attack data should be used with caution or not at all, as these can greatly decrease usability and security properties. (iii) To achieve an optimal basic configuration in a short period of time, administrators can use the introduced ML-based RBA parameter optimization. For further adjustment, they can increase or reduce the resulting access thresholds to decrease or increase the security and average re-authentication count, respectively. (iv) Non-optimized RBA is potentially vulnerable to denial of service when multiple users log in at similar times. To mitigate this, and to considerably shorten authentication time, implementation of the RBA algorithm needs to be carefully designed using efficient data structures such as hash tables.

Our results and the synthesized dataset support online services to gain a widespread understanding of RBA. In future work, we will use these insights to create an open-source RBA solution that significantly improves usability, security, and privacy compared with the only open solution currently available, OpenAM [42]. The dataset will be used as testing data to evaluate our solution.

APPENDICES
A SYNTHESIZED DATASET

Our synthesized dataset can be downloaded at https://github.com/das-group/rba-dataset. As this article introduces our open dataset, we ask researchers to cite this article whenever they publish their own results obtained from adopting our dataset. For broad compatibility, we provide the data as a CSV file. Table 6 shows the available features in the synthesized dataset and the data types. We kept the feature values in readable notation to facilitate understanding. The values may need to be converted to more efficient data types to speed up risk score calculation. For instance, the IP address could be converted to its integer value. Other string-based features could be hashed and converted to integer values as well.

Common synthetic data generation methods rely on learning the real values of a dataset, which, however, does not provide a strong privacy guarantee [57]. Therefore, we used our own generation method that was tailored to the use in RBA models employed by the majority of online services in the wild [24, 42, 69]. We created the synthetic dataset as follows. As common RBA models rely only on the number of occurrences of categorical data, we did not have to reuse any of the original

Table 6. Available Features in the Synthesized Dataset

Feature	Data Type	Range or Example
IP Address	String	0.0.0.0 - 255.255.255.255
Country	String	US
Region	String	New York
City	String	Rochester
ASN	Integer	0 - 65535
User Agent String	String	Mozilla/5.0 (Windows NT 10.0; Win64; . . .
OS Name + Version	String	Windows 10
Browser Name + Version	String	Chrome 70.0.3538
Device Type	String	(mobile, desktop, tablet, bot, unknown)
User ID	Integer	[Random pseudonym]
Login Timestamp	Integer	[64 Bit timestamp]
RTT [ms]	Integer	1 - 8600000
Login Successful	Boolean	(true, false)
Is Attack IP	Boolean	(true, false)
Is Account Takeover	Boolean	(true, false)

feature values. We transformed all unique dataset values to a random pseudonymous identifier and added randomness to the timely order to keep the statistical relations between different feature values. Then, we randomly assigned feature values from a Location-to-ASN-to-IP dataset in backwards hierarchy (i.e., Country → Region → City → ASN → IP address). If the original IP address was used in an attack, we made sure that the randomly assigned IP address was an attack IP address as well. We repeated the same procedure for the user agent strings. Depending on the synthesized geolocation data and the login success status, we randomly drew the RTTs for the corresponding entries to ensure that the RTT results were realistic. Finally, we generated the login timestamps with random noise following the distributions and timely order (see Figure 10 in Section 11).

We verified the synthesized dataset by recalculating the study results from this article. As our generation process does not reuse feature values from the original dataset, the synthesized dataset should not provide any personal identifiable information. Nevertheless, we evaluated the privacy attributes of the synthesized dataset using Stadler and Oprisanu [56] and other comparable frameworks.

