The State of User Authentication in the Wild

Nils Quermann
Ruhr University Bochum
Bochum, Germany
nils.quermann@rub.de

Marian Harbach
Audi AG
Ingolstadt, Germany
marian.harbach@audi.de

Markus Dürmuth
Ruhr University Bochum
Bochum, Germany
markus.duermuth@rub.de

ABSTRACT

User authentication is a field under active research, both from the academic community, startups, and established companies. In this context, passwords are regularly declared "dead", and there is a strong desire to replace passwords as the most widely used authentication method. In this paper we intend to map the current state of user authentication, as typically seen by end users. We evaluated the mechanisms used by 48 different services, including websites, IoT/smart home devices, and mobile devices.

Our main findings are: (i) Passwords are still the most prevalent primary authentication method in the wild. (ii) Most services support 2FA, either requested in combination with the first factor (immediate 2FA) or when a more sensitive operation is triggered (delayed 2FA). (iii) No service offered a simple way to recover a second factor, without having another second factor set up. Based on the findings, we derive recommendations for a more comprehensive perspective on user authentication.

1. INTRODUCTION

User authentication is an essential requirement for a number of services, ranging from websites holding personalized services and sensitive data, over mobile devices and specifically smartphones holding a host of sensitive information, to smart homes and the Internet of things. Password-based authentication is widely used, as it is easy to implement and easy to understand for users. However, human-chosen passwords are far from being secure as they typically have low entropy and bad memorability, are often re-used, and more. Consequently, there is a substantial research effort to improve user authentication, both from the academic community, startups, and established companies. Passwords are declared "dead" regularly,[1] and radical replacements are sought.

From the users' perspective, however, it seems obvious that passwords are far from dead, and are widely used instead. At the same time, there are a several application scenarios were alternatives are used: (i) mobile devices, where biometric authentication as well as graphical passwords are predominately used [10, 5], and (ii) high profile accounts, where often two-factor authentication (2FA) using a password and a physical token are used. We find that there is no common understanding of how authentication is presented to users as a big picture. Having such an overview will help to come up with solutions that can fit users needs and facilitate understanding when interacting with different kinds of devices or services.

In this short paper we aim to map out the state of user authentication in the wild to fill this gap. We concentrate on user authentication as encountered by average users, on services such as websites, mobile devices, IoT devices, and laptop computers while also included lifecycle processes, such as setup and recovery.

1.1 Related work

Research on specific aspects of user authentication is plentiful. The state of the art includes work on passwords and the corresponding policies, 2FA in general, lock screens for mobile phones as well as myriads of more or less practical proposals to replace existing mechanisms. Covering the practical security of password-related mechanisms used for authentication, previous work has addressed password strength (e.g. [1, 13, 15]), corporate (e.g. [12]) and end-user password policies (e.g. [16, 14, 7]), password reuse (e.g. [4]) and the effective entropy of security questions for fallback authentication (e.g. [2, 9]). Beyond that, numerous proposals have been made to replace passwords, for which Bonneau et al. [3] provides an excellent overview and framework.

Comparable to the use of passwords, the authentication landscape for mobile devices (i.e. lock screens) is well explored [10, 11, 6, 5, 17]. Having a choice betweens PINs, passwords, graphical passwords and biometrics, smartphones provide the users with the most diverse options to authenticate. However, we are not aware of any related work that tries to paint a more comprehensive picture of all digital consumer authentication mechanisms commonly in use today.

2. ANALYZING USER AUTHENTICATION

Authentication factors never work in isolation, but need to be set up, used, and potentially recovered in varying contexts. Users also do not interact with them in isolation, but encounter various forms of authentication during a typical day or even within a single interaction. It is therefore pos-

[1] *https://www.cnet.com/news/google-security-exec-passwords-are-dead/*

USENIX Symposium on Usable Privacy and Security (SOUPS) 2018.
August 12–14, 2018, Baltimore, MD, USA.

sible that forms of authentication that diverge from existing patterns influence the usability and thus security of the overall authentication process. Also, best practices from one kind of authentication scheme could inspire improvements to another. For example, a combination of authentication factors may work well for one service that is mainly web-based, while it becomes unusable for an IoT device. We therefore chose to provide an overview of the state of the art in user authentication across services and devices many users come into contact with today, identifying patterns and gaps.

