Is This Really You? An Empirical Study on Risk-Based Authentication Applied in the Wild

Stephan Wiefling[1], Luigi Lo Iacono[1], and Markus Dürmuth[2]

[1] TH Köln - University of Applied Sciences, Cologne, Germany
{stephan.wiefling, luigi.lo_iacono}@th-koeln.de
[2] Ruhr University Bochum, Germany markus.duermuth@rub.de

Abstract. Risk-based authentication (RBA) is an adaptive security measure to strengthen password-based authentication. RBA monitors additional implicit features during password entry such as device or geolocation information, and requests additional authentication factors if a certain risk level is detected. RBA is recommended by the NIST digital identity guidelines, is used by several large online services, and offers protection against security risks such as password database leaks, credential stuffing, insecure passwords and large-scale guessing attacks. Despite its relevance, the procedures used by RBA-instrumented online services are currently not disclosed. Consequently, there is little scientific research about RBA, slowing down progress and deeper understanding, making it harder for end users to understand the security provided by the services they use and trust, and hindering the widespread adoption of RBA.
In this paper, with a series of studies on eight popular online services, we (i) analyze which features and combinations/classifiers are used and are useful in practical instances, (ii) develop a framework and a methodology to measure RBA in the wild, and (iii) survey and discuss the differences in the user interface for RBA. Following this, our work provides a first deeper understanding of practical RBA deployments and helps fostering further research in this direction.

1 Introduction

Weaknesses in password-based authentication have been known for a long time [21]. They range from weak and easy to guess passwords [4, 29] or password re-use [9] to being susceptible to phishing attacks. Still, passwords are the predominant authentication mechanism deployed by online services today [6, 23]. To increase the users' security, service operators should implement additional measures. *Two-factor authentication (2FA)* [22] is one widely offered measure that improves account security, but is rather unpopular (e.g. in January 2018, less than 10 % of active Google accounts used 2FA [19]). *Risk-based authentication (RBA)* [11] is an approach that increases security with minimal impact on user interaction, and thus has the potential to provide secure authentication with good usability. It is among the approaches suggested by the NIST digital identity guidelines to mitigate online guessing attacks [14].

Postprint version of a paper accepted for IFIP SEC 2019. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-030-22312-0_10

Risk-based Authentication (RBA) RBA is typically used in addition to passwords or other forms of user authentication. It is designed to protect against a rather strong attacker that either knows the correct credentials (i.e., username / password pair) or can guess correct credentials with a low number of guesses. Examples include *credential stuffing attacks* [30] where an attacker tries credentials leaked from another service, *phishing attackers*, or *online guessing attacks* [29]. During password entry RBA monitors and records additional features that are contextually available. In principle, a number of various distinct features can be taken into account (see Table 1), including the *IP address* and derived features such as *geolocation* or *country*, and the *user agent* [5, 11]. Some features are better suited for risk assessment than others: The IP address, e.g., could be rated as "more important" than the user agent string since spoofing an IP address is considered as more difficult than the latter [3].

From these features a *risk score* is calculated. It is then typically classified into three buckets (low, medium and high risk) [11, 20, 16]. Depending on the risk score and its classification, a variety of actions can be performed by the service. When a risk score exceeds e.g. the low threshold and falls into the medium risk category, the service typically requests additional authentication factors from the user (e.g. verification of email address or phone number [17, 24, 11]), requires to solve a CAPTCHA [24], or informs the user about suspicious activities [13]. If the risk score is deemed high, the service can decide to block access altogether, but this event is rare, as it will not allow legitimate users mistakenly classified as a high risk to recover. The thresholds of when a user becomes suspicious have to be carefully chosen for each individual RBA use case scenario.

Contribution We investigate how RBA is used on eight high-traffic online services (Amazon, Facebook, GOG.com, Google, iCloud, LinkedIn, Steam and Twitch). We created 28 virtual identities and 224 user accounts for this purpose. During a period of 3.5 months we conducted studies to determine (an approximation to) a set of features that contributes to the risk score computation, and studied the influence of these features. We also captured and analyzed the deployed additional authentication factors. Our studies revealed serious vulnerabilities emphasizing the need for an open discussion on RBA in science.