B   FULL RESULTS: EVALUATING RBA IN PRACTICE (RQ1)
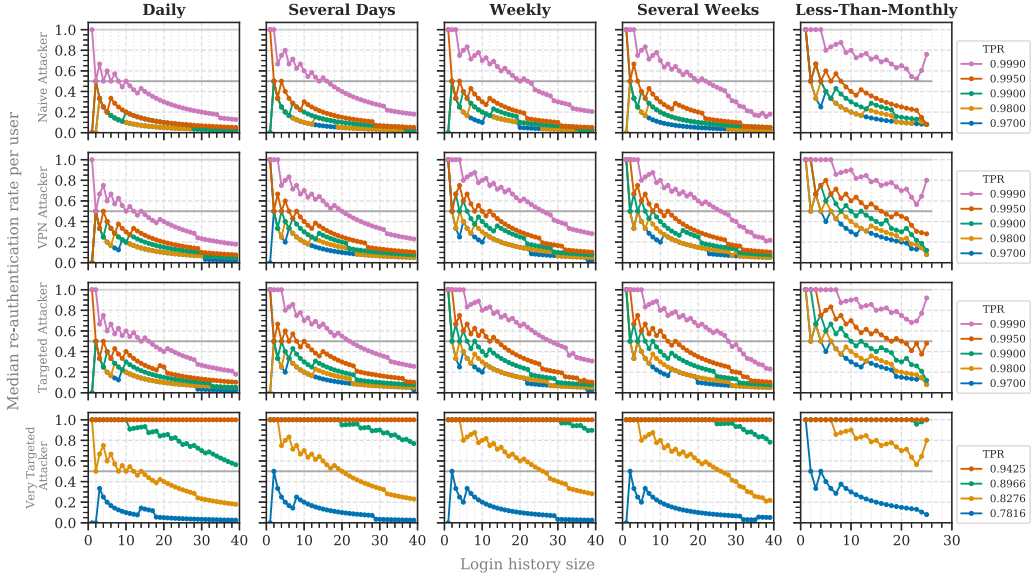

Fig. 14. Median re-authentication rates based on the login frequency and attacker model.

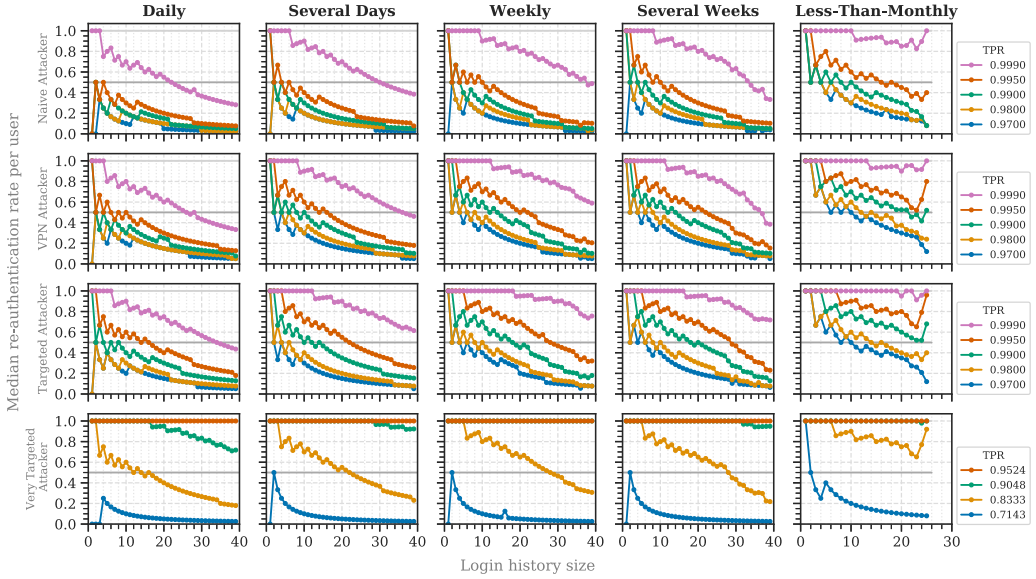## C   FULL RESULTS: LOGIN HISTORY MINIMIZATION (RQ3)

