

What's in Score for Website Users: A Data-Driven Long-Term Study on Risk-Based Authentication Characteristics

Stephan Wiefling^{1,2} , Markus Dürmuth², and Luigi Lo Iacono¹ 

¹ H-BRS University of Applied Sciences, Sankt Augustin Germany

{stephan.wiefling,luigi.lo_iacono}@h-brs.de

² Ruhr University Bochum, Bochum, Germany

{stephan.wiefling,markus.duermuth}@rub.de

Abstract. Risk-based authentication (RBA) aims to strengthen password-based authentication rather than replacing it. RBA does this by monitoring and recording additional features during the login process. If feature values at login time differ significantly from those observed before, RBA requests an additional proof of identification. Although RBA is recommended in the NIST digital identity guidelines, it has so far been used almost exclusively by major online services. This is partly due to a lack of open knowledge and implementations that would allow any service provider to roll out RBA protection to its users.

To close this gap, we provide a first in-depth analysis of RBA characteristics in a practical deployment. We observed $N=780$ users with 247 unique features on a real-world online service for over 1.8 years. Based on our collected data set, we provide (i) a behavior analysis of two RBA implementations that were apparently used by major online services in the wild, (ii) a benchmark of the features to extract a subset that is most suitable for RBA use, (iii) a new feature that has not been used in RBA before, and (iv) factors which have a significant effect on RBA performance. Our results show that RBA needs to be carefully tailored to each online service, as even small configuration adjustments can greatly impact RBA's security and usability properties. We provide insights on the selection of features, their weightings, and the risk classification in order to benefit from RBA after a minimum number of login attempts.

Keywords: Risk-based Authentication (RBA) · Authentication features · Big Data Analysis · Usable Security.

1 Introduction

Despite their long known weaknesses [29,5,48,12,19,15], passwords are still used for authentication on most online services [35]. However, threats to password-based authentication continue to evolve to attacks involving targeted guessing [32] or stolen credentials sourced from data breaches [43].

Thus, online services need to implement alternative or additional measures to protect their user base. Two-factor authentication (2FA) is such a measure, but tends to be only accepted in online banking use cases [36,17,45]. Also, universal

second factor (U2F) or biometric authentication require additional hardware and active user enrollment, which makes them impractical for online services [13,21].

For these reasons, several major online services deployed risk-based authentication (RBA) to protect their users [46]. RBA is an adaptive authentication mechanism which increases password security with minimal impact on the user. It achieves better usability than comparable 2FA methods [45] and is recommended by NIST [22] to mitigate credential stuffing.

During the password entry, RBA monitors and records features that are available in this context. These feature range from network information, or device information, to behavioral information. Based on these features, RBA calculates a risk score related to the login attempt. The score is typically classified by an access threshold into low, medium, and high risk [20,28,24]. Based on the estimated risk, the RBA system can invoke multiple actions. If the score is under the threshold, i.e., a low risk, access is granted. If the score is above this threshold, i.e., medium or high risk, the online service asks for additional information (e.g., confirming an email address) or even blocks access.

RBA schemes, their configuration, and features have not been researched thus far. These are, however, of crucial importance, since they can highly impact security and usability for website users. A feature might reduce the number of re-authentication requests but could also weaken the attack protection. To further investigate this topic, we formulated the following research questions.

Research Questions. With these research questions, we aim to provide answers on how RBA performs in a practical deployment and how RBA can be configured to provide the best balance between security and usability.

- RQ1:** a) How often does RBA request for re-authentication in a practical deployment?
 b) How many user sessions need to be captured and stored in the login history to achieve a stable and reliable RBA setup?
- RQ2:** a) Which RBA features have to be chosen to achieve good security?
 b) How do RBA features need to be combined to achieve good security?
 c) How often will different RBA feature combinations request legitimate users for re-authentication?
- RQ3:** a) How practical are different RBA configurations regarding performance?
 b) How scalable and cost-efficient are different RBA configurations?

Contributions. We provide the first long-term data-driven analysis of RBA characteristics. (i) We monitored and recorded the login behavior and features of 780 users on a real-world online service for over 1.8 years. (ii) We derived two RBA models based on the majority of deployments used in current practice. (iii) We evaluated the two models on our data set and identified features that, in combination, provide good security and usability. (iv) We proposed and tested a new feature that had not yet been seen in the RBA and browser fingerprinting context before. (v) We derived how specific factors influence RBA’s performance.

The results show that even small changes to RBA settings, e.g., the feature set or access threshold, can strongly affect the usability and security properties of RBA. Our work supports service owners regarding RBA design decisions on their website. It helps administrators select suitable RBA properties—including the RBA scheme, feature set, and weightings—for their website’s characteristics and needs. Finally, researchers obtain insights on RBA’s inner workings in practice. Understanding these factors can provide a comprehensive understanding of RBA and foster a widespread adoption that goes beyond the current use by only major online services.

2 RBA Models

We derived and evaluated two RBA models based on observations on the RBA behavior of major online services [46] and algorithm descriptions in literature.

The **simple model** (SIMPLE) extends the single-feature model used in the open source single sign-on solution OpenAM [31] and is assumed to be used at GOG.com [46]. It also partly reflects models given in literature [42,24,16]. We based our implementation on OpenAM, since it is freely available and probably widely used. The SIMPLE algorithm checks a number of features for an exact match in the user’s login history. The risk score is the number of inspected features with at least one match in the login history divided by the total number of considered features. Thus, the risk score granularity increases with the number of observed features. We tested this model in two variations to observe the potential of OpenAM’s original implementation. For a fair comparison with an influential RBA algorithm in literature [20], the first variation used the features *IP address* with *IP-based geolocation*, and *user agent string* (SIMPLE-IPUA). In the second variation, we enabled the maximum number of features in the OpenAM solution to test its maximum potential (SIMPLE-ALL). Besides the three features, there were *registered client* (HTML5 canvas and WebGL fingerprint), and *last login* (i.e., logged in within the last 31 days).

The **extended model** (EXTEND) is comparable to the multi-features model that Google, Amazon, and LinkedIn used [46] and presumably still use in some form. We based this model on Freeman et al. [20], since it was the only comparable algorithm described in the literature. The model calculates the risk score S for a user u and a given feature set (x^1, \dots, x^d) with d features as [20]:

$$S_u(x) = \left(\prod_{k=1}^d \frac{p(x^k)}{p(x^k|u, \textit{legitimate})} \right) \frac{p(u|\textit{attack})}{p(u|\textit{legitimate})} \quad (1)$$

$p(x^k)$ is the probability of a feature value in the global login history and $p(x^k|u, \textit{legitimate})$ is the probability that a legitimate user has this feature value in its own login history. Since we did not collect attack data, we assumed that all users are equally likely to be attacked. Thus, we set $p(u|\textit{attack}) = \frac{1}{|U|}$, where U is the set of users with $u \in U$. The probability of legitimate logins for the user is based on the proportion of logins, i.e., $p(u|\textit{legitimate}) = \frac{\textit{Number of user logins}}{\textit{Number of all logins}}$.