Methodology In our analysis, we examine a set of online services and devices most users are (to some extent) familiar with. To begin to understand how existing services set up their authentication schemes, we examine the following properties and their interplay:

- Whether or not *3rd party logins* are supported to bootstrap account creation.
- The *account identifier* a user often needs to remember in order to log in.
- The *primary authenticator* a user is asked to provide during authentication.
- The *secondary authenticator* supported to increase account security.
- The *effort required* to set up additional authenticators.
- The *recovery process* when access to one or more authenticators is lost.

To collect data about the support and implementation of authentication mechanisms, we either created a user account or gathered the information from support documents. We also consulted additional resources to supplement the information accessible at the service itself: *turn it on*[2] and *Two Factor Auth*[3]. Services that did not allow us to create an account and did also not provide public information about their authentication scheme were excluded.

For most properties, we gathered all available variations and entered them into Table 1. In order to arrive at a pragmatic estimate of the effort for setting up a second factor, we settled on counting the clicks necessary for enabling SMS codes from the home page of the service until the factor was successfully activated. While counting clicks can only be a rough estimate, it can approximate how much time the setup takes and how easily the feature can be found. We focused on SMS codes for this measure, as setup complexity varies between different 2F types and SMS codes were the most widely supported second factor in our sample. Data was collected in October 2017 from a location in Germany.

Sampling End users currently experience explicit, digital authentication in two main contexts. First they experience authentication with a variety of online services (websites and apps), ranging from email and social networks over work-related to banking and government-related services. Second, they authenticate to physical devices such as smartphones, tablets, computers, or other IoT devices such as smart door-locks. The online services we evaluated were inspired by the set studied by Florêncio and Herley [7]. More precisely, we focused on five categories, namely *top traffic sites*, *banks*, *universities with top CS departments*, *government sites* and *other sites* in order to capture a wide variety usage contexts that are relevant to a large audience as well as newly emerging trends. Going beyond the list from Florêncio and Herley, we added some of the largest European banks in terms of total assets,[4] as well as several typical devices. For mobile devices and (laptop) computers we included several of the most widely used operating systems. For IoT/smart home applications, we focused on the most established products.

Limitations The set of services and devices users authenticate to is large and diverse, but is likely too small to capture all possible implementations. Still, we believe the results can give a reasonable overview over widely deployed practices. We only consider user-facing elements of the authentication system. Some services implement additional measures such as risk-based authentication, transaction monitoring, or continuous authentication (cf. [8]). These are typically not visible to the user and thus have no (direct) impact on usability. Furthermore, documentation and information on some websites may be outdated, available options may depend on the country the service is accessed from, or there is no public documentation and it is (for example due to geographical restrictions) not possible to create some user account for our own research.

3. RESULTS

Table 1 outlines our results, we discuss the findings in the sequel.

3.1 3rd Party Logins

We saw only a handful of services supporting 3rd party logins, which enables users to use a proof of access to an already existing account (we only saw Google and Facebook as identity providers) as an authenticator. This supposedly increases usability, as only one password is required, but potentially raises privacy issues. Interestingly, we observed that users are able to supplement an existing 2FA at the 3rd party with another secondary factor with the target service (e. g. Dropbox).

3.2 Primary Authenticator

In our analysis, we recognized one recurring pattern when it comes to the primary authentication mechanism. In all cases users are asked to provide some sort of identifier along with a shared secret. While the shared secret almost always is a password, we identified three different identifiers used: *usernames* are the most common identifier (69 %), followed by email addresses (45 %) and in rare cases phone numbers (5 %) – with some services supporting multiple identifiers (cf. Table 1). We find that, despite their problems, passwords are still the most used authenticator for websites. This is similar to the conclusion of Bonneau et al. [3], where several password alternatives were evaluated, none of which offered better properties than passwords in every analyzed aspect.

3.3 Secondary Authenticator

We found that there are seven main types of secondary authenticators used across the analyzed websites. These include SMS codes, software code generators, offline codes, smartphone notifications, U2F keys, key fobs and biometrics. With this variety of different secondary authenticators available today, we were curious to identify patterns in which