Fig. 15. Median re-authentication rates based on the login frequency and attacker model (minimized 6 months).
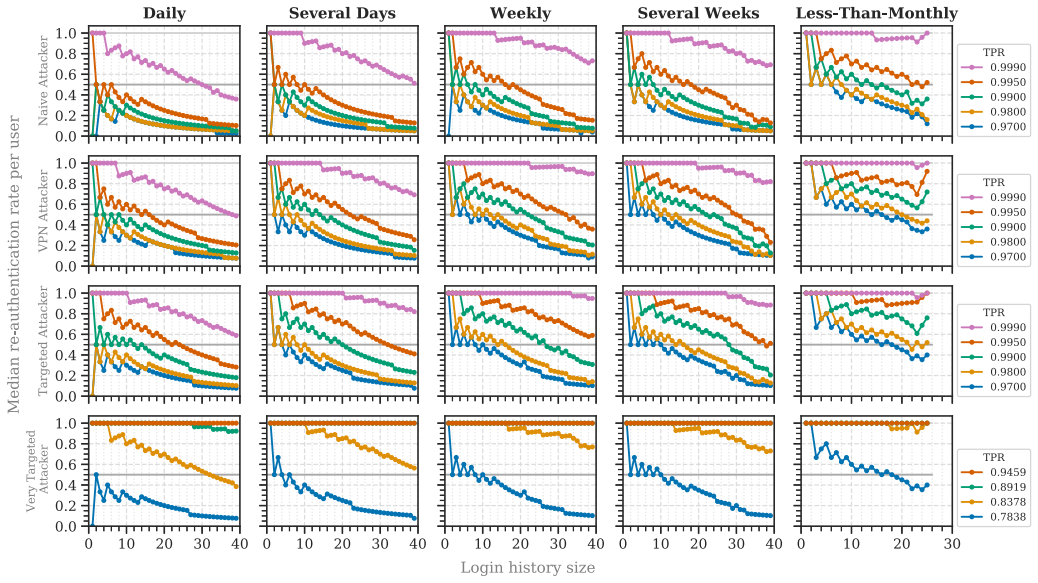
Fig. 16. Median re-authentication rates based on the login frequency and attacker model (minimized 3 months).
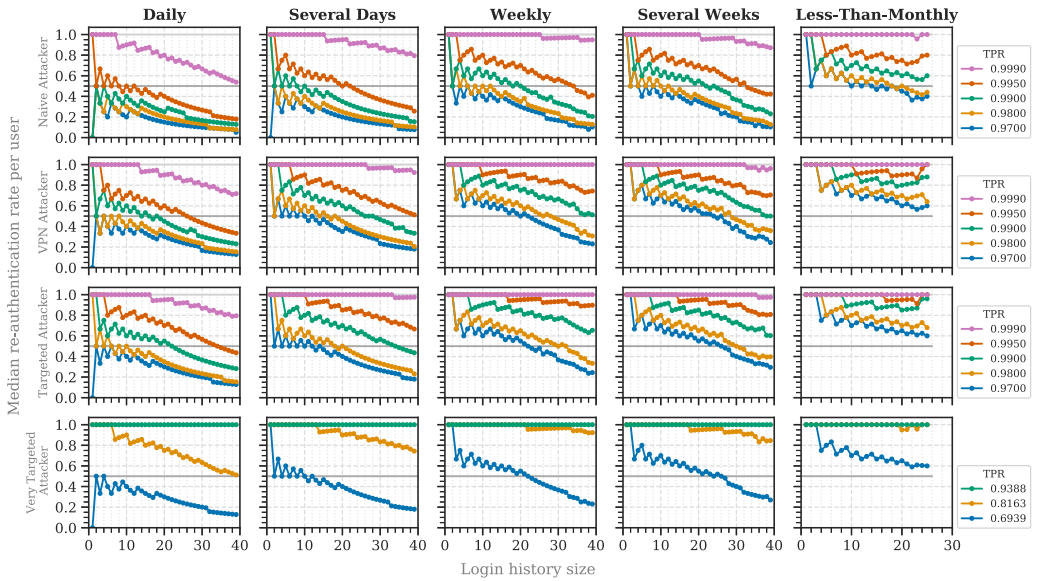

Fig. 17. Median re-authentication rates based on the login frequency and attacker model (minimized 1 month).

D   ML-BASED RBA PARAMETER OPTIMIZATION (RQ4)

Table 7. RQ4, Study 1: Achieved TPRs
for the ML-Generated Dynamic
Access Thresholds in Practice

ML Model	Range	Median	SD
Naïve Attacker			
Linear	0.9319-0.9961	0.9713	0.0173
Poly	0.9312-0.9966	0.9718	0.0179
Hybrid	0.9338-0.9967	0.9730	0.0170
VPN Attacker			
Linear	0.8646-0.9929	0.9250	0.0319
Poly	0.8640-0.9954	0.9249	0.0335
Hybrid	0.8671-0.9957	0.9278	0.0327
Targeted Attacker			
Linear	0.8646-0.9929	0.9283	0.0324
Poly	0.8646-0.9949	0.9290	0.0338
Hybrid	0.8670-0.9951	0.9316	0.0331
Very Targeted Attacker			
Linear	0.6782-0.8506	0.7701	0.0520
Poly	0.6552-0.8621	0.7931	0.0523
Hybrid	0.6782-0.8736	0.7931	0.0500

We trained the ML models on 24
consecutive slices of the dataset. The
hybrid method achieved higher TPRs
than the linear and poly models.