Since the risk score depends on the global login history size, the risk score granularity increases with the number of entries in the global login history.

We smoothed the features with linear interpolation to add probabilities for previously unseen but plausible values [20]. We also subdivided some features into subfeatures with individual weightings (IP address \rightarrow autonomous system number (ASN) and country; user agent string \rightarrow browser/OS name and version, and device type, i.e., mobile or desktop). Freeman et al. evaluated these features and subfeatures with the help of LinkedIn [20]. Thus, these potentially represent a practical RBA feature set, which is why we chose and tested them as a baseline.

3 Data Set

We evaluated the RBA models with a data set containing real-world user behavior to identify the model characteristics in a practical deployment.

Data Collection. We recorded user data from August 2018 to June 2020 on an e-learning website for medical students. During course enrollment, they were registered at the website by the faculty staff. The students used this online service to exercise for their study courses and exams. After each successful login, we collected 247 different features of the user’s online browser, network, and device (see the pre-proceedings paper version [44] for the full list of features). The features were relevant in the field of device fingerprinting [34,3] and could help to identify users in RBA as well.

The data set is very challenging for RBA since the users are mostly located in the same city. Thus, they could get similar feature values, e.g., IP addresses, with higher probability. Testing this data will answer whether practical RBA deployments can protect users in such a challenging scenario.

Survey. The e-learning website collected usernames, hashed passwords, and features only. After the collection phase, we surveyed users between July and August 2020 to improve data quality (see Appendix A for the questionnaire).

We recruited via a mailing list of the University of Cologne, addressing students who potentially used the e-learning website between August 2018 and June 2020. We introduced the study as a survey on the overall website perception. We drew 12 Amazon vouchers worth €10 among all participants after the study.

After verifying their account, the users were redirected to the survey. Besides demographics, we included some questions about the website experience to distract from our actual study purpose. To improve data quality, we asked whether the users knew about someone illegitimately logging into their website account. We based this question on Shay et al. [39].

Demographics. In total, 182 website users (26.6% of login sessions) answered the survey. 168 users passed the attention check. The users were 61.3% female and 38.1% male (0.6% did not state the gender). The majority of users (79.7%)

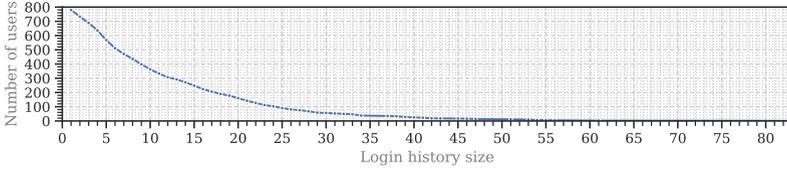


Fig. 1. Login history sizes and number of users in our data set

were between 18 and 24 years old. The remaining users were 25-34 years (18.5%), and 35-54 years old (1.8%). The age and gender distribution corresponds to the expected demographics for such a study course.

Login Sessions. The data set consisted of 780 users and 9555 logins. The users mostly logged in daily (44.3%) or several times a week (39.2%). They logged in between one and 83 times (mean: 12.25, median: 9, SD: 11.18; see Figure 1). They used desktop (81.1%) and mobile devices (18.9%). The desktop devices were Windows (62.5%), macOS (37.2%), and Linux (0.3%) based. Mobile devices were iOS (75.2%) and Android (24.8%) based. The browsers were mainly Safari (40.4%), Chrome (29.0%), Firefox (26.1%), and Edge (3.3%). To improve the quality and validity of our results, we removed users who stated an illegitimate login attempt in the survey. However, there were no such users (93.5% did not notice, 6.5% did not know).

Feature Optimization. To improve the expected performance of some of the features, we optimized them based on procedures found in literature [24,20,42,3] and as described in the following.

We extracted additional subfeatures from the IP address, user agent string, and timestamp features. Besides only extracting the hour [24], we also extracted combinations of weekday and hour to gain more information.

Administrators aiming to deploy the EXTEND model need to adjust the feature weightings to appropriate values. Freeman et al. [20] did not provide subfeature weightings for IP address and user agent string. Thus, we calculated weightings for our data set following the method described in their paper. As a result, we set the 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). We chose the weightings based on the value of information when present. They only relate to our specific data set, but can give an impression of their distribution in practice.

New Feature: Round-Trip Time. We propose a new feature that has not been seen in RBA and browser fingerprinting literature at the time of study. In concurrent and independent work, Rivera et al. [37] proposed a similar idea based on the work-in-progress resource timing API. Apart from it being a different

approach, their feature is also client originated and thus less trustworthy than our solution.

The web sockets technology [27], which is present in most online browsers today [9], allows measuring the round-trip-time (RTT). The server requests a data packet from the client and measures the time until the response. Popular online browsers Chrome and Firefox did not display this process (ping and pong frames) inside their developer tools at the time of study. RTTs can give information on whether the user’s device is really located in the indicated region, or whether the location was potentially spoofed, e.g., by VPNs or proxies [1,8]. This is also true in the presence of Content Delivery Networks (CDNs), where the CDN edge node can be linked to the RTT. This results in an even better measurement, since the edge nodes close to the user’s device are also considered.

When users entered the login credentials, we measured the RTT five times. Then, we stored the smallest RTT value to get the best possible value and to mitigate larger RTT variations, e.g., due to mobile connectivity. Besides the RTT in microseconds (RTT-RAW), we stored RTTs in milliseconds (RTT-MS), and rounded to the nearest five (RTT-5MS) and ten milliseconds (RTT-10MS).

Legal and Ethical Considerations. The participants were part of a model medical education program. During enrollment, they signed a consent form agreeing to the data collection for study purposes. They were always able to view their data on request. The collected data was stored on encrypted hard drives. Only the study researchers had access to them. The passwords on the website were hashed with `bcrypt` [33]. All participants gave informed consent on these procedures. All survey questions included a “don’t know” option.

We do not have a formal IRB process at our university. But besides our ethical considerations above, 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 EU General Data Protection Regulation.

4 Attacker Models

We evaluated the RBA systems using three attacker models based on known ones in the RBA context [20,47]. All attackers possess the victim’s login credentials.



Fig. 2. Overview of the attacker models tested in the study

The **naive attacker** tries to log in via an IP address of a random ISP located somewhere in the world and uses popular user agent strings. We simulate these attackers using a random subset of IP addresses sourced from real-world online

attacks [18]. Other feature values not related to the IP address are sourced from our data set. The **VPN attacker** knows the same as the naive attacker plus the correct country of the victim. The attacker spoofs the IP geolocation with VPN services and uses popular user agent strings. We simulate these attackers with known attacker IP addresses [18] located in the victim’s country. Feature values not derived from the IP address are sourced from our data set. We also included IP addresses not directly related to VPN services to consider services that tunnel traffic through client devices. The **targeted attacker** extends the knowledge of the VPN attacker by including locations and user agents of the victim. The attacker accesses IP addresses of ISPs in that location, likely including the victim’s ones. This attacker is identical to Freeman et al.’s *phishing attacker* [20]. We used a different term, however, as phishing is just one of the ways to obtain this level of knowledge. We simulate this attacker with our data set. The feature values are taken from all users except the victim. Since location dependent feature values in our data set were in close proximity to each other, our simulated attacker is aware of these circumstances and chooses feature values in a similar way.