To achieve reliable and repeatable results, we developed an automated browser testing framework and simulated human-like user behavior with individual activities on each of the online services. The framework contains enhanced technical camouflage measures to be indistinguishable from human users. The developed testing framework[3] can be used to analyze black boxed services for RBA features. Our work is intended to support both research and development. Researchers benefit from an increased transparency on the current practice of RBA deployment. Also, they obtain a test methodology and tooling for running replication or follow-up studies. Developers obtain guided insights on how to best create or improve own RBA implementations. The same is true for administrators aiming at integrating RBA as an additional line of defense in their online services. This

[3] Provided as open source software at https://github.com/DASCologne/HOSIT

all contributes to an open scientific discussion on RBA, ultimately leading to a comprehensively understood security measure, leaving no room for obscurities. We hope that public research on RBA will enable a broader adoption of RBA and thus protect a larger user base, while currently only larger online services are capable to offer RBA techniques (beyond very basic and inaccurate service).

Outline The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the developed automated testing framework, created identities and prerequisites for the studies. The study setup and obtained results are described in Section 4. We discuss findings and limitations in Section 5 and conclude with the main contributions and an outlook on future work in Section 6.

2 Related Work

The features and authentication factors deployed by RBA-instrumented online services are currently either not disclosed or just briefly mentioned by the respective companies [19, 17, 18]. This lack hinders any scientific debate and rigorous analysis to facilitate the effective and open use of RBA. These debates and analyses are even more important today since RBA is recommended by NIST [14] and therefore becoming a requirement for many IT security professionals.

Most of the RBA-related research is focused on evaluating the reliability and robustness of certain features. A RBA method based on mouse and keyboard dynamics was developed and tested by Traore et al. [27]. Judging from the observed equal error rate, they concluded that this method is not suitable for RBA inside the login process. Hurkala and Hurkala [16] published a software architecture of a RBA system. The features *IP address*, *login time*, *availability of cookie*, *device profiling* and *failed login attempts* are implemented in the RBA system. The limitations and effectiveness of these features were not estimated. Freeman et al. [11] presented the, to the best of our knowledge, first publicly known RBA algorithm using *IP address* and *user agent* as features. Steinegger et al. [26] presented another RBA implementation, with *browser fingerprint*, *failed login attempts* and *IP based geolocation* as features. Alaca and van Oorschot [3] classified and rated 29 distinct methods for device fingerprinting regarding possible *"distinguishing info"*. They rated *IP address* and *geolocation* as *"high"*. Daud et al. [10] introduced an adaptive authentication method applying HTML5 canvas fingerprinting. The effectiveness of this method is unclear due to the lack of testing with participants. Herley and Schechter [15] presented a method for authentication servers to distinguish attacks from legitimate traffic. They rated the *password used for a failed attempt* as a strong feature to identify attacks.

Petsas et al. [22] estimated the quantity of Google user accounts with enabled 2FA functionality. They used headless browser automation with enhancements for user simulation. Their methodology, using browser automation and observing reactions, is roughly similar to ours. However, due to the complex nature of RBA and novel browser automation detection methods [28], a considerably higher amount of effort was necessary in our studies.

3 Black Box Testing RBA

In this section we introduce the developed approach for black box testing RBA implementations in the wild. The basic methodology is to create accounts on the inspected online services and to observe the behavior when accessing the service using these accounts for a variety of scenarios. This seemingly simple procedure is complicated by a number of factors: (i) The account's login history may influence the risk score. Thus, testing multiple scenarios with the same account may produce unreliable results. (ii) Automated testing is likely influencing the outcome, as one of the tasks of RBA is specifically to detect bots. (iii) The list of features that potentially may be used by online services to determine the risk score is vast, and simply testing all combinations is next to impossible. (iv) Depending on the service's implementation of RBA, the feedback can be coarse-grained, i.e., giving mostly binary information (RBA triggered/not triggered), while other online services provide more fine-grained information.

Our approach considers these issues and mitigates their effects on the results. We created a larger number of virtual identities and spent several weeks to train them on legitimate behavior. The data collection uses an extensively patched version of Chromium and a careful planning to protect against detection.

3.1 Creation of Identities

We created 28 identities for our studies. User accounts for all eight inspected online services were created with each identity. We used a random identity generator for identity creation. Each identity consisted of first and last name, birthday, gender (50% male, 50% female), job title (function, company) as well as typing speed. Each identity owns an individual IP address (geolocation: TH Köln) and a personal computer (virtual machine running Ubuntu Linux 16.04 LTS). We conducted a one month pilot phase with one identity in order to optimize our identity creation, training and testing automation. Afterwards, we started the automated training and testing with the remaining 27 identities. The account creation for Facebook required some extra care, as RBA is not activated per se for all accounts [17]. We manually conducted extra training to these accounts (e.g. friend requests) prior to the studies. Resulting of the higher effort, 14 Facebook accounts (5 male, 9 female) were created. Six accounts (4 male, 2 female) were suspended during training because of "suspicious" activities. Since female accounts had higher success rates in terms of accepted friend requests or messages, we preferred them in Facebook account creation. Thus, in total we created 224 accounts of which 210 remained available for training and 204 for inspecting the targeted online services.