E   SIMPLE MODEL EVALUATION

To provide a baseline evaluation, we repeated the study of RQ1a and RQ1c (see Section 7) with
the SIMPLE model used in the open-source SSO solution OpenAM [42] (see Section 2). The model
checks the feature values for an exact match in the user's login history. In contrast to the Freeman
et al. model [24], the SIMPLE model does not allow taking attack data into account. As in the other
studies, the model used the IP address and user agent string as features. The results show that the
achievable TPRs were very coarse grained, for example, the TPR for targeted attackers dropped
from 0.9552 to 0.5295 when the access threshold was lowered by a tiny fraction to the next possible
level (Figure 18). This confirms previous findings [69]. Also, the high end of TPRs varied largely
across the different attackers, for example, 0.9816 for naïve attackers and 0.9494 for VPN attackers.
This uncertainty makes the SIMPLE model difficult to configure for large-scale online services.
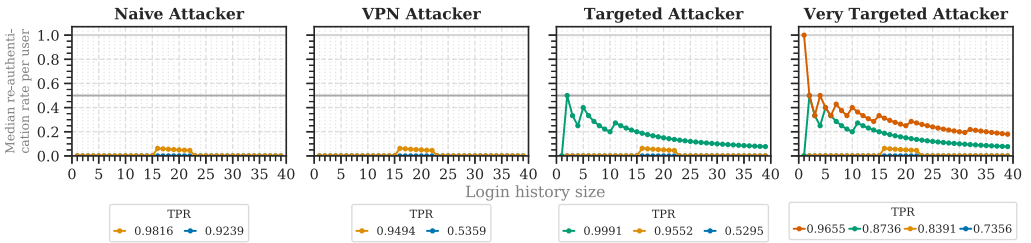

Fig. 18. Median re-authentication rates for the four attacker models when using the SIMPLE model. The
model's risk score is coarse grained. Therefore, the TPRs show the configuration steps that are possible
under these conditions. For orientation, we added the baseline for 2FA (light-gray line), and the stable setup
threshold (gray line).

ACKNOWLEDGMENTS

REFERENCES

[1] Abdelrahman Abdou and Paul C. Van Oorschot. 2018. Secure client and server geolocation over the internet. *;login: Spring 2018* 43, 1 (2018), 7.

[2] Alejandro Acien, Aythami Morales, Ruben Vera-Rodriguez, Julian Fierrez, and John V. Monaco. 2020. TypeNet: Scaling up keystroke biometrics. In *2020 IEEE International Joint Conference on Biometrics (Houston, TX, USA) (IJCB'20)*. IEEE. https://doi.org/10.1109/IJCB48548.2020.9304908

[3] Akamai. 2019. Credential stuffing: Attacks and economies. *[state of the internet] / security* 5, Special Media Edition (April 2019). Retrieved July 7, 2022 from https://web.archive.org/web/20210824114851/https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-credential-stuffing-attacks-and-economies-report-2019.pdf.

[4] Akamai. 2020. Loyalty for sale – retail and hospitality fraud. *[state of the internet] / security* 6, 3 (Oct. 2020). Retrieved July 7, 2022 from https://web.archive.org/web/20201101013317/https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/soti-security-loyalty-for-sale-retail-and-hospitality-fraud-report-2020.pdf.

[5] Furkan Alaca and P. C. van Oorschot. 2016. Device fingerprinting for augmenting web authentication: Classification and analysis of methods. In *32nd Annual Computer Security Applications Conference (Los Angeles, CA, USA) (ACSAC'16)*. ACM, 289–301. https://doi.org/10.1145/2991079.2991091

[6] Nampoina Andriamilanto, Tristan Allard, and Gaëtan Le Guelvouit. 2021. "Guess who?" Large-scale data-centric study of the adequacy of browser fingerprints for web authentication. In *Innovative Mobile and Internet Services in Ubiquitous Computing*. Springer, 161–172. https://doi.org/10.1007/978-3-030-50399-4_16

[7] Australian Cyber Security Centre. 2021. *Australian Government Information Security Manual*. Technical Report. Retrieved July 7, 2022 from https://web.archive.org/web/20210830131917/https://www.cyber.gov.au/sites/default/files/2021-06/01.%20ISM%20-%20Using%20the%20Australian%20Government%20Information%20Security%20Manual%20(June%202021).pdf.