5 Evaluating RBA Practice (RQ1)

Below, we analyze the RBA behavior in a practical deployment. We describe our methodology to reproduce the RBA behavior and present the results.

Step 1: Calibrating Risk Scores. The risk scores of the RBA models have different granularity (see Section 2). For a fair comparison, we calibrated the risk score access thresholds of both RBA models. We adjusted regarding the percentage of blocked attacks in each attacker model, which we call the true positive rate (TPR), as in related work [20]. We approximated the TPRs as close as possible. However, due to their granularity properties, SIMPLE TPRs were more coarse-grained than those of EXTEND.

Step 2: Determine Re-Authentication Count. By replaying user sessions, we determined how often the data set’s legitimate users were asked for re-authentication based on the number of logins. For each login attempt, we (i) restored the state at the time of the login attempt, (ii) calculated the risk score with the RBA model, and (iii) finally applied the calibrated RBA access threshold to the risk score and stored the access decision.

To provide an average estimation of the RBA behavior, we calculated the median re-authentication counts and rates for each login history size.

5.1 Results

Figure 3 shows the results for the targeted attacker case. We answer our research questions regarding practical RBA deployments in the following.

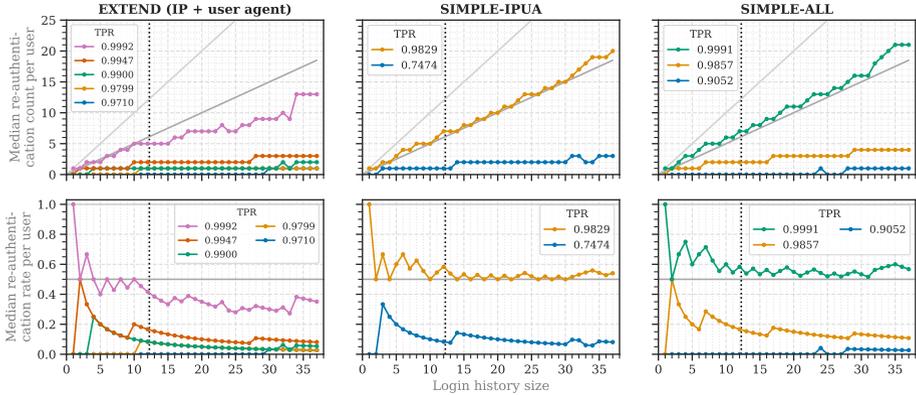


Fig. 3. Median re-authentication counts (top) and rates (bottom) per user based on the login history size. The TPR (percentage of blocked attacks) relates to targeted attackers. We added the baseline for 2FA (light grey line), the stable setup threshold (dark grey line), and the mean login count (dotted black line) for orientation. Below the stable setup threshold, users had to re-authenticate less than every 2nd login attempt.

Table 1. Median login count until re-authentication when blocking targeted attackers

| Model | TPR | Median logins until re-authentication | Model | TPR | Median logins until re-authentication | Model | TPR | Median logins until re-authentication |
|--------|---------|---------------------------------------|------------|---------|---------------------------------------|-------------|---------|---------------------------------------|
| EXTEND | 0.9992 | 2.4 | SIMPLE-ALL | 0.9991 | 1.71 | SIMPLE-IPUA | 0.9829 | 1.71 |
| | 0.9947 | 6 | | 0.9857 | 6 | | 0.7474 | 12 |
| | 0.9900 | 12 | | <0.9857 | ∞ | | <0.7474 | ∞ |
| | 0.9799 | 12 | | | | | | |
| | <0.9799 | ∞ | | | | | | |

Login history size: 12

Number of re-authentication requests in practice (RQ1a). The users logged into the website 12.25 times on mean. Thus, we considered a login history size of 12 to determine the re-authentication count for the average user in our data set. We define the median login count until re-authentication as the login history size divided by the median re-authentication count. In the following, we show the results with TPRs adjusted for each attacker model. Note that due to the risk score characteristics, attackers of lower hierarchy were always blocked as well (e.g., all naive attackers were blocked when blocking all VPN attackers).

Even when blocking all **naive attackers** with the highest possible TPR, legitimate users were never asked for re-authentication at all, except for SIMPLE-IP with TPR 0.999 (every 12th time). When **VPN attackers** were blocked, legitimate users were mostly not asked for re-authentication at all. In the other cases, they were prompted every 2.4th time (TPR 0.9995) and every 12th time (TPR 0.9946, 0.9903) with EXTEND, every 12th time with SIMPLE-IP (TPR 0.9933), and every 6th time with SIMPLE-ALL (TPR 0.9999). When blocking **targeted attackers**, our legitimate users were never asked for re-authentication with TPRs lower than 0.98 in most cases (see Table 1 and Figure 3).

Overall, the median re-authentication rate became lower with an increase in the login history size. For very high TPRs, however, the numbers did not decrease to a high degree, especially with the SIMPLE model.

Concluding the results, RBA rarely requests re-authentication for most cases in our real-world data set, even when blocking targeted attackers up to a TPR of over 0.9945 with EXTEND. However, the re-authentication rate strongly depends on the RBA model and the assumed attacker model. The influence of the feature set and the feature weightings will be analyzed in Section 6.

Required login history size (RQ1b). Since RBA is designed to request less re-authentication than 2FA for legitimate users, this difference needs to be noticeable in sensible RBA deployments. As a baseline to request less than every second login attempt, we defined the required login history size as the size above which the median re-authentication rate remains below 0.5. For statistical validity, we considered login history sizes lower than 38 since these had at least 30 users (see Section 3).

In our data set, most TPRs required one or even no history entry for blocking targeted attackers in both models (see Figure 3). However, EXTEND required ten entries for TPR 0.9992. The SIMPLE models partly did not fulfill the requirement (TPRs: 0.9829 SIMPLE-IPUA, 0.9991 SIMPLE-ALL). Based on our results, we conclude that storing one entry is already sufficient for a stable setup that blocks more than 99.45% of targeted attackers with the EXTEND model. To block 99.92% of attack attempts, ten entries are needed in our use case.

5.2 Discussion

Small variations of the access thresholds (see Section 1) can greatly affect the TPR. For instance, changing a tiny fraction of the threshold lowered the TPR from a very good 0.9829 to 0.7474 in SIMPLE-IPUA. We assume that this can make it difficult for administrators to adjust the access thresholds correctly. To foster a widespread RBA adoption in the wild, we suggest that RBA properties must be easy for administrators to estimate, apply, and control. A possible solution could be a dashboard showing the aggregated re-authentication rates and risk scores per user. These metrics can help to control and adjust the thresholds continuously and whenever necessary.

Even in settings involving a high TPR, the RBA models hardly ask for re-authentication at all. While this is a very good sign for the security properties of RBA, this influences users. Users will only feel protected by RBA if they get prompted for re-authentication at least once [45]. To support users in feeling protected, we suggest to inform about RBA being active.