3.2 Training of Identities

Each online service was trained with individual user activities for each identity in a 3.5 month period between December 2017 and March 2018. Each identity executed 20 user sessions lasting between 1.5 and 2 hours within a training

period of 2 or 4 weeks. The start of the browsing sessions varied randomly between two time spans (9:00 - 9:30 AM, 1:00 - 1:30 PM) to mitigate possible automation detection by the online services. For further mitigation, the identities were created iteratively in small batches of three to four identities per week.

We developed individual automated user activities for each online service. Activities include the login process, actions on the online services at logged in state (*user action*) and the logout process. In the login process, our user opens the targeted online service in a new browser tab, enters its login credentials and accesses this service. We considered typical user activities for the user actions, e.g. scrolling in the news feed or browsing on the online service. These actions included randomness and fine-grained variations to avoid being spotted as a "scripted human". Also, the user behavior differed between genders. For the logout, our user logs out of the online service and closes the tab.

We simulated browsing activities on other websites in separate tabs, as online services may track this browsing behavior [7]. Users visited a search engine and looked for current events in local media. They followed some of the links in the search results and "read" the website's content by scrolling and waiting.

These activities were conducted inside browsing sessions. Each session was initiated with an empty browser history including cookies and local cache. The cookies were retained inside each browsing session. Afterwards, the testing sequence of online services was shuffled to a random order. We did this to prevent that our user logs into online services at the same time throughout the study.

3.3 Implementation of RBA Inspection System

The implemented RBA inspection system is based on the browser Chromium 64.0.3253.3. For browsing automation, the library Puppeteer 0.13.0 is used. The obtained observations during the test phase are stored in a MongoDB log.

Chromium was operated in a custom *headful mode* (browser is launched with visible graphical user interface inside a virtual window session). We used the headful mode to avoid detection of our automated browsing. When Chromium is executed in *headless mode*, which is specifically designed for browsing automation, a number of differences in Chromium's behavior allow websites to detect the automation mode [28]. In fact, during pilot testing we experienced situations in which inspected online services treated a browser in headless mode differently.

Furthermore, we modified the Chromium source code to minimize possible detection of our automated RBA inspection system.

We implemented the user automation framework using Puppeteer, a library to control Chromium. We found that several of the provided automation functions can be detected by online services. The constant delay in the standard Puppeteer key typing function is used to detect automated input. We therefore modified and enhanced several Puppeteer library functions to mimic human behavior more closely: (i) We added randomized delays between pressing and releasing key buttons as well as consecutive button presses. (ii) We adjusted the default mouse input behavior of clicking on the exact center of a specified element by selecting a random click point in the center quarter of the element. Moreover,

Table 1: Comparison of possible RBA features (bold: selected for the studies)

Feature	RBA references (except [3])	Distinguishing info [3]
IP address[#]	[11, 12, 8, 2, 16, 26]	High
User agent string	[11, 16, 25]	High*
Language	[11, 16, 8]	High*
Display resolution	[10, 25]	High*
Login time	[11, 16, 12, 25, 8]	Low[+]
Evercookies	[16]	Very high
Canvas fingerprinting	[10, 20, 26]	Medium
Mouse and keystroke dynamics	[27]	- (*Low for scroll wheel fingerprinting*)
Failed login attempts	[16, 26]	-
WebRTC	-	Medium
Counting hosts behind NAT	-	Low
Ad blocker detection	-	Very low

[#] Includes IP based *geolocation*.
* Refers to *major software and hardware details*
[+] Refers to *system time and clock drift*. Alaca and van Oorschot did not consider the login time. Hurkala and Hurkala [16] estimated a medium risk level for unusual login times.

the default time between pressing and releasing the mouse button of zero was replaced with a more realistic randomized click time. (iii) We implemented a scrolling function to imitate human-like reading of website contents.

We integrated external services providing CAPTCHA solving capabilities in order to allow our RBA inspection system to operate fully automated.