[8] Joseph R. Biden Jr. 2021. Executive order on improving the nation's cybersecurity. *The White House* (May 2021). Retrieved July 7, 2022 from https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/.

[9] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. IEEE, 553–567. https://doi.org/10.1109/SP.2012.44

[10] Andriy Burkov. 2020. *Machine Learning Engineering*. True Positive Inc.

[11] Jan Camenisch, Maria Dubovitskaya, Anja Lehmann, Gregory Neven, Christian Paquin, and Franz-Stefan Preiss. 2013. Concepts and languages for privacy-preserving attribute-based authentication. In *Third IFIP WG 11.6 Working Conference on Policies and Research in Identity Management (London, UK) (IDMAN'13)*. Springer, 34–52. https://doi.org/10.1007/978-3-642-37282-7_4

[12] Michele Campobasso and Luca Allodi. 2020. Impersonation-as-a-service: Characterizing the emerging criminal infrastructure for user impersonation at scale. In *2020 ACM SIGSAC Conference on Computer and Communications Security (CCS'20)*. 1665–1680. https://doi.org/10.1145/3372297.3417892

[13] Kapil Chalil Madathil, Joel S. Greenstein, and Kristin Horan. 2019. Empirical studies to investigate the usability of text- and image-based CAPTCHAs. *International Journal of Industrial Ergonomics* 69 (Jan. 2019), 200–208. https://doi.org/10.1016/j.ergon.2018.12.003

[14] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). L. Erlbaum Associates.

[15] Heather Crawford and Karen Renaud. 2014. Understanding user perceptions of transparent authentication on a mobile device. *Journal of Trust Management* (June 2014). https://doi.org/10.1186/2196-064X-1-7

[16] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The tangled web of password reuse. In *2014 Network and Distributed System Security Symposium (San Diego, CA, USA) (NDSS'14)*. Internet Society. https://doi.org/10.14722/ndss.2014.23357

[17] Debayan Deb, Arun Ross, Anil K. Jain, Kwaku Prakah-Asante, and K. Venkatesh Prasad. 2019. Actions speak louder than (pass)words: Passive authentication of smartphone users via deep temporal features. In *2019 International Conference on Biometrics (Crete, Greece) (ICB'19)*. IEEE. https://doi.org/10.1109/ICB45273.2019.8987433

[18] Nebojsa Djosic, Bojan Nokovic, and Salah Sharieh. 2020. Machine learning in action: Securing IAM API by risk authentication decision engine. In *2020 IEEE Conference on Communications and Network Security (Avignon, France) (CNS'20)*. IEEE. https://doi.org/10.1109/CNS48642.2020.9162317

[19] Periwinkle Doerfler, Kurt Thomas, Maija Marincenko, Juri Ranieri, Yu Jiang, Angelika Moscicki, and Damon McCoy. 2019. Evaluating login challenges as a defense against account takeover. In *The World Wide Web Conference (San Francisco, CA, USA) (WWW'19)*. ACM, 372–382. https://doi.org/10.1145/3308558.3313481

[20] Jonathan Dutson, Danny Allen, Dennis Eggett, and Kent Seamons. 2019. "Don't punish all of us": Measuring user attitudes about two-factor authentication. In *4th European Workshop on Usable Security (Stockholm, Sweden) (EuroUSEC'19)*. https://doi.org/10.1109/EuroSPW.2019.00020

[21] Rainer Einzenberger. 2016. "If it's on the Internet it must be right": An interview with Myanmar ICT for development organisation on the use of the Internet and social media in Myanmar. *Austrian Journal of South-East Asian Studies* 9, 2 (Dec. 2016), 301–310. https://doi.org/10.14764/10.ASEAS-2016.2-9

[22] European Union. 2016. General data protection regulation. Regulation (EU) 2016/679. https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[23] FireHOL. 2021. All Cybercrime IP Feeds. (May 2021). Retrieved July 7, 2022 from http://iplists.firehol.org/?ipset=firehol_level4.