6 Analyzing RBA Features (RQ2)

Based on our 247 collected features, we determined a subset that is suitable for RBA use. To be qualified for RBA use, we defined necessary criteria. The features

need to: (A) **Have both a good level of stability and at least minimum entropy**: In contrast to fingerprinting properties for tracking purposes [34], we require a certain level of entropy to make it harder for attackers to reproduce the feature values by simply brute forcing them. This might cause RBA to ask for re-authentication at a higher frequency. However, showing RBA presence by very few re-authentication requests can lead to increased (perceived) security [45]. (B) **Be spoofable only with a high amount of effort**: Easy-to-guess features will not bring any attack detection advantage to the RBA feature set baseline. (C) **Increase differentiation between legitimate users and attackers**: When added to the baseline feature set, risk scores differences between legitimate users and attackers should increase.

6.1 Study Setup

Based on the defined criteria, we developed and conducted several big data computing jobs to analyze the performance of all features in our data set.

Test A: Entropy. To identify easy-to-spoof features, we calculated the Shannon entropy of the feature values x_{i_j} of each feature $x_i \in X$ in the login history with $n = |x_i|$:

$$H_{x_i} = - \sum_{j=0}^n x_{i_j} \cdot \log_2(x_{i_j}) \quad (2)$$

We calculated two variants of entropy. To observe overall differences, we calculated the entropy $H_{global_{x_i}}$ for the global login history. To observe the feature stability inside the login history of each user, we calculated the mean Shannon entropy $\overline{H_{user_{x_i}}}$ of each feature in the user’s login history. As a result, features with $H_{global_{x_i}} = 0$ did not contain any information to distinguish between users. Similarly, features with $\overline{H_{user_{x_i}}} = 0$ did not change inside the users’ login histories.

Test B: Number of Feature Values. Some of the collected features can be spoofed by attackers with low effort. This is especially true for client submitted features, e.g., output of a JavaScript function executed in the user’s browser.

To make features harder to guess for attackers, they need to have a large range of values with equal distribution. Assuming that accounts will be locked after RBA detected an illegitimate login, it will be difficult for attackers to guess correct feature values with increasing numbers of unique feature values.

Test C: Risk Score Changes. We studied the risk score behavior of the features to evaluate their potential to improve the detection of attackers and legitimate users. We tested the features with the EXTEND model since it provides fine grained risk scores.

For each feature, we calculated the risk scores of all illegitimate login attempts by targeted attackers per user (attacker risk scores). We then calculated the risk scores of all legitimate login attempts (legitimate risk scores). After that, we determined the risk score relation (RSR) as the relation between the mean attacker and mean legitimate risk scores:

$$RSR_{basic} = \frac{\text{mean attacker risk score}}{\text{mean legitimate risk score}} \quad (3)$$

To ease comparison, we normalized the RSRs for each feature $x_i \in X$ to the baseline:

$$RSR_{x_i} = RSR_{basic_{x_i}} - RSR_{basic_{baseline}} \quad (4)$$

The feature baseline varied depending on the feature being compared to, e.g., the IP address when all compared features were added to the IP address. When testing only one feature, the baseline was a feature without any entropy, to observe risk score differences when entropy was added. If the RSR of a feature $x_i \in X$ is greater than the baseline RSR, i.e., $RSR_{x_i} > 0.0$, this feature increased the differentiation between legitimate users and attackers compared to the baseline.

Subset Extraction. For each test, we defined the following thresholds to extract a subset of suitable features for RBA use: (Test A) To extract features having at least minimum entropy, we only considered features with $H_{global_{x_i}} > 0.1$ and $\overline{H_{user_{x_i}}} > 0.1$. Based on the third quantile and the specific characteristics of the data set, we chose this threshold as a minimum baseline. (Test B) To focus on harder-to-guess features for RBA, we considered those with more than ten unique feature values. More features were considered for both desktop and mobile users in the global login history to adequately address security. We made sure to check both mobile and desktop devices since mobile devices tend to have less unique RBA feature values than desktop devices [40]. (Test C) To ignore features causing only small RSR improvements, we considered features with $RSR_{x_i} > 0.1$.

Feature Reliability. The extracted features were present on all user sessions but were very diverse, ranging from client originated to server side recorded. Thus, we labeled them by the following properties: (i) **Server side:** These features are measured on the server side. Since they do not depend on client originated input, they add a high level of trust. (ii) **Client side JavaScript not required:** There might be users that deactivated JavaScript, e.g., for privacy reasons. To ensure compatibility, we labeled features that can be measured without JavaScript.

Based on the properties, we distinguished three categories of RBA features: **Single features** add a high level of reliability and provide good RBA performance on their own. **Major add-on features** are similar, but they only achieve good RBA performance when added to a single feature. Both feature types can

Table 2. Single and major add-on features that qualified for RBA use. The only single feature is the IP address (bold). The other ones are major features that can be used in addition to a single feature. All features are server originated and hence hard to spoof.

| Feature | JavaScript not required | RSR | H_{global} | \overline{H}_{user} | Unique values | Median logins until re-authentication |
|-------------------|----------------------------|-------|--------------|-----------------------|---------------|--|
| IP address | ● | 1.20 | 10.51 | 1.96 | ●●●●● | **2.00 |
| RTT-10MS | ○ | 1.75 | 2.45 | 1.04 | ●●○○○ | 1.50 |
| RTT-5MS | ○ | 1.37 | 3.27 | 1.33 | ●●○○○ | 1.71 |
| ASN (IP) | ● | 0.91 | 3.17 | 0.76 | ●●○○○ | 3.00 |
| RTT-MS | ○ | 0.56 | 5.43 | 2.00 | ●●●●○ | 2.00 |
| Hour | ● | 0.23 | 4.06 | 2.31 | ●○○○○ | 4.00 |
| Region (IP) | ● | 0.15 | 1.20 | 0.31 | ●○○○○ | 1.71 |
| Weekday and hour | ● | 0.15 | 6.72 | 2.78 | ●●●○○ | 4.00 |

Significantly higher than the baseline: ** p < 0.01

Baselines: Zero entropy feature (single feature), IP address (major add-on feature)

Unique values: Five dot scale (very low, low, medium, high, very high) mapped to the values (10-24, 25-74, 75-149, 150-300, >300).

be used with high weighting and are measured on the server side. **Add-on features** are not as reliable as the features above but can be used in addition to single features. They are client originated. Therefore, it is possible that some of them could be blocked or modified, e.g., by anti-tracking measures [7].

Re-authentication Count Changes. We assume that less requests for re-authentication can increase RBA usability and user acceptance [45]. Thus, we measured whether certain features have the potential to decrease the requests for legitimate users. We calculated the median login count until re-authentication for average legitimate users (i.e., 12 logins) and a TPR of 0.8 (targeted attackers) for each feature. We selected the TPR to allow all features to get a TPR close to the desired TPR for fair comparison. Also, selecting targeted attackers allowed us to test the features against the best possible attacker.

