Hochschule Wismar
University of Applied Sciences
Technology, Business and Design
Faculty of Engineering, Department EE & CS

Master-Thesis

Security Evaluation of Multi-Factor Authentication in Comparison with the Web Authentication API

Submitted by: August 19, 2019

from: Tim Brust
born 03/31/1995
in Hamburg, Germany

First supervisor: Prof. Dr.-Ing. habil. Andreas Ahrens
Second supervisor: Prof. Dr. rer. nat. Nils Gruschka

Purpose of this thesis

The purpose of this master-thesis is to introduce, analyze and evaluate existing multi-factor authentication solutions in regards of their technical functionality, usability in web projects and potential security risk.
Those multi-factor authentication solutions are compared to the Web Authentication API in order to identify if the Web Authentication API is a suitable replacement or a complementary addition to the multi-factor authentication solutions.

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special contents, but the length of words should match the language.

Kurzreferat

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special contents, but the length of words should match the language.

Contents

1 Introduction

1.1 Problem Statement and Motivation

> »Usernames and passwords are an idea that came out of 1970s mainframe architectures. They were not built for 2016.«[1]
>
> *Alex Stamos*

The secure handling of passwords is a problem for many users. Passwords are re-used between different websites and often shared across private and work environments. This renders the (private) user data, but also business secrets at high risk. To make things worse very few people are using multi-factor authentication (MFA) nor a password manager in 2019. The majority of the users are either remembering their passwords or writing them down on a piece of paper - in cleartext.

At the same time the recorded amount of cybercrime cases is still increasing and for example phishing remains a big problem. While MFA can protect against threats such as brute force attacks or stolen credentials, but they are still affected by phishing attacks. Besides that the SMS traffic is not considered secure anymore, yet it's used by a lot of MFA.

To counter this negative trends new application programming interfaces (APIs) are emerging, for example the Web Authentication API. It's a standardized API supported in major browsers such as Chrome, Firefox or Edge that allows a secure registration, login and two-factor authentication (2FA) - all without the generation, storage and remembering of passwords by using asymmetric cryptography. The private keys are stored e.g. on external devices like USB sticks, but can be stored on built-in hardware, too and for example be protected by a fingerprint sensor.

[1] See [Col16]

1.2 Goals of this thesis

The goals of this thesis are an introduction into MFA and the different authentication factors such as »knowledge, possession and biometrics« including the technical functionality, usability in web project and respectively web browser and their security risks alongside an introduction to the Web Authentication API. The Web Authentication API is being analyzed if it's suitable as an alternative or addition to existing MFA solutions. In this connection the question has to be answered if the Web Authentication API can increase the security and user comfort and usability. Of course the security and potentials risks of the Web Authentication API need to be taken into account.

1.3 Target audience

The target audience of this thesis are technically experienced readers that have a good understanding of data security and privacy. Additionally the reader should have basic knowledge about the mathematics and functionality of algorithms like RSA, Elliptic-curve cryptography (ECC) or symmetric and asymmetric key exchange (e.g. Diffie–Hellman key exchange). Furthermore the thesis is tailored towards interested (web) developers that want to understand the pros and cons of alternative registration and logins solutions, MFA solutions and asymmetric cryptography and if the Web Authentication API suits their needs.

1.4 Delimitation of this thesis

Existing algorithms and concepts, as long as not required for the understanding of this thesis, are not explained in detail. It is not the goal of this thesis to perform a complete cryptanalysis, but to take other factors such as usability for the user, technical feasibility and web browser support into account. Different, but adjacent, technologies such as OAuth (2.0), OpenID Connect or Single Sign-on (SSO) neither are a focus of this thesis.

1.5 Approach and methodology

Initially in chapter 2 the reader is introduced into the basics of authentication. After that in the following chapters the areas

- Single-Factor-Authentication
- MFA

are explained, for example their technical functionality, and analyzed in regards of their security and potential risks and attacks such as phishing or Man-in-the-Middle (MITM) attacks.

Hereupon the Web Authentication API is introduced in chapter 6 and described in detail. The technical functionality is a key apsect of this chapter. Additionally the attacks the Web Authentication API can offer protection against are explained, but also asserted which security risks exists, too. Where suitable example source code listings are used to highlight these analysis.

In the chapter 7 the Web Authentication API is compared with existing MFA solutions. Therefore it's reviewed if the Web Authentication API can be used in conjunction or as replacement for MFA.