[24] David Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. 2016. Who are you? A statistical approach to measuring user authenticity. In *Network and Distributed System Security Symposium (San Diego, CA, USA) (NDSS'16)*. Internet Society. https://doi.org/10.14722/ndss.2016.23240

[25] Google. 2018. 1.5 Billion Users and Counting. Thank You. (Oct. 2018). Retrieved July 7, 2022 from https://web.archive.org/web/20211030190958/https://twitter.com/gmail/status/1055806807174725633.

[26] Google. 2021. Making Sign-In Safer and More Convenient. (Oct. 2021). Retrieved July 7, 2022 from https://web.archive.org/web/20211006012012/https://blog.google/technology/safety-security/making-sign-safer-and-more-convenient/.

[27] Paul A. Grassi, James L. Fenton, Elaine M. Newton, Ray A. Perlner, Andrew R. Regenscheid, William E. Burr, Justin P. Richer, Naomi B. Lefkovitz, Jamie M. Danker, Yee-Yin Choong, Kristen K. Greene, and Mary F. Theofanos. 2017. *Digital Identity Guidelines: Authentication and Lifecycle Management*. Technical Report NIST SP 800-63b. National Institute of Standards and Technology, Gaithersburg, MD. Retrieved July 7, 2022 from https://doi.org/10.6028/NIST.SP.800-63b

[28] Morey J. Haber. 2020. Attack vectors. Apress, Berkeley, CA, USA, 65–85. https://doi.org/10.1007/978-1-4842-5914-6_4

[29] Have I. Been Pwned. 2022. Pwned websites. (March 2022). Retrieved July 7, 2022 from https://haveibeenpwned.com/PwnedWebsites/.

[30] Adam Hurkała and Jarosław Hurkała. 2014. Architecture of context-risk-aware authentication system for web environments. In *Third International Conference on Informatics Engineering and Information Science (Lodz, Poland) (ICIEIS'14)*.

[31] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 103. Springer. https://doi.org/10.1007/978-1-4614-7138-7

[32] Hassan Khan, Urs Hengartner, and Daniel Vogel. 2015. Usability and security perceptions of implicit authentication: Convenient, secure, sometimes annoying. In *Eleventh Symposium On Usable Privacy and Security (Ottawa, Canada) (SOUPS'15)*. USENIX Association, 225–239.

[33] Katharina Kohls and Claudia Diaz. 2022. VerLoc: Verifiable localization in decentralized systems. In *31st USENIX Security Symposium (Boston, MA, USA) (USENIX Security'22)*. USENIX Association. Retrieved July 7, 2022 from https://www.usenix.org/conference/usenixsecurity22/presentation/kohls.

[34] Dapeng Liu, Zengdi Cui, Shaochun Xu, and Huafu Liu. 2014. An empirical study on the performance of hash table. In *13th International Conference on Computer and Information Science (ICIS'14)*. IEEE, 477–484. https://doi.org/10.1109/ICIS.2014.6912180

[35] Peter Mayer, Yixin Zou, Florian Schaub, and Adam J. Aviv. 2021. "Now I'm a bit angry:" Individuals' awareness, perception, and responses to data breaches that affected them. In *30th USENIX Security Symposium (USENIX Security'21)*. USENIX Association.

[36] Alexey Melnikov and Ian Fette. 2011. *The WebSocket Protocol*. Number 6455. Retrieved July 7, 2022 from https://doi.org/10.17487/RFC6455

[37] Grzergor Milka. 2018. Anatomy of account takeover. In *Enigma 2018 (Santa Clara, CA, USA)*. USENIX Association. https://www.usenix.org/node/208154.

[38] M. Misbahuddin, B. S. Bindhumadhava, and B. Dheeptha. 2017. Design of a risk based authentication system using machine learning techniques. In *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (San Francisco, USA) (SmartWorld)*. IEEE. https://doi.org/10.1109/UIC-ATC.2017.8397628

[39] Mozilla. 2021. Mozilla VPN. (July 2021). https://www.mozilla.org/en-US/products/vpn/.

[40] National Cyber Security Centre. 2018. *Cloud security guidance: 10, Identity and authentication*. Technical Report. Retrieved July 7, 2022 from https://www.ncsc.gov.uk/collection/cloud-security/implementing-the-cloud-security-principles/identity-and-authentication.