High re-authentication counts can signal administrators to weigh this feature lower, in combination with other features having lower counts, to balance usability.

6.2 Results

In the following, we present our results ordered by the three RBA feature categories. For statistical testing, we used Kruskal-Wallis tests for the omnibus cases and Dunn’s multiple comparison test with Bonferroni correction for post-hoc analysis. We considered p-values lower than 0.05 as significant.

We calculated the risk scores on a high-performance computing (HPC) cluster with more than 2400 CPU cores. This was necessary since such calculations were computationally intensive. Using the HPC cluster reduced the calculation time to approximately two days for all features (instead of 123.5 days using 32 cores).

After combining the features that passed all three tests, only the IP address qualified as a **single feature** for RBA use. When being used in addition to the IP address, seven features qualified as **major add-on features**, all of them network or behavior based (see Table 2). Since the IP address was the only appropriate

Table 3. Add-on features that qualified for RBA use in addition to single features. In comparison to major add-on features, they are client originated and thus spoofable.

| Feature | JavaScript not required | RSR | H_{global} | $\overline{H_{user}}$ | Unique values | Median logins until re-authentication |
|------------------------------------|----------------------------|-------|--------------|-----------------------|---------------|--|
| Session Cookie | ● | 22.39 | 9.51 | 0.51 | ●●●●● | **12.00 |
| User agent string (w/ subfeatures) | ● | 10.33 | 7.43 | 1.21 | ●●●●● | **12.00 |
| Screen width and height | ○ | 3.28 | 4.70 | 0.64 | ●●●●○ | 3.00 |
| WebGL fingerprint | ○ | 3.14 | 4.12 | 0.55 | ●●●○○ | 4.00 |
| Accept language header | ● | 3.01 | 2.57 | 0.33 | ●●●○○ | 3.00 |
| App version | ○ | 2.74 | 6.29 | 1.01 | ●●●●● | 2.40 |
| Available width and height | ○ | 2.72 | 6.36 | 0.95 | ●●●●● | 3.00 |
| OS full version | ● | 2.64 | 3.90 | 0.59 | ●●●○○ | 2.40 |
| WebGL Version | ○ | 2.59 | 2.14 | 0.36 | ●○○○○ | 2.40 |
| WebGL extensions | ○ | 2.52 | 3.62 | 0.56 | ●○○○○ | 6.00 |
| HTML5 canvas fingerprint | ○ | 2.28 | 6.45 | 0.77 | ●●●●● | 3.00 |
| OS name and version | ● | 2.27 | 3.91 | 0.59 | ●●○○○ | 2.40 |
| Browser major version | ● | 2.03 | 4.28 | 0.80 | ●●○○○ | **6.00 |
| Device pixel ratio | ○ | 1.94 | 2.66 | 0.51 | ●○○○○ | 2.40 |
| User agent string (no subfeatures) | ● | 1.74 | 7.43 | 1.21 | ●●●●● | 3.00 |
| Main language | ○ | 1.61 | 1.35 | 0.24 | ●○○○○ | 3.00 |
| Browser full version | ● | 1.27 | 5.49 | 0.96 | ●●○○○ | **12.00 |
| Browser name and version | ● | 1.14 | 5.85 | 1.02 | ●●○○○ | **6.00 |
| Local IP address | ○ | 1.13 | 3.27 | 0.49 | ●●●●● | 1.00 |
| Webkit temporary storage | ○ | 0.92 | 3.30 | 0.39 | ●●●●● | 1.00 |
| Battery discharging time | ○ | 0.75 | 1.95 | 0.48 | ●●●●● | 1.00 |

Significantly higher than the baseline: ** $p < 0.01$

Unique values: Five dot scale (very low, low, medium, high, very high) mapped to the values (10-24, 25-74, 75-149, 150-300, >300).

The session cookie was set by the server. RBA simply compared the stored value.

We omitted similar features for space reasons (see Table 4 in Appendix B for all results).

single feature for this case, we extracted the **add-on features** using this feature. 27 features qualified by passing all three tests (see Table 3).

Conclusion. In summary, a set of features has to be chosen in most cases rather than a single feature to achieve good RBA security. Using only one feature for RBA risk estimation will make it hard to reliably distinguish between attackers and legitimate users. The feature set needs to at least include features that we identified as single or major add-ons (see Table 2) for good RBA security.

6.3 Discussion

The results confirm our previous findings [46] that IP address, user agent, display resolution, language, and login time are useful RBA features and hence, find adoption in the wild. The results also show that most of the 247 analyzed features are not suitable for RBA use. Many of them had few unique values or low RSRs. This is good for privacy, as few features need to be collected. Also, many of the popular features [46] are collected on the server side anyway, e.g., in the logs [25]. Still, some of them may contain sensitive data [6] and must be protected against data breaches. But, as we considered all features as categorical data, these can be hashed, or even truncated to some degree, to produce the same results. Our results suggest a set of relevant RBA features may provide security benefits while preserving usability. This set is rather small compared to the 247 evaluated features. Thus, we discuss how to design a minimal RBA feature set to also balance privacy. We discuss a selection of relevant features and feature combinations based on our results and findings in literature below.

Features. The **IP address** proved to be the only RBA feature that can be used as a single feature. The **region** and **ASN** are also hard to fake and to obtain since they require network access from a specific ASN in a specific location.

The **RTT** turned out as a promising new RBA feature when being rounded to milliseconds at least. Attackers need access to a device physically located inside the victim’s location to forge this feature. Thus, using the **RTT** would add high costs for attackers. However, due to more re-authentication requests, the **RTT** needs to be weighted lower than other features to balance usability.

Timing features like **weekday and hour** increased security attributes while having few re-authentication requests. Successful attacks need to estimate the victim’s usual login times right to the day and hour, which can be greater effort. This is especially the case for services that are not used on a daily basis.

The **user agent string** performed very well when used in combinations with a subfeature hierarchy, confirming findings of Freeman et al. [20].

Since it can be used as a unique session identifier, the **cookie** seems to be an obvious feature choice, and our results would support this view. However, cookies should only be used very carefully or not at all as a RBA feature. They would have to be stored permanently in the login history. Since there is no revocation mechanism in the current RBA models, every cookie inside the login history would always be valid. Thus, a stolen and even outdated cookie might have a negative impact on the risk score, leading to false positives.

Feature combinations. The **IP address and user agent string** features are often named in literature [20,24,40]. According to our observations related to the data set, they increased the RSR and significantly reduced re-authentications compared to the single features.

RBA models in literature often use **user agent strings** to identify a browser [24,40,20,46,16]. However, **HTML5 canvas and WebGL fingerprints** [30,14] are newer approaches considered more difficult to fake. Both approaches received lower RSRs and significantly higher re-authentication counts compared to the user agent string in our data set. Following that, if canvas or WebGL fingerprinting should be used to strengthen security, one should consider using them with lower weightings.