3.4 Inspection of RBA Features

A wide variety of features can be used for RBA deployments, ranging from browser provided information to network information [27, 3]. To reduce complexity, we selected five features based on the number of mentions in literature and the evaluations in [3] in terms of highest *"distinguishing info"* (see Table 1).

We selected the features *IP address*, *user agent string*, *language*, *login time* and *display resolution* for our investigations.

Canvas fingerprinting and evercookies provide a high level of information [3, 10, 1]. Canvas fingerprinting can be seen as a more robust and fine-grained version of user agent strings. Evercookies can uniquely identify a device. Since both features are considered as harder to fake, they add a high level of trust, possibly bypassing RBA security mechanisms. Since we aimed to test the "uncertain" area in terms of RBA risk scores, we did not consider both for our studies.

Prior to the study design, we estimated possible risk score results for specific variations of feature values. We used these estimations to design the final studies. Since no public information on the analyzed RBA implementations was known, we considered three publications [11, 16, 8] as a baseline for the estimation. We made use of the maximum possible range of ratings. However, since IP addresses are considered as more spoofing resistant than the other features [3], we expect this feature to be weighted highest inside the black box RBA implementations.

We assume that the *IP address* risk score increases with both geographical distance towards the usual values and changes in IP address and internet service provider (ISP). Since users are more likely moving in their current region, we expect the risk score to be *medium* at a maximum inside the same country. We assume changes in continents to be more unusual, so we expect a *high* risk score in that case. We rated the risk score for IP addresses of the anonymization service *Tor* as *unknown* for two reasons: (i) Tor exit nodes (and Tor users) can be identified through a public list. Thus, one publication [16] estimated a high risk score for Tor. (ii) Facebook explicitly supports Tor. Hence, lower risk scores can also be possible. We subdivided the *user agent string* into *browser*, *operating system* (OS) and *version*. We expect users to switch browsers more likely than the OS, which is why we weighted browser changes lower than those in the OS. For the remaining three features, we assume that changes in one or more parameters will increase the score equally.

4 Studies

In this section, we describe the setup and results of the studies we conducted to evaluate the eight analyzed online services for their RBA behavior. We conducted two studies. In the first one, we tested how the online services reacted to six different variations of IP addresses to reduce the number of required test conditions for the second study (see Section 4.1). In the second and main study we then determined which of the five investigated features (see Table 1) play a role in RBA decision-making (see Section 4.2). We tested all possible combinations of these features for each online service and observed the results. We did this to determine whether a certain feature was included in the online service's feature set and to ascertain how a particular feature was weighted in the online service's RBA decision-making. Finally, we also did several activities on user accounts so that online services might offer diverse selections of additional authentication factors. We did this to capture as many additional authentication factors applied by the targeted online services as possible (see Section 4.3). An extended version of our results including all captured dialogs can be found online [31].

4.1 Study 1: Determining IP Feature Thresholds

The feature space that can be used for RBA is huge, and even with the restrictions put forth in Section 3.4 the search space is still too large for the type of study we envision. Even the particularly important IP address feature has a wide range of possible values. Possibly interesting variations range from dynamic IPs (same ISP, same geolocation) or different access points (work, home, mobile) at similar locations, to national or international travelling or Tor (see Table 2). Thus, in a first step we treated the IP space separately and tried to find thresholds for the individual online services that are close to the decision boundary of the decision procedure. This will simplify the subsequent experiments and reduce the number of required probes.

Methodology In this first study varied the IP address only. We equipped seven of the trained identities with new IP addresses (Table 2). Probe 0 uses the identical IP from which the online services were trained before. Probe 1 and probe 2 are located in close vicinity of the training IP (same city, physical distance less than 1 km), where probe 1 is from the same ISP (a university) and probe 2 is from a different ISP. Probes 3 to 5 used IPs with an increasing distance from the training origin. We used VPN tunnels through Amazon Web Services (AWS) instances for these probes. Probe 6 used the Tor network, with an IP of the exit node that is potentially known by service providers and sometimes treated differently. Logins at all online services were conducted with the new IP address and reactions of the online services were recorded.

Results The obtained results are depicted in Table 3. We see that the thresholds seem to be at IP variation probe 4 (Google, Amazon, LinkedIn) and probe 1 (GOG.com). Facebook, Steam, Twitch and iCloud did not request additional authentication factors, if only the IP address was varied. A CAPTCHA inside the Steam login form was visible in probe 6 (Tor). A reCAPTCHA on the Twitch login form was not displayed in probe 2 (Netcologne) while being visible vice versa. These might rather be signs for blacklisting (Steam) or whitelisting (Twitch) than for RBA. Google sent an email containing a security alert on two occasions before reaching the threshold of asking for additional authentication factors.