[41] Lily Hay Newman. 2021. Facebook Will Force More At-Risk Accounts to Use Two-Factor. (Dec. 2021). Retrieved July 7, 2022 from https://web.archive.org/web/20211212185008/https://www.wired.com/story/facebook-protect-two-factor-authentication-requirement/.

Pump Up Password Security! Evaluating and Enhancing RBA

6:35

[42] Open Identity Platform. 2016. OpenAM: Adaptive Authentication Module. (Aug. 2016). Retrieved July 7, 2022 from https://github.com/OpenIdentityPlatform/OpenAM/blob/master/openam-authentication/openam-auth-adaptive/src/main/java/org/forgerock/openam/authentication/modules/adaptive/Adaptive.java.

[43] Jarrod Overson. 2020. The State of Credential Stuffing and the Future of Account Takeovers. (Feb. 2020). Retrieved July 7, 2022 from https://youtu.be/XgPtLZQKQzo.

[44] B. Pal, T. Daniel, R. Chatterjee, and T. Ristenpart. 2019. Beyond credential stuffing: Password similarity models using neural networks. In *2019 IEEE Symposium on Security and Privacy (Los Alamitos, CA, USA) (SP'19)*. IEEE, 814–831. https://doi.org/10.1109/SP.2019.00056

[45] Tommy Pauly and Delziel Fernandes. 2021. Get ready for iCloud Private Relay. (June 2021). Retrieved July 7, 2022 from https://developer.apple.com/videos/play/wwdc2021/10096/.

[46] Thanasis Petsas, Giorgos Tsirantonakis, Elias Athanasopoulos, and Sotiris Ioannidis. 2015. Two-factor authentication: Is the world ready?: Quantifying 2FA adoption. In *Eighth European Workshop on System Security (Bordeaux, France) (EuroSec'15)*. ACM. https://doi.org/10.1145/2751323.2751327

[47] Nils Quermann, Marian Harbach, and Markus Dürmuth. 2018. The state of user authentication in the wild. In *Who are you?! Adventures in Authentication Workshop 2018 (Baltimore, MD, USA) (WAY'18)*. https://wayworkshop.org/2018/papers/way2018-quermann.pdf.

[48] Elissa M. Redmiles, Everest Liu, and Michelle L. Mazurek. 2017. You want me to do what? A design study of two-factor authentication messages. In *Who Are You?! Adventures in Authentication Workshop 2017 (Santa Clara, CA, USA) (WAY'17)*. USENIX Association. Retrieved July 7, 2022 from https://www.usenix.org/conference/soups2017/workshop-program/way2017/redmiles.

[49] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. 2019. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security (Santa Clara, CA, USA) (SOUPS'19)*. USENIX Association, 357–370. https://www.usenix.org/conference/soups2019/presentation/reese.

[50] Gerardo Reynaga and Sonia Chiasson. 2013. The usability of CAPTCHAs on smartphones. In *10th International Conference on Security and Cryptography (Reykjavik, Iceland) (SECRYPT'13)*. SciTePress, 427–434. https://doi.org/10.5220/0004533904270434

[51] Esteban Rivera, Lizzy Tengana, Jesús Solano, Alejandra Castelblanco, Christian López, and Martín Ochoa. 2020. Risk-based authentication based on network latency profiling. In *13th ACM Workshop on Artificial Intelligence and Security (AISec'20)*. ACM, 105–115. https://doi.org/10.1145/3411508.3421377

[52] Ahmad Sabouri. 2016. On the user acceptance of privacy-preserving attribute-based credentials – a qualitative study. In *11th International Workshop on Data Privacy Management and Security Assurance*. Springer, 130–145. https://doi.org/10.1007/978-3-319-47072-6_9

[53] scikit learn. 2020. OneHotEncoder. (2020). Retrieved July 7, 2022 from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

[54] scikit learn. 2021. Novelty and Outlier Detection. (2021). Retrieved July 7, 2022 from https://web.archive.org/web/20210812084024/https://scikit-learn.org/stable/modules/outlier_detection.html.

[55] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer, 145–158. https://doi.org/10.1007/978-3-642-23808-6_10

[56] Theresa Stadler and Bristena Oprisanu. 2021. Privacy evaluation framework for synthetic data publishing. (July 2021). https://github.com/spring-epfl/synthetic_data_release.