7 Analyzing RBA Configurations (RQ3)

For good usability, the latency between submitting the login credentials and getting the risk decision needs to be low. An acceptable delay ranges below 300 ms when considering the page load time [41]. Thus, we analyzed which properties have an impact on the risk score calculation time. This can help to design RBA systems with both good security and a low authentication time.

We replayed all legitimate logins with both models and measured the time it took to calculate the risk score. 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 used Kruskal-Wallis tests to check for significant differences between features. For

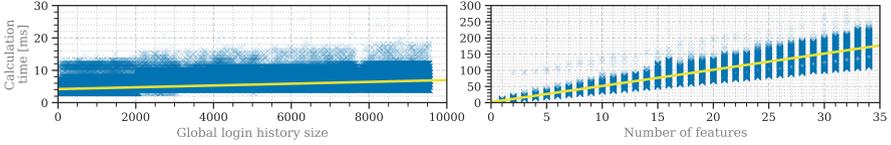


Fig. 4. Relationship between risk score calculation time and the size of the global login history (left) or number of features (right) for EXTEND. The diagonal line represents the fitted linear regression model. Left: We limited the y-axis to 30 ms for readability.

variables suggesting a relation, we calculated the linear least squares regression between them. We determined the effect sizes based on Cohen [11].

(Test 1) We first measured the calculation times for **every feature**. The median calculation times ranged 4.5-8 ms for EXTEND (median: 5.63; SD: 0.9), and 0.07-2.7 ms for SIMPLE (median: 0.08; SD: 0.17). There were no significant differences between the features. However, there was a large significant effect between the calculation time and the global login history size for EXTEND. The linear regression yielded $y = 4.1912 + 0.0003 \cdot x$, with y being the time in ms and x the global login history size ($R^2=0.42$; $f=0.85$; $p \ll 0.0001$).

(Test 2) We then measured the calculation time based on the **number of features** in the feature set. However, testing all $2^{247} - 1$ combinations was not feasible. Since there were no significant differences between all features in Test 1, we chose the feature that ranged in the middle of all median calculation times. We did this to select a feature that matches all features as well as possible. We took this feature, added it to the feature set, measured the times, and did it again until we reached the maximum number of features found in RQ2.

The results showed significant effects between the number of features and the calculation time (see Figure 4). The fitted linear regression model resulted in $y = 1.5568 + 5.0038 \cdot x$ and a large effect size for EXTEND ($R^2=0.93$; $f=3.71$; $p \ll 0.0001$), with y being the time in ms and x the number of features. Linear regression for SIMPLE resulted in $y = -0.0119 + 0.0013 \cdot x$ and a medium effect ($R^2=0.12$; $f=0.37$; $p \ll 0.0001$). However, the latter effects were hardly noticeable.

Discussion. Administrators need to keep track of the included global login history and features to ensure an acceptable authentication speed. The results show that including a high amount of features impacts the performance for EXTEND. However, our results for RQ1 already showed that capturing few features was sufficient for good security and usability.

8 Limitations

We implemented the RBA models using Python. High-level programming languages like C++ might have reduced the calculation time. Nevertheless, our results can still give estimates on factors that influence RBA performance.

The results are limited to the data set tested and the users who participated. Our results are not representative of large-scale online services, but represent

a typical use case scenario of a daily to weekly use online service in a certain country. We assume that the IP country feature would have qualified with an international user base [20]. To allow a fair comparison of all features, we weighted all features equally. We expect, however, that service owners weigh features individually, possibly improving the RBA performance. Thus, we assume that our study results represent a RBA performance baseline.

As in similar studies, we can never fully exclude that the website was targeted by intelligent attackers. However, we implemented multiple countermeasures. The website URL was only provided to students signing an informed consent. The URL was not accessible via search engines due to geoblocking and other measures to disallow crawling the site. IP scans reaching the website’s IP address only received a white page instead of the e-learning website. The TLS certificate also did not reveal the real DNS entry in this case. The fact that users did not notice illegitimate login attempts and no data breaches were known underlines that the website was likely not infiltrated.

9 Related Work

In previous work, we studied RBA’s usability characteristics [45,47]. The results helped to estimate the usability of RBA characteristics in this study. To the best of our knowledge, no studies analyzing RBA characteristics with long-term login data exist in literature. Freeman et al. [20] tested their RBA model on a LinkedIn data set using only IP address and user agent string as features. In contrast to them, we tested their model with a huge set of features.

There is also related work regarding browser fingerprinting features for user authentication purposes. Alaca and van Oorschot [3] classified 29 fingerprinting features which have the potential to be used for user authentication. They selected the features based on literature research but, in contrast to our study, did not test them on real data. Spooren et al. [40] tested OpenAM’s RBA mechanism on simulated data with six features, which were screen resolution, browser plugins, fonts, timezone, user agent, and geolocation. They found that mobile devices were less reliable in terms of being uniquely identified. We were able to confirm their findings for these six features. However, our study shows that there are other features that can reliably identify mobile device users. Campobasso and Allodi [8] studied a criminal infrastructure that tries to bypass RBA on malware infected victim devices. Since its geolocation spoofing relied on SOCKS5 proxies, our new RTT feature can detect these attacks. Andriamilanto et al. [4] tested fingerprints of website users regarding their capability to be used for authentication purposes. In contrast to our study, their data set did not relate to login attempts, contained only client originated features, and was not tested on RBA.

10 Conclusion

As long as password-based authentication predominates, constantly evolving data breaches and targeted attacks with breached passwords [2] increase the

need of RBA for online services to protect their users. NIST recommends RBA use since 2017 [22]. However, the current body of knowledge does not provide insights on RBA characteristics. Understanding these is important to ensure that practical RBA deployments protect users as much as possible while balancing usability. To close this gap, we studied RBA characteristics with long-term usage data of a real-world online service. Our results show that RBA can achieve low re-authentication rates for legitimate users when blocking more than 99.45% of targeted attacks with the EXTEND model. Moreover, our findings also show that only few of the 247 collected features can be considered useful for practical RBA deployments. The IP address is confirmed to be a must-have feature in general, but it should be enriched by add-on features. Among them, the introduced RTT showed to be a new promising feature. Cookies, however, should only be used with great care or not at all, as stolen credentials together with a stolen cookie might outweigh other features and falsely grant access.

Our contribution indicates that simply acquiring one of the commercially or freely available RBA solutions is not sufficient. They still need to be customized for the targeted online service in order to be optimized in terms of security and usability. We provided insights on how to select proper features, their weightings, and the access threshold. Based on our findings, we recommend to use RBA algorithms comparable to the introduced EXTEND model, since its security and usability properties outweighed the SIMPLE model. Overall, RBA protection should be put in place shortly after the first deployment, as the login history size did not affect it in our study.

Acknowledgments. Thanks to the anonymous reviewers, our shepherd Gunes Acar, and Florian Dehling for their detailed feedback, which greatly helped improve the paper. We would like to thank Rudolf Berrendorf and Javed Razzaq for providing us a huge amount of computational power for our big data analysis. We also thank Gaston Pugliese for providing us his fingerprinting script, Annette Rieke and Jan Herrmann for their support and cooperation, and Tanvi Patil for proofreading the paper. This research was supported by the research training group “Human Centered Systems Security” (NERD.NRW) sponsored by the state of North Rhine-Westphalia. The Platform for Scientific Computing was supported by the German Ministry for Education and Research, and the Ministry for Culture and Science of the state North Rhine-Westphalia (research grant 13FH156IN6).