Based on the results, we extracted three IP settings for use in the subsequent experiments. These were selected for each online service separately, reflecting the determined thresholds. We set probe 0 (TH Köln) for GOG.com, probe 3 (Frankfurt) for Google, Amazon and LinkedIn as well as probe 5 (Oregon) for Facebook, Steam, Twitch and iCloud. We did not use Tor in subsequent studies, due to its unpredictable nature (frequent variations of IP addresses) which could produce unreliable results. Varying the ISP to AWS (probe 3) inside the same country did not result in requesting additional authentication factors. Hence, we assume that using AWS IP addresses did not affect the reliability of our results.

4.2 Study 2: Examining RBA Usage

In the second and main study, we determined which features play a role in the overall RBA decision-making and under which circumstances the inspected online services request additional authentication factors.

Table 2: Setup of study 1 to determine the RBA triggering threshold for the IP feature

	IP	ISP	Geolocation	Description
probe 0	fixed	TH Köln	Cologne, Germany	same IP as used during training
probe 1	fresh	TH Köln	Cologne, Germany	fresh IP in the same building
probe 2	fresh	Netcologne	Cologne, Germany	different provider in the same city
probe 3	fresh	AWS	Frankfurt, Germany	same country, different provider
probe 4	fresh	AWS	Paris, France	same continent, different provider
probe 5	fresh	AWS	Oregon, USA	different continent
probe 6	fresh	random	random (Tor exit node)	Tor exit node at random location

Table 3: Results of study 1 showing the determined RBA triggering thresholds for the IP feature (bold lines).

IP variation	Identity	Facebook	Google	Amazon	LinkedIn	GOG.com	Steam	Twitch	iCloud
probe 0 (TH Köln, fixed)	*All identities*	-	-	-	-	-	-	-	-
probe 1 (TH Köln, fresh)	IDA, IDAA^{+}	-	-	-	-	A	-	-	-
probe 2 (Netcologne)	IDB	-	S	-	-	A	-	Ø	-
probe 3 (Frankfurt)	IDC	-	S	-	-	A	-	-	-
probe 4 (Paris)	IDD	-	A	A	A	A	-	-	-
probe 5 (Oregon)	IDE	-	A	A	A	A	-	-	-
probe 6 (Tor)	IDF	-	A	A	A	A	O	-	-

A: Additional authentication factors requested
S: Security alert submitted (via email)
- : No RBA triggered
O: CAPTCHA displayed before login
Ø: reCAPTCHA not displayed before login
$^{+}$: Facebook login was conducted with this identity

Methodology We tested all 31 possible combinations of the five parameters *IP address*, *user agent string*, *language*, *time parameters* and *display resolution* for triggering RBA. Each trained account conducted one or two login attempts with different parameter combinations. The *IP address* was chosen one step beneath the determined RBA triggering threshold. The remaining parameters were chosen to represent the highest possible risk estimation as defined in Section 3.4 (see Table 4). We chose a far distance country with a different national language than in the training country as the testing country. Based on the online services' behavior of all 31 parameter combinations, we are able to derive possible feature set parameters.

Results *Google* sent a security alert via email when either of the features *IP address*, *user agent* or *resolution* changed (see Table 5). Changes in one of the features *language* and *time* didn't result in a warning instead. In contrast to that, we have seen before that strong variations of the IP address result in a request for additional authentication factors (see Table 3). When modifying two features, all combinations resulted in a security warning, except for the combination of *language* and *time*. Modifying three features resulted at least in a security warning, and the combination of *IP address*, *user agent*, and *time parameters* led to an additional authentication factor requested. Concluding all results, our derived Google feature set contains *IP address* (highest weighting), *time parameters* (lower weighted than IP), *user agent* and *resolution*.

LinkedIn's RBA was triggered with combinations of *IP address* and at least one of the other parameters (see Table 6). Thus, LinkedIn's feature set comprises

Table 4: Setup of study 2 showing the probed features. We tested all possible combinations, i.e., $2^5 - 1 = 31$ variations per online service.

	Neutral/Training	Testing
IP address	as in training	as determined in Sect 4.1
User agent	Chrome/Linux	Firefox/Windows 10
Languages	de-DE,de,en-US,en	es-MX,es,en-US,en
Time Timezone	UTC+1 (Europe/Berlin)	UTC-6 (Mexico/General)
Login times [UTC+1]	9:00 AM - 2:30 PM	0:00 - 1:00 AM
Display resolution	1366x768	1280x1024

Table 5: Results of Study 2 for Google modifying a *single feature* (left), *two features* (middle), and *more than two features* (right).
(A: Additional authentication factors requested, - : No RBA triggered, S: Security alert, C: Critical security alert)

	Result
IP address	S
User agent	S
Language	-
Time	-
Resolution	S

	IP	**UA**	**L**	**T**	**R**
IP address		S	S	S	S
User agent	S		S	S	S
Language	S	S		-	S
Time	S	S	-		S
Resolution	S	S	S	S	

IP	**UA**	**L**	**T**	**R**	Result
X		X	X		S
	X	X	X		S
		X	X	X	S
X	X		X		A/C
X	X	X	X		A/C
X	X		X	X	A/C
X	X	X	X	X	A/C

Table 6: Results of Study 2 for LinkedIn modifying a *single feature* (left) and *two features* (right).
(A: Additional authentication factors requested, - : No RBA triggered)

	Result
IP address	-
User agent	-
Language	-
Time	-
Resolution	-

	IP	**UA**	**L**	**T**	**R**
IP address		A	A	A	A
User agent	A		-	-	-
Language	A	-		-	-
Time	A	-	-		-
Resolution	A	-	-	-	

IP address, *user agent*, *language*, *time parameters* and *resolution*. The IP address seems to be higher weighted since it triggered RBA in the prior study alone.

Facebook seems to have RBA deactivated by default. We could not trigger RBA on accounts having at least 50 connections to other accounts (friends). However, we could trigger RBA on two female accounts having both 40-50 friends and a high interaction rate based on received friendship requests and messages from other users. Due to the possible dissimilarities between the test accounts (RBA enabled or disabled), we cannot deduce the exact feature set here. However, our results show that Facebook requested additional authentication factors when at least *IP address*, *user agent* and *resolution* were changed.

On *Amazon* and *GOG.com* we could not trigger RBA with more or other parameters than the IP address. Thus, their derived feature sets contain only the *IP address* of our probed features.

The remaining online services *Steam*, *Twitch* and *iCloud* did not show any reaction in both studies. Possible reasons for this behavior could be: (i) RBA was not implemented or not activated by the user behavior. (ii) Other features than the five tested were rated as more important. (iii) An internal warning was triggered informing operational staff about suspicious behavior.

4.3 Study 3: Analyzing Additional Authentication Factors

With RBA being triggered, additional authentication factors are requested by the respective online service. Depending on internal account settings, online services might vary the set of requested additional authentication factors. Overviews of neither the additional authentication factors nor the corresponding RBA user

Table 7: Captured additional authentication factors
(*: Authentication factor was offered in all tested parameter variations)

Service	Requested authentication factors
Facebook	Approve login on another computer* Identify photos of friends* Asking friends for help* Verification code (text message)
Google	Enter the city you usually sign in from Verification code (email, text message, app, phone call) Press confirmation button on second device (tablet, smartphone)
LinkedIn	Verification code (email)*
Amazon	Verification code (email*, text message)
GOG.com	Verification code (email)*

interfaces in current practice were published in literature to date. For this reason, we tried to capture as many variations as possible. In order to achieve this, we added a mobile phone number, a smartphone or tablet as a second device and did additional user actions (e.g. writing a private message with phone number included). We triggered RBA on desktop and mobile devices with all possible combinations and monitored the demanded authentication factors (see Table 7).

5 Discussion

According to our findings, all tested RBA-instrumented online services used the *IP address* in their feature sets. Most online services also used additional features as *user agent* or *display resolution*. All tested online services offered verification codes as an additional authentication factor. The test results confirmed our hypothesis that online services rated the *IP address* higher than other parameters.

Facebook's verification code feature leaked the full phone number. We consider this as a bad practice and a threat for privacy. In so doing, phone numbers of users can be obtained. Also, attackers can call the number and gain access to the verification code by social engineering. We are convinced that such a RBA solution will not mitigate incentives for credential stuffing or online password guessing attacks. Thanks to the prompt reaction by Facebook, this vulnerability is now fixed: We contacted Facebook about the phone number leak on September 4th, 2018. Facebook resolved the issue on September 6th, 2018. Since this issue seemingly remained undiscovered by Facebook before our disclosure, this underlines the demand for more research on RBA to improve its overall security.