[57] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic data – anonymisation groundhog day. In *31st USENIX Security Symposium (Boston, MA, USA)*. USENIX Association. https://www.usenix.org/conference/usenixsecurity22/presentation/stadler.

[58] Wiktor Stadnik and Ziemowit Nowak. 2018. The impact of web pages' load time on the conversion rate of an E-commerce platform. In *38th International Conference on Information Systems Architecture and Technology*. Springer, 336–345. https://doi.org/10.1007/978-3-319-67220-5_31

[59] statcounter. 2021. Browser Market Share Norway. (July 2021). Retrieved July 7, 2022 from https://gs.statcounter.com/browser-market-share/all/norway/.

[60] State of California. 2018. California Consumer Privacy Act. (June 2018). Assembly Bill No. 375. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375.

[61] Roland H. Steinegger, Daniel Deckers, Pascal Giessler, and Sebastian Abeck. 2016. Risk-based authenticator for web applications. In *21st European Conference on Pattern Languages of Programs (Kaufbeuren, Germany) (EuroPlop'16)*. ACM. https://doi.org/10.1145/3011784.3011800

[62] Krish Sukhani, Sahil Sawant, Sarthak Maniar, and Renuka Pawar. 2021. Automating the bypass of image-based CAPTCHA and assessing security. In *2021 12th International Conference on Computing Communication and Networking Technologies (Kharagpur, India) (ICCCNT'21)*. https://doi.org/10.1109/ICCCNT51525.2021.9580020

[63] Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. 2017. Data breaches, phishing, or malware?: Understanding the Risks of Stolen Credentials. In *2017 ACM SIGSAC Conference on Computer and Communications Security (Dallas, TX, USA) (CCS'17)*. ACM, 1421–1434. https://doi.org/10.1145/3133956.3134067

[64] Y. Tian, C. Herley, and S. Schechter. 2020. StopGuessing: Using guessed passwords to thwart online password guessing. *IEEE Security & Privacy* 18, 03 (May 2020), 38–47. https://doi.org/10.1109/MSEC.2020.2970639

[65] Twitter. 2022. Account Security - Twitter Transparency Center. (Jan. 2022). Retrieved July 7, 2022 from https://web.archive.org/web/20220211182429/https://transparency.twitter.com/en/reports/account-security.html#2021-jan-jun.

[66] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. 2003. CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques (Warsaw, Poland) (EUROCRYPT'03)*. Springer, 294–311. https://doi.org/10.1007/3-540-39200-9_18

[67] W3Counter. 2021. Web Browser Usage Trends. (July 2021). https://www.w3counter.com/trends.

[68] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. 2016. Targeted online password guessing: An underestimated threat. In *2016 ACM SIGSAC Conference on Computer and Communications Security (Vienna, Austria) (CCS'16)*. ACM, 1242–1254. https://doi.org/10.1145/2976749.2978339

[69] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. 2021. What's in score for website users: A data-driven long-term study on risk-based authentication characteristics. In *25th International Conference on Financial Cryptography and Data Security (Grenada) (FC'21)*. Springer, 361–381. https://doi.org/10.1007/978-3-662-64331-0_19

[70] Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. 2020. More than just good passwords? A study on usability and security perceptions of risk-based authentication. In *36th Annual Computer Security Applications Conference (Austin, TX, USA) (ACSAC'20)*. ACM, 203–218. https://doi.org/10.1145/3427228.3427243

[71] Stephan Wiefling, Luigi Lo Iacono, and Markus Dürmuth. 2019. 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 (Lisbon, Portugal) (IFIP SEC'19)*. Springer, 134–148. https://doi.org/10.1007/978-3-030-22312-0_10

[72] Stephan Wiefling, Tanvi Patil, Markus Dürmuth, and Luigi Lo Iacono. 2020. Evaluation of risk-based re-authentication methods. In *35th IFIP TC-11 International Conference on Information Security and Privacy Protection (Maribor, Slovenia) (IFIP SEC'20)*. Springer, 280–294. https://doi.org/10.1007/978-3-030-58201-2_19

[73] Stephan Wiefling, Jan Tolsdorf, and Luigi Lo Iacono. 2021. Privacy considerations for risk-based authentication systems. In *2021 International Workshop on Privacy Engineering (Vienna, Austria) (IWPE'21)*. IEEE, 315–322. https://doi.org/10.1109/EuroSPW54576.2021.00040