A Survey

We balanced all survey questions where applicable to mitigate social desirability bias [38]. The questions were presented in random order to randomly distribute ordering effects [26]. We varied the scale direction of the questions for a random half of survey participants. For questions without an ordinal scale, we randomized the response options for each participant. We did all this to randomly distribute response order bias [10,23]. We also included an attention check similar to previous work [47].

A.1 Online Service

Question (ii) and (iii) were on a five-point Likert scale including a “don’t know” option.

- (i) Which of these online services did you use at least once in the last three years? [Multiple choice] [website] Google Facebook Twitch [made-up online service that did not exist] Other: -----
The order of the subquestions varied randomly in this question.
- (ii) How much or little did [website] support you in learning the lecture material? (5 - Did fully support, 1 - Did not support at all)
- (iii) Please rate your agreement: I think I would recommend [website] to other students. (5 - Strongly agree, 1 - Strongly disagree)
- (iv) As far as you know, has anyone ever illegitimately logged into your personal [website] account? Yes, more than once Yes, only once No I don't know

A.2 Demographics

- (i) How old are you? 18-24 25-34 35-44 45-54 55-64 65-74 75 or older
 Prefer not to say
- (ii) What is your gender? Female Male Non-Binary Prefer not to say

B Features

Table 4. List of single (bold) and (major) add-on features that qualified for RBA use. All features are present in all sessions of the data set.

| Feature | Server side required | JS not required | RSR | H_{global} | \overline{H}_{user} | Unique values | Median logins until re-auth. | p |
|------------------------------------|----------------------|-----------------|-------|--------------|-----------------------|---------------|------------------------------|---------|
| IP address | ● | ● | 1.20 | 10.51 | 1.96 | 4073 | **2.00 | <0.0001 |
| Session Cookie | ○ | ● | 22.39 | 9.51 | 0.51 | 1534 | **12.00 | <0.0001 |
| User agent string (w/ subfeatures) | ○ | ● | 10.33 | 7.43 | 1.21 | 638 | **12.00 | <0.0001 |
| Screen width and height | ○ | ○ | 3.28 | 4.70 | 0.64 | 176 | 3.00 | - |
| WebGL fingerprint | ○ | ○ | 3.14 | 4.12 | 0.55 | 90 | 4.00 | 0.8057 |
| Screen height | ○ | ○ | 3.09 | 4.34 | 0.64 | 126 | 4.00 | 0.3426 |
| Accept language header | ○ | ● | 3.01 | 2.57 | 0.33 | 91 | 3.00 | - |
| Available screen width | ○ | ○ | 2.93 | 4.38 | 0.69 | 150 | 3.00 | - |
| Screen width | ○ | ○ | 2.93 | 4.28 | 0.63 | 138 | 4.00 | 0.8883 |
| App version | ○ | ○ | 2.74 | 6.29 | 1.01 | 534 | 2.40 | - |
| Available width and height | ○ | ○ | 2.72 | 6.36 | 0.95 | 411 | 3.00 | - |
| OS full version | ○ | ● | 2.64 | 3.90 | 0.59 | 93 | 2.40 | - |
| Available screen height | ○ | ○ | 2.59 | 5.91 | 0.95 | 289 | 3.00 | - |
| WebGL Version | ○ | ○ | 2.59 | 2.14 | 0.36 | 56 | 2.40 | - |
| Supported languages | ○ | ○ | 2.53 | 2.54 | 0.36 | 87 | 3.00 | - |
| WebGL extensions | ○ | ○ | 2.52 | 3.62 | 0.56 | 69 | 6.00 | 0.1601 |
| HTML5 canvas fingerprint | ○ | ○ | 2.28 | 6.45 | 0.77 | 386 | 3.00 | - |
| OS name and version | ○ | ● | 2.27 | 3.91 | 0.59 | 95 | 2.40 | - |
| Browser major version | ○ | ● | 2.03 | 4.28 | 0.80 | 57 | **6.00 | 0.0046 |
| Device pixel ratio | ○ | ○ | 1.94 | 2.66 | 0.51 | 70 | 2.40 | - |
| RTT-10MS | ● | ○ | 1.75 | 2.45 | 1.04 | 51 | 1.50 | - |
| User agent string (no subfeatures) | ○ | ● | 1.74 | 7.43 | 1.21 | 635 | 3.00 | - |
| Main language | ○ | ○ | 1.61 | 1.35 | 0.24 | 21 | 3.00 | - |
| RTT-5MS | ● | ○ | 1.37 | 3.27 | 1.33 | 67 | 1.71 | - |
| Browser full version | ○ | ● | 1.27 | 5.49 | 0.96 | 118 | **12.00 | 0.0005 |
| Browser name and version | ○ | ● | 1.14 | 5.85 | 1.02 | 161 | **6.00 | 0.0064 |
| Local IP address | ○ | ○ | 1.13 | 3.27 | 0.49 | 716 | 1.00 | - |
| Webkit temporary storage | ○ | ○ | 0.92 | 3.30 | 0.39 | 735 | 1.00 | - |
| ASN (IP) | ● | ● | 0.91 | 3.17 | 0.76 | 43 | 3.00 | - |
| Battery discharging time | ○ | ○ | 0.75 | 1.95 | 0.48 | 1007 | 1.00 | 0.0860 |
| Battery level | ○ | ○ | 0.73 | 2.36 | 0.75 | 99 | 1.00 | 0.0935 |
| RTT-MS | ● | ○ | 0.56 | 5.43 | 2.00 | 170 | 2.00 | - |
| Hour | ● | ● | 0.23 | 4.06 | 2.31 | 24 | 4.00 | - |
| Region (IP) | ● | ● | 0.15 | 1.20 | 0.31 | 16 | 1.71 | - |
| Weekday and hour | ● | ● | 0.15 | 6.72 | 2.78 | 145 | 4.00 | 0.2117 |

Significantly higher than the baseline: * p < 0.05 ** p < 0.01
We omitted p-values of 1.0 for readability reasons.