5.1 Derived RBA Models Applied in Practice

Based on our findings, we are able to derive three distinct types of conceptual RBA models. Note that due to the abstract nature of these models, they do not provide implementation details.

The **Single-Feature Model** relies on a single feature only. The password authentication process is extended to search for an exact match of the IP address in the IP address history of the user. If there is no such match, additional authentication steps are requested. We assume that GOG.com adopted this model. This model is easy to implement, since only one feature has to be stored and evaluated. Thus, a minimum of sensitive data has to be collected and stored. However, this approach entails potential usability problems. Since IP addresses might change frequently in time [3], this can result in frequent re-authentication. Hence, we do not consider this as a sensible RBA solution for practical use.

The **Multi-Features Model** extends the single-feature model. It derives additional features from the IP address. These are evaluated together with additional features in a scoring model, which compares the current feature values with the authentication history. Depending on the resulting risk score, multiple types of actions are performed (e.g. sending security alerts or requesting additional authentication factors). According to our observations this model was adopted by Google and—in slightly more simplified form without security alerts—by Amazon and LinkedIn. This model has the potential to increase usability compared to the single-feature model since additional authentication factors can be requested less frequently. However, attackers are possibly able to learn about the RBA implementation based on detailed information delivered in security alerts.

The **VIP Model** protects only special users. Depending on the user's status (e.g. important or not important), RBA is active or inactive. We assume that Facebook used this model. This procedure will make it harder for attackers to gain information about the used RBA implementation. However, if such a mechanism is known, attackers are able to find out whether an account is considered as important by the online service (which is the case when RBA is triggered). Also, this model puts some users at risk since it does not protect all users.

5.2 Limitations

We were able obtain a high amount of information with the described studies. However, the RBA behavior could only be determined from visible reactions disclosed by the online services. Hence, we can only estimate internal weightings for features. It is still possible that the real weightings might vary in detail. In addition, RBA is required to be activated anytime for determining feature sets accurately. It is still possible that online services (additionally) use other features which were not tested in the studies (e.g. canvas fingerprinting).

Although we took a lot of care of not being detectable as an automated user, we cannot fully exclude that the inspected online services identified our identities as non-humans. Judging some of the hints we obtained during our pilot phase, we are strongly convinced, though, that our investigations remained under respective detecting thresholds.

5.3 Ethical Considerations

It is commonly found that tools and techniques used for security analysis are "dual use", i.e., can be used for illegitimate purposes as well. We believe our work

is justified, as the expected security gain (from broader adoption of RBA) outweighs the expected security implications. Furthermore, we designed our study to keep the potential impact on the server infrastructure minimal. Finally, we followed the principle of responsible disclosure.

6 Conclusion

RBA is becoming more and more important to strengthen password-based authentication without affecting the user interface at the same time. As RBA is still in its infancy, it is of paramount importance that RBA approaches and implementations are rigorously analyzed following common scientific policies. Unfortunately, almost all early adopters of RBA restrain their approaches and experiences, preventing the required scientific dialogue and the widespread adoption. To close this information gap, we developed distinct studies enabling to verify whether a particular online service adopted RBA. Moreover, we were able to determine the underlying feature sets and requested authentication factors.

We can confirm the general trend in RBA of using the IP address as a high weighted indicator to determine risks of login attempts. Some services also used additional lower weighted indicators (e.g. user agent). Furthermore, verification codes are currently the unwritten standard for additional RBA authentication factors. Our research disclosed potential vulnerabilities and usability problems on specific RBA implementations (one vulnerability was fixed after we contacted the company in charge). Since RBA usually evaluates sensitive data, there is need for more open research on this technology to mitigate such potential risks.

Acknowledgement This research was supported by the research training group "Human Centered Systems Security" (NERD.NRW) sponsored by the state of North-Rhine Westphalia.

References

1. Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., Diaz, C.: The web never forgets: Persistent tracking mechanisms in the wild. In: CCS'14. pp. 674–689. ACM (2014)
2. Akhtar, N., ul Haq, F.: Real time online banking fraud detection using location information. In: Proc. CIIT 2011, pp. 770–772. Springer (2011)
3. Alaca, F., van Oorschot, P.C.: Device fingerprinting for augmenting web authentication. In: Proc. ACSAC '16. pp. 289–301. ACM (2016)
4. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: 2012 IEEE Security & Privacy. pp. 538–552. IEEE (May 2012)
5. Bonneau, J., Felten, E.W., Mittal, P., Narayanan, A.: Privacy concerns of implicit secondary factors for web authentication. In: WAY Workshop (2014)
6. Bonneau, J., Herley, C., van Oorschot, P.C., Stajano, F.: Passwords and the evolution of imperfect authentication. Comm. ACM **58**(7), 78–87 (Jun 2015)
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), 1476–1510 (2017)

8. Cser, A., Maler, E.: The Forrester Wave: Risk-Based Authentication, Q1 (2012)
9. Das, A., Bonneau, J., Caesar, M., Borisov, N., Wang, X.: The tangled web of password reuse. In: NDSS'14. vol. 14, pp. 23–26. San Diego, CA (Feb 2014)
10. Daud, N.I., Haron, G.R., Othman, S.S.S.: Adaptive authentication: Implementing random canvas fingerprinting as user attributes factor. In: ISCAIE. pp. 152–156. IEEE (2017)
11. 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)
12. Golan, L., Orad, A., Bennett, N.: System and method for risk based authentication (Oct 2013), US Patent 8,572,391
13. Google: Notifying Android users natively when devices are added to their account. Online (2016), https://gsuiteupdates.googleblog.com/2016/08/notifying-android-users-natively-when.html
14. Grassi, P.A., Fenton, J.L., Newton, E.M., Perlner, R.A., Regenscheid, A.R., Burr, W.E., Richer, J.P., Lefkovitz, N.B., Danker, J.M., Choong, Y.Y., Greene, K.K., Theofanos, M.F.: Digital identity guidelines. Tech. Rep. NIST SP 800-63b (2017)
15. Herley, C., Schechter, S.: Distinguishing attacks from legitimate authentication traffic at scale. In: NDSS'19. San Diego, CA, USA (2019)
16. Hurkala, A., Hurkala, J.: Architecture of context-risk-aware authentication system for web environments. In: Proc. ICIEIS2014. Lodz, Poland (Sep 2014)
17. Iaroshevych, O.: Improving second factor authentication challenges to help protect Facebook account owners. In: SOUPS 2017. Santa Clara, CA, USA (Jul 2017)
18. Johansson, J., Canavor, D., Hitchcock, D.: Risk-based authentication duration (Mar 2014), US Patent 8,683,597
19. Milka, G.: Anatomy of Account Takeover. In: Enigma 2018. USENIX (Jan 2018)
20. Molloy, I., Dickens, L., Morisset, C., Cheng, P.C., Lobo, J., Russo, A.: Risk-based Security Decisions Under Uncertainty. In: CODASPY'12. pp. 157–168. ACM (2012)
21. Morris, R., Thompson, K.: Password security. Commun. ACM **22**(11), 594–597 (Nov 1979)
22. Petsas, T., Tsirantonakis, G., Athanasopoulos, E., Ioannidis, S.: Two-factor authentication: Is the world ready? In: EuroSec'15. pp. 4:1–4:7. ACM (2015)
23. Quermann, N., Harbach, M., Dürmuth, M.: The state of user authentication in the wild. In: Who are you? Adventures in Authentication Workshop 2018 (Aug 2018)
24. Shepard, L., Chen, W., Perry, T., Popov, L.: Using social information for authenticating a user session (Dec 2014)
25. Spooren, J., Preuveneers, D., Joosen, W.: Mobile device fingerprinting considered harmful for risk-based authentication. In: EuroSec'15. pp. 6:1–6:6. ACM (2015)
26. Steinegger, R.H., Deckers, D., Giessler, P., Abeck, S.: Risk-based authenticator for web applications. In: Proc. EuroPlop '16. pp. 16:1–16:11. ACM (2016)
27. Traore, I., Woungang, I., Obaidat, M.S., Nakkabi, Y., Lai, I.: Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments. In: Proc. ICDH 2012. pp. 138–145. IEEE (Nov 2012)
28. Vastel, A.: Detecting Chrome headless. Online (2018), https://antoinevastel.com/bot%20detection/2018/01/17/detect-chrome-headless-v2.html
29. Wang, D., Zhang, Z., Wang, P., Yan, J., Huang, X.: Targeted online password guessing: An underestimated threat. In: CCS'16. pp. 1242–1254. ACM (2016)
30. Wang, X., Kohno, T., Blakley, B.: Polymorphism as a defense for automated attack of websites. In: ACNS. pp. 513–530. Springer International Publishing (2014)
31. Wiefling, S., Lo Iacono, L., Dürmuth, M.: Risk-based Authentication website. Online (2019), https://riskbasedauthentication.org