Concluding follows an evaluation based on the gained insights from the previous chapters with a conclusion and an outlook.

2 Basics of authentication

2.1 Methods of authentication

There are multiple different methods or forms respectively that can be used to authenticate a user against someone or something. Traditionally only knowledge, possession and trait are considered the different forms of authentication,[2] but other sources also introduce or take new methods into account like location- or time-based authentication.[3] Therefore this thesis accounts for them, too and describes the methods in the following sections briefly and a detailed analysis of the security, potential risk and threats is done in section 3.1.

2.1.1 Knowledge

The most common method of authentication is knowledge, i.e. »something the user knows«. Commonly used in information technology (IT) are passwords. Other forms of knowledge are for example a personal identification number (PIN) used e.g. banking (ATM's, credit cards) or telephony (SIM card), a passphrase, secret or recovery questions or a one-time password (OTP). The security relies on the fact that the knowledge method is considered a secret that only the user knows.[4]

2.1.2 Possession

Another form of authentication is the possession, i.e. »something the user has« (physically). The most basic example is a key for a lock. Other forms are for example a bank or ID card which can use techniques like radio-frequency identification (RFID), an onboard chip or magnetic stripes to store the information. In IT security tokens are often used, which can be a hardware (such as a YubiKey or an RSA SecurID) or software (e.g. a smartphone application) token. They can either be

[2] See TW75, p. 299; See BB17, p. 140; And08, p. 47.
[3] ZKM12; See DRN17, p. 191.
[4] See Eck14, p. 467.

disconnected, connected (e.g. via USB or as a smart card) or contactless (e.g. via near-field communication (NFC), Bluetooth Low Energy (BLE) or RFID).[5]

The security of this method relies on the fact that only a legitimate user has access to the possession factor and that no intruder has access or can for example make a copy. The biggest risk is the lost or theft of the possession, although for instance some security tokens are itself protected with a form of authentication like the fingerprint of the legitimate user.

2.1.3 Biometrics

Besides the knowledge and possession factors, another one is the biometrics. This factor is classified as »something the user is« and commonly includes the fingerprint, facial or iris scan. In theory many other characteristics like the gait, the ear, DNA or even the human odor could be used as a biometric factor.[6]
These intrinsic factors are sometimes referred to as traits or inherits, too.[7]

While it seems naturally to authenticate a person with a biometric, it also comes with a couple of challenges. Both, the false rejection rate (FRR), i.e. a user is rejected even though it's a legitimate authentication attempt and false acceptance rate (FAR), i.e. an imposter is granted access, need to be accounted for the usage. Compared to knowledge and possession factors the enrollment of the biometric and the continuous update of the sample is more complicated and expensive.[8]

On the other hand, it's more complicated to steal, share or copy this factor than the others. The usability varies because of the quality of the used biometrics module, the chosen biometric itself and the availability of the biometric.

2.1.4 Further methods of authentication

While the mentioned authentication forms above are considered a standard in the literature, other forms exist, too. Those include for example the location of the user. The location-based approach grants or denies the access based on the current location. This can either be physical (e.g. via GPS) or digital with e.g. an IP-address.[9]

[5] See Tod07, p. 24; DLE19; See Kei17, pp. 8–11.
[6] See JRN11, pp. 30–34.
[7] See DRN17, p. 186.
[8] See JRN11, pp. 18–24.
[9] ZKM12.

Yet another form is the time-based authentication. A common example is the time restricted access to a banking safe, which can only be opened at specific times of the day, it's secured by a time lock. In IT this form of authentication helps to protect against for instance phishing attacks from abroad, because the access is granted or denied based on the time and general time routines where for instance a user normally logs on.[10]

2.2 Wording differences between multi-factor, multi-step, authentication and verification

The naming of the chosen authentication or verification methods by companies is often confusing or difficult to understand. The terms used by companies varies from 2FA, often just calling it 2FA,[11] to two-step-verification, sometimes written as 2-Step Verification, too.[12].

One could argue that the different authentication factors can be reduced to a single one, e.g. that an OTP is »something the user knows«, since it relies on a secret that *could* in theory be memorized, too.
In this case the term MFA or 2FA is technically incorrect, since it's instead a multi-step authentication, because the same factor is used multiple times. However, it has to be noted that using the same authentication factor multiple times is weaker than using different authentication factors.[13]

Besides that (user) verification is a part of the authentication process this fine differentiation of verification vs. authentication and multi-step vs. multi-factor is not important for this thesis and the term MFA is used throughout.

[10] See DRN17, p. 191.
[11] Sup19a.
[12] Sup19b; Pla; Goo; Mic19.
[13] See Gri17, p. 117.

3 Single-factor Authentication

3.1 Threats

3.1.1 What you know (Password)

Just knowledge. Often weak, re-used. Meant to be remembered. One factor only. Protection by the server often not given, user's are writing it down etc.

1. re-usage
2. phishing -> stealing in general
3. secret might be known by others, too (e.g. security questions)
4. guessing
5. brute force
6. copied

3.1.2 Possession

The main risks of authentication by possession it that it's not tied to the user itself and can be lost or even worse stolen by an attacker. Besides that, possession factors can be shared between multiple users, allowing attacks such as a malicious insider attack. Often the possession factors are not protected itself so e.g. a keycard to open a door can be used by the attacker, too.
Another usage implication is that it has to be carried with the user and can be forgotten which makes the authentication impossible if no access to the possession is possible and no backup or different authentication methods are available. Yet another risk is that the possession can be damaged or destroyed. For example security keys that are carried on the keyring can be damaged when the key is dropped or exposed to liquids.[14]

[14] See Sho14, pp. 263–264.

Especially possessions that use wireless transmissions such as BLE, NFC or RFID can be copied even over some distances. For instance credit cards could be copied in crowded places such as trains or busses.[15]

3.1.3 Biometrics

In contrast to possession and knowledge the biometric trait can't be easily stolen. While it can be copied, e.g. the fingerprint from high resolution photographs or face models to circumvent face recognition systems.[16] In the recent past researchers were able to copy both German Chancellor's Angela Merkel's iris and the fingerprint of Ursula von der Leyen, the now the elected President of the European Commission, from high resolution photographs.[17]

Yet another implication is that the biometric characteristics can change over time or be temporarily unavailable because of injuries. While some can heal over time, others, especially scars, can permanently change the biometric trait and therefore render it unusable.[18]
Traits such as facial recognition also must be usable with different amounts of facial hair, hair styles, with and without glasses, etc.[19]

Another high risk is the data privacy and protection.

3.1.4 Further methods

A high risk of the location-based authentication is the spoofing of the actual location by an attacker. An attacker can choose different attack vectors such as spoofing the source IP address that tries to access a system. Another form of spoofing is the GPS spoofing where an attacker modifies the actual GPS by broadcasting false information or the called ID spoofing for e.g. usage with VoIP. Besides these techniques the most common variant remains the usage of a VPN network or DNS proxy to hide the real location.

[15] KSM14.
[16] FKH14; FSS18.
[17] Kre14.
[18] See JRN11, p. 52.
[19] See JRN11, p. 98.

For time-base authentication an attacker could use attacks against the Network Time Protocol (NTP) in order to either gain access of the verification system or to modify the synchronized time in order to allow the login attempt to succeed.[20]

3.1.5 General threats

General threat: security of transmission!

[20] See Mal+15.

4 Multi-factor authentication

In this chapter a list of different MFA solutions is described in detail.

4.1 OTP

4.1.1 HMAC

Keyed-Hashing for Message Authentication (HMAC) Code is an extension of a Message Authentication Code (MAC) and standardized in Request For Comments (RFC) and National Institute of Standards and Technology (NIST) abc.

HTOP

HMAC-based One-time Password algorithm, counter based. RFC 899. Configurable length (6-10). Default SHA1. Truncation of HMAC

TOTP

Time based instead of counter based. RFC 123 and Initiative For Open Authentication (OATH).

pros

1. Collisions in MD5 or SHA1 are no problem, already stated/analyzed in the RFC

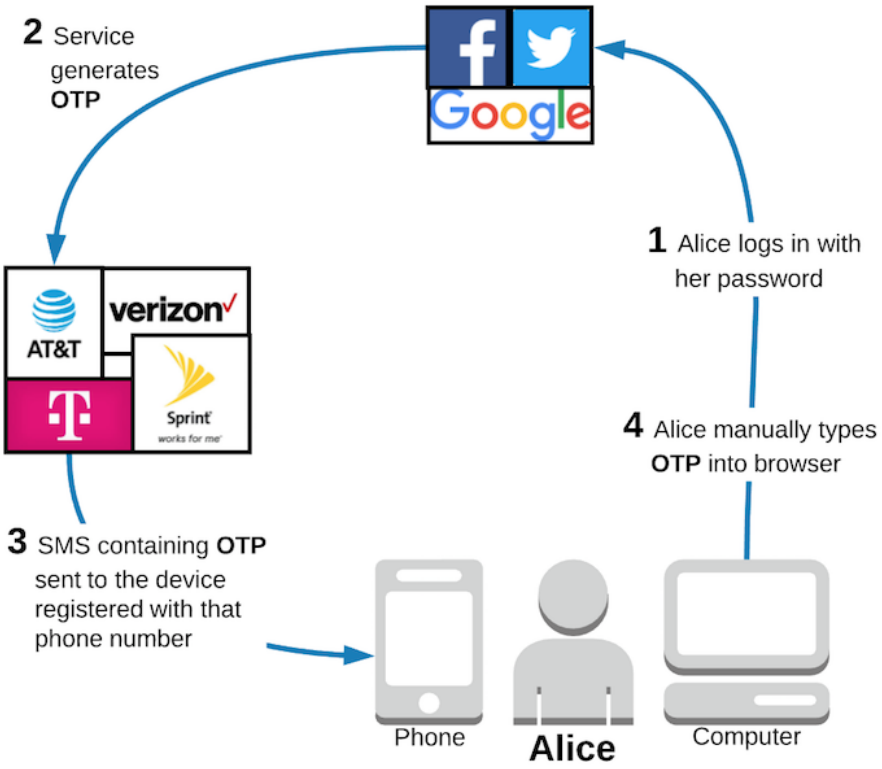


Figure 4.1:

cons

"Just an algorithm"

1. synchronization
2. invalidation
3. nobody knows how the algorithm is implemented (RFC = no standard)
4. Differences (e.g. Steam - only 5 digits, limited Alphabet)
5. Brute Force if server does not limit
6. Not phishing resistant

4.2 Smart Cards

4.3 Hardware Tokens

5 Security of Multi Factor Authentication

5.1 Introduction

In this chapter the introduced MFA solutions are analyzed in regards of their security aspects, ranging from algorithms to transportation risks.

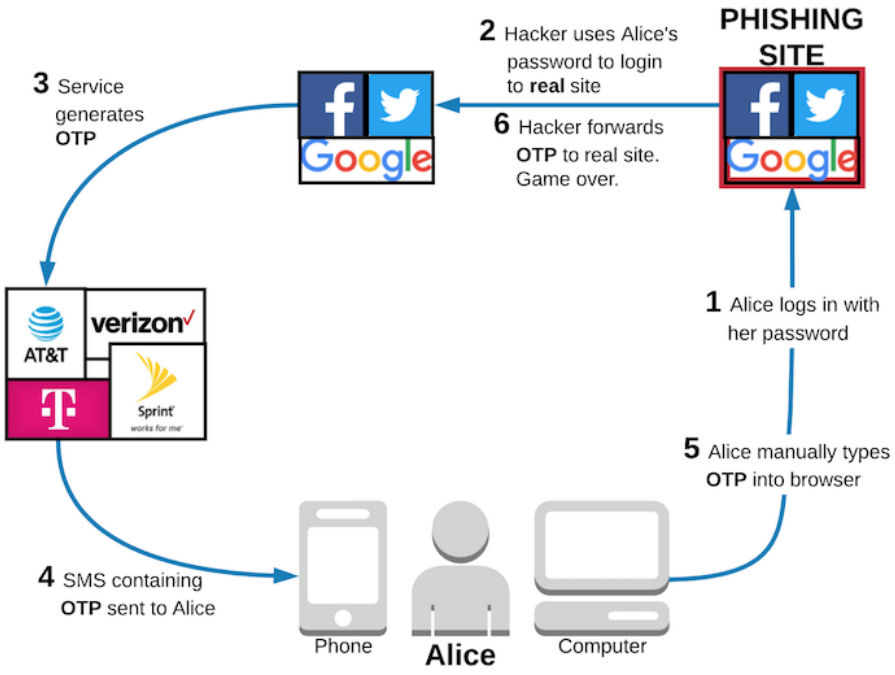


Figure 5.1:

5.2 HOTP and TOTP

In this section the security of both HMAC-based One-time Password algorithm (HTOP) and Time-based One-Time Password algorithm (TOTP) is being