[2] *https://www.turnon2fa.com/*

[3] *https://twofactorauth.org/*

[4] *https://www.relbanks.com/top-european-banks/assets*

Service			Primary Authenticator		Additional Authenticators								Recovery			
		Identifier	3rd Party Login	Password	SMS Code	Code Generator	Offline Codes	Smartphone Notification	U2F Key	Key Fob	Biometric	Clicks required for setup	Message send to Email	Additional Factor(s) required	Personal Identification	Account Specific Questions
Top Traffic Sites	Google	@ 👤		✓	✓	✓	✓	✓	✓		(✓)	9	✓	✓		✖
	Amazon	@		✓	✓	✓						8	✓			
	Facebook	@ ☎		✓	✓	✓	✓	✓	✓			6	✓	✓		
	Ebay	@ 👤		✓		✓				✓		?	✓			
	Wikipedia	👤		✓								?	✓			
	Twitter	@ ☎ 👤		✓	✓	✓		✓				9	✓			
	Yahoo	@ 👤		✓	✓							5		✓		
	Reddit	👤		✓								?	✓			
	Yelp	@	f	✓								-	✓			
Universities with top CS Dep.	MIT	👤		✓	✓	✓		✓	✓			?			✓	✓
	Stanford	👤		✓	✓	✓	✓	✓		✓		?				✓
	UC Berkeley	👤		✓	✓	✓		✓				?	✓			
	CMU	👤		✓		✓		✓		✓		?			✓	✓
	UIUC	👤		✓		✓		✓		✓		?	✓		✓	
	Cornell	👤		✓		✓		✓		✓		?				✓
Banks	Fidelity	@ 👤		✓	✓	✓					✓	?		✓		✓
	Vanguard	👤		✓	✓							?				✓
	Schwab	👤		✓		✓				✓		?				✓
	Bank of America	👤		✓	✓					✓		7			(✓)	
	Chase	👤		✓	✓					✓		0			(✓)	
	PayPal	@		✓	✓	✓		✓		✓		5	✓	✓		✓
	Deutsche Bank	👤		PIN	✓	✓	✓					0			✓	
	Credit Agricole	👤		✓			✓			✓		0			✓	
	HSBC Group	👤		✓		✓				✓	✓	7				✓
	BNP Paribas (Consors)	👤		✓	✓		✓					0			✓	
	Santander	👤		PIN	✓					✓		0		✓		
	ING-DiBa	👤		PIN+✓		✓	✓					0		✓	✓	
	Sparkasse	👤		✓	✓		✓	✓				0			✓	
	Bitcoin.de	@		✓	✓	✓	✓		✓			4	✓	(✓)		
Government	USPS	👤		✓	(✓)							-	✓	(✓)		✓
	jobs.ca.gov	👤		✓								-	✓			
	fsaid.ed.gov	@ 👤		✓	(✓)							-	✓	(✓)		✓
	census.gov	@		✓								-				✓
	ssa.gov	👤		✓								-			(✓)	
	nasa.gov	👤		✓								-	✓			
Other Services	Github	@ 👤	G	✓	✓	✓	✓		✓			7	✓	✓		
	Dropbox	@	G	✓	✓	✓	✓		✓			11	✓	✓		
	Steam	👤		✓		✓		✓				5	✓	✓		✓
	Origin	@ 👤		✓	✓	✓						7	✓			
	Blizzard	@	f	✓	✓	✓		✓		✓		9	✓	✓		
	FIPS 201-2	👤		PIN+✓							✓	0			✓	
IoT Devices	Apple TV	@		✓	✓		✓	✓				-		✓		✓
	Kwikset Kevo	@		✓	✓							-	✓			
	August Smart Lock	@ ☎		✓	✓							0	✓			
	Danalock v3	@		✓	✓							-	✓			
	Philips Hue	🎫 - only		✓								9	revocation + creation			
	Alexa Voice Service	@		✓			(✓)				⚙	5	cf. Amazon			
	Nest Home	@		✓	✓							9	✓	✓		
	Samsung Smart TV	@		✓	✓							6	✓			
Operating Systems	Windows 8	👤		✓								-				
	Windows 10	👤		✓								-		✓		
	macOS 10.13	👤		✓								-		✓		
	Ubuntu 16.04 LTS	👤		✓								-				
	Android	(@ 👤)		✓							✓	6	no way to reset			
	iOS	(@)		✓							✓	5	no way to reset			

Table 1: Overview of user authentication across the examined services/devices. Features are either supported ✓, partially supported (✓), no longer supported ✖, in development ⚙, or not supported at all (no symbol). @: Email address, ☎: Phone number, 👤: Username, 🎫: Token based authentication, "?": Unknown.

options services chose to offer. We found that the majority of services support *SMS Codes* as secondary authenticator. We also observe a large support for *code generators* and *interaction with smartphones*.

Another notable finding is that all services but German banks did not require a secondary authenticator by default. This might be due to the sensitivity and perceived risk that is connected with a compromised banking account as well as German regulations for liability in online banking. Some services (e. g. Google and some universities) strongly encourage their users to set up some secondary authenticator. The limited adoption of *biometrics* as secondary authenticator outside of mobile devices is particularly notable. A likely explanation is that many services currently rely on the ability to log in on a wide variety of devices and thus cannot ensure that the necessary hardware is available. Additionally, biometrics often create either privacy concerns (from storing biometrics on a server) or additional effort, when reenrollment is necessary upon using another device. Finally, biometrics always suffer from being a probabilistic measure.

Delayed Step-Up Authentication German banks apply secondary authenticators in a unique manner: the second factor is only required when a sensitive operation is requested, such as transferring money. It is not required when only checking transactions or balances. In contrast, all other services we investigated require the secondary authenticator to be entered immediately after providing the primary. This delayed, step-up 2F authentication scheme offers interesting usability advantages when there are less sensitive operations that are accessed more frequently than sensitive operations.

A similar form of delayed authentication is commonly used by online shopping services. In these cases, the primary authenticator (password or biometric) is requested again to confirm the transaction. This is effectively a delayed 1F reauthentication and is most likely intended to protect against an authentication session that was inadvertently left active.

3.4 Setup of Additional Authenticators

The *setup process* of additional authenticators varies considerably in the number of clicks users have to perform. The analyzed services range from 4 to 11 clicks with a mean of 7 clicks for setting up SMS codes. Additionally, services varied in the level of detail explaining 2FA to their users. Some took it for granted that users know what 2FA is and only offer the opportunity to enable it. Others explained the risk of not activating 2FA, either during usage or as a mandatory step during registration where users are asked to further secure the account.

3.5 Primary Authenticator Recovery

The analyzed services provide three different methods for recovering a primary authenticator (also called *fallback authentication*), for example, when they forgot their identifier or password. The most common way for websites to handle recovery is to send a message to the user's email account. This message either contains a link to the website where the user may set a new password, or a code for being allowed to set a new password. This seemingly annoying feature (i. e. entering the code manually instead of simply following a link) can help to protect the user from phishing attempts where an attacker tries to obtain the code.

We saw another common method where users are asked to answer security question(s) usually created during registration. We identify two distinct types of questions: personal questions (e. g. name of a pet) and questions about the interaction with the website (e. g. date of registration, last password the user can remember). While the latter may provide better protection against attacks, it remains an open question whether this holds true in practice. Some services required to identify the user in person with a legal document (e. g. driver's license). We see this method mainly at universities and banking services, who both typically have a local office near the user or are required to strongly identify their users by law. When users have set up an secondary authenticator, they are almost always asked to provide them during recovery (e. g. providing some generated code). When services offer the opportunity to log in with a 3rd-party account, users have to use the recovery mechanism of the respective identity provider.

3.6 Secondary Authenticator Recovery

One key finding of this work is that users cannot easily recover secondary factors. When users no longer have access to a secondary factor, websites almost always require them to contact the support. We posit that this is for a lack of suitable fallback authentication, given that proof-of-possession for an email address is almost always already tied to the primary authenticator.

We identified two approaches to cope with this lack of self-service recovery: Several services encourage their user to set up additional 2F mechanisms. This enables users to fall back to, for example, SMS codes when they cannot provide a code from their code generator. While this enables self-service 2F recovery, it also puts a considerable burden on 2F users, who need to enroll for and maintain *two* different secondary authenticators. Whether or not this is hindering adoption should be subject of further research. Also, if all enrolled 2F mechanisms rely on users' smartphones, it becomes increasingly difficult to recover from a lost smartphone.

Another commonly seen method are *recovery codes*. Here, users have to print (or otherwise securely store) one or several codes that can be used instead of their secondary factor in the event of a lost authenticator. This, however, may introduce potential security risks as users either do not know *how* or do not *like to* store these codes securely — or simply lose them as they are very infrequently used. In Table 1, we counted them as a form of *Additional Factor(s) required*.

Finally, we found a number of methods used by a small number of services. In one case, users are able to use a device where they are still logged in to the account to grant access for the other device. In case of a compromised device, this will, however, enable an adversary to overcome the additional protection intended by the second factor. *Bitcoin.de* uses a notable method, where users may *permanently* remove their secondary factor *before* authentication. To maintain account security, this user cannot make any transactions until the identity has been verified again, by sending a specified amount of money from their banking account to the service, thus proving ownership of the associated bank account. Stanford University uses a method where students can provide the number printed on their student card together with their date of birth and answer one security question to reset the secondary authenticator, thus applying personal security questions to 2F recovery.

4. DISCUSSION

The presented analysis confirms several common assumptions, e.g. the unbroken prevalence of passwords, but there are a few surprising findings summarized below.

Default Authentication Uses Identifiers Unsurprisingly, *default authentication almost always* included the combination of a user-chosen password and one of three identifiers: *usernames*, *email addresses* and *phone numbers*. Some services enable users to choose which type of identifier they may want to provide. However, only very few services designed their authentication schemes to not require an identifier at all. Yet, from a user experience point of view, this can offer additional benefits (i. e. users do not have to memorize and type their identifier). At the same time, removing identifiers holds challenges in itself, as support processes or social interactions such as invitations may depend on them. This trade-off will also become more relevant once new device-based authentication standards, such as W3C WebAuthN[5], get adopted. Future work should look at opportunities and implications of removing user-chosen account identifiers.

Secondary Authentication Timing Another important finding is that all services but German banks use the second factor immediately with the primary factor. That is, the quality of user authentication is always elevated to the highest assurance level for these services. With German banks, users can authenticate with only their password first which allows them to view transactions details. Once a new transaction or another change to their account is desired, they are prompted to provide the secondary factor. This delayed second factor offers benefits in usability, as read access to the account is easier. Additional security provided by the second factor is also contextualized, making its purpose more evident to users.

Recovery of Secondary Authenticators We found that recovery of secondary authenticators is not as easy as recovering the primary authenticator. While seemingly unsurprising, it can have severe consequences if an account can no longer be accessed when the second factor is lost. The most common way of addressing this problem is to set up *two* (or more) secondary authenticators to compensate for the loss of one of those. It also allows the user to choose from a set of mechanisms, as there does currently not seem to be an unanimous industry standard. We posit that the difficulty of recovering the second factor may be a detriment to 2F adoption, as it adds to the apparent complexity of using such an authentication scheme. We believe that more studies on the topic is needed to better understand the trade-offs.

Overall, we argue that it is important to take the entire user authentication ecosystem into account when discussing its properties and designing future systems. Given the increasing importance of smartphones and the creation of new standards for password-less authentication, the overall user authentication experience will become more intertwined, as authenticating to unlock a device will also equal authenticating to online services.

[5] https://www.w3.org/TR/webauthn/

5. REFERENCES

[1] J. Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *Proc. SOUPS*, 2012.

[2] J. Bonneau, E. Bursztein, I. Caron, R. Jackson, and M. Williamson. Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google. In *Proc. ACM WWW*, 2015.

[3] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proc. S&P*, 2012.

[4] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The tangled web of password reuse. In *Proc. NDSS*, 2014.

[5] A. De Luca, A. Hang, E. von Zezschwitz, and H. Hussmann. I feel like I'm taking selfies all day!: Towards understanding biometric authentication on smartphones. In *Proc. CHI*, 2015.

[6] S. Egelman, S. Jain, R. S. Portnoff, K. Liao, S. Consolvo, and D. Wagner. Are you ready to lock? In *Proc. CCS*, 2014.

[7] D. Florêncio and C. Herley. Where Do Security Policies Come from? In *Proc. SOUPS*, 2010.

[8] D. Freeman, S. Jain, M. Dürmuth, B. Biggio, and G. Giacinto. Who are you? a statistical approach to measuring user authenticity. In *Proc. NDSS*, 2016.

[9] M. Golla and M. Duermuth. Analyzing 4 million real-world personal knowledge questions (short paper). In *Proc. International Conference on Passwords*, 2015.

[10] M. Harbach, A. De Luca, and S. Egelman. The anatomy of smartphone unlocking: A field study of android lock screens. In *Proc. CHI*, 2016.

[11] M. Harbach, E. von Zezschwitz, A. Fichtner, A. D. Luca, and M. Smith. It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception. In *Proc. SOUPS*, 2014.

[12] P. G. Inglesant and M. A. Sasse. The true cost of unusable password policies: Password use in the wild. In *Proc. CHI*, 2010.

[13] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proc. S&P*, 2012.

[14] P. Mayer, J. Kirchner, and M. Volkamer. A Second Look at Password Composition Policies in the Wild: Comparing Samples from 2010 and 2016. In *Proc. SOUPS*, 2017.

[15] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur. Measuring password guessability for an entire university. In *Proc. CCS*, 2013.

[16] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor. Designing password policies for strength and usability. *ACM Trans. Inf. Syst. Secur.*, 18(4), 2016.

[17] S. Uellenbeck, M. Duermuth, C. Wolf, and T. Holz. Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns. In *Proc. CCS*, 2013.