References

1. Abdou, A., van Oorschot, P.C.: Secure Client and Server Geolocation over the Internet. *login: Spring* 2018 **43**(1) (2018)
2. Akamai: Credential Stuffing: Attacks and Economies. [state of the internet] / security **5**(Special Media Edition) (Apr 2019)
3. Alaca, F., van Oorschot, P.C.: Device Fingerprinting for Augmenting Web Authentication: Classification and Analysis of Methods. In: ACSAC '16 (Dec 2016)
4. Andriamilanto, N., Allard, T., Guelvouit, G.L.: "Guess Who?". In: IMIS '20 (2021)
5. Bonneau, J.: The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In: SP '12 (May 2012)
6. Bonneau, J., Felten, E.W., Mittal, P., Narayanan, A.: Privacy concerns of implicit secondary factors for web authentication. In: WAY '14 (2014)
7. Bujlow, T., Carela-Espanol, V., Lee, B.R., Barlet-Ros, P.: A Survey on Web Tracking: Mechanisms, Implications, and Defenses. *Proc. IEEE* **105**(8) (Aug 2017)
8. Campobasso, M., Allodi, L.: Impersonation-as-a-service: Characterizing the emerging criminal infrastructure for user impersonation at scale. In: CCS '20 (Nov 2020)
9. caniuse.com: Web sockets (Jul 2020)
10. Chan, J.C.: Response-Order Effects in Likert-Type Scales. *Educational and Psychological Measurement* **51**(3) (Sep 1991)
11. Cohen, J.: Statistical power analysis for the behavioral sciences. 2nd ed edn. (1988)
12. Das, A., Bonneau, J., Caesar, M., Borisov, N., Wang, X.: The Tangled Web of Password Reuse. In: NDSS '14 (Feb 2014)
13. Das, S., Dingman, A., Camp, L.J.: Why Johnny Doesn't Use Two Factor A Two-Phase Usability Study of the FIDO U2f Security Key. In: FC '18 (Feb 2018)
14. Daud, N.I., Haron, G.R., Othman, S.S.S.: Adaptive authentication: Implementing random canvas fingerprinting as user attributes factor. In: ISCAIE '17 (Apr 2017)
15. Dhamija, R., Tygar, J.D., Hearst, M.: Why phishing works. In: CHI '06 (Apr 2006)
16. Djopic, N., Nokovic, B., Sharieh, S.: Machine Learning in Action: Securing IAM API by Risk Authentication Decision Engine. In: CNS '20 (Jun 2020)
17. Dutson, J., Allen, D., Egeget, D., Seamons, K.: "Don't punish all of us": Measuring User Attitudes about Two-Factor Authentication. In: EuroUSEC '19 (Jun 2019)
18. FireHOL: All cybercrime ip feeds (Aug 2020), http://iplists.firehol.org/?ipset=firehol_level4
19. Florencio, D., Herley, C.: A large-scale study of web password habits. In: WWW '07 (May 2007)
20. Freeman, D., Jain, S., Dürmuth, M., Biggio, B., Giacinto, G.: Who Are You? A Statistical Approach to Measuring User Authenticity. In: NDSS '16 (Feb 2016)
21. Gaddam, A.: Usage of Behavioral Biometric Technologies to Defend Against Bots. In: Enigma 2019 (Jan 2019)
22. Grassi, P.A., et al.: Digital identity guidelines: authentication and lifecycle management. Tech. Rep. NIST SP 800-63b, NIST, Gaithersburg, MD (Jun 2017)
23. Hartley, J.: Some thoughts on Likert-type scales. *International Journal of Clinical and Health Psychology* **14**(1) (Jan 2014)
24. Hurkała, A., Hurkała, J.: Architecture of context-risk-aware authentication system for web environments. In: ICIEIS '14 (Sep 2014)
25. IBM: Log File Formats: NCSA Combined Log Format (2003)
26. Kalton, G., Schuman, H.: The Effect of the Question on Survey Responses: A Review. *Journal of the Royal Statistical Society. Series A (General)* **145**(1) (1982)

27. Melnikov, A., Fette, I.: The WebSocket Protocol. No. 6455 in Request for Comments (Dec 2011)
28. Molloy, I., Dickens, L., Morisset, C., Cheng, P.C., Lobo, J., Russo, A.: Risk-based Security Decisions Under Uncertainty. In: CODASPY '12 (Feb 2012)
29. Morris, R., Thompson, K.: Password security. *Comm. ACM* **22**(11) (Nov 1979)
30. Mowery, K., Shacham, H.: Pixel Perfect. In: W2SP '12 (May 2012)
31. Open Identity Platform: OpenAM: Adaptive Authentication Module (Aug 2016), <https://git.io/JteWg>
32. Pal, B., Daniel, T., Chatterjee, R., Ristenpart, T.: Beyond Credential Stuffing: Password Similarity Models Using Neural Networks. In: SP '19 (May 2019)
33. Percival, C., Josefsson, S.: The scrypt Password-Based Key Derivation Function. *Tech. Rep. RFC7914* (Aug 2016)
34. Pugliese, G., Riess, C., Gassmann, F., Benenson, Z.: Long-Term Observation on Browser Fingerprinting. *Proc. PETS* **2020**(2) (Apr 2020)
35. Quermann, N., Harbach, M., Dürmuth, M.: The State of User Authentication in the Wild. In: WAY '18 (Aug 2018)
36. Reynolds, J., Smith, T., Reese, K., Dickinson, L., Ruoti, S., Seamons, K.: A Tale of Two Studies: The Best and Worst of YubiKey Usability. In: SP '18 (May 2018)
37. Rivera, E., Tengana, L., Solano, J., Castelblanco, A., López, C., Ochoa, M.: Risk-based authentication based on network latency profiling. In: AISec '20 (2020)
38. Shaeffer, E.M.: Comparing the Quality of Data Obtained by Minimally Balanced and Fully Balanced Attitude Questions. *Public Opinion Quarterly* **69**(3) (Sep 2005)
39. Shay, R., Ion, I., Reeder, R.W., Consolvo, S.: "My religious aunt asked why i was trying to sell her viagra". In: CHI '14 (Apr 2014)
40. Spooren, J., Preuveeners, D., Joosen, W.: Mobile device fingerprinting considered harmful for risk-based authentication. In: EuroSec '15 (Apr 2015)
41. Stadnik, W., Nowak, Z.: The Impact of Web Pages' Load Time on the Conversion Rate of an E-Commerce Platform. In: ISAT '17 (Sep 2018)
42. Steinegger, R.H., Deckers, D., Giessler, P., Abeck, S.: Risk-based authenticator for web applications. In: EuroPlop '16 (Jun 2016)
43. Thomas, K., et al.: Protecting accounts from credential stuffing with password breach alerting. In: USENIX Security '19 (Aug 2019)
44. Wiefling, S., Dürmuth, M., Lo Iacono, L.: What's in Score for Website Users: A Data-driven Long-term Study on Risk-based Authentication Characteristics. In: FC '21 (Pre-Proceedings) (Mar 2021), <https://nbn-resolving.org/urn:nbn:de:hbz:1044-opus-53053>
45. Wiefling, S., Dürmuth, M., Lo Iacono, L.: More Than Just Good Passwords? A Study on Usability and Security Perceptions of Risk-based Authentication. In: ACSAC '20 (Dec 2020)
46. Wiefling, S., Lo Iacono, L., Dürmuth, M.: Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild. In: IFIP SEC '19 (Jun 2019)
47. Wiefling, S., Patil, T., Dürmuth, M., Lo Iacono, L.: Evaluation of Risk-based Re-Authentication Methods. In: IFIP SEC '20 (Sep 2020)
48. von Zezschwitz, E., De Luca, A., Hussmann, H.: Honey, I shrunk the keys. In: NordiCHI '14 (Oct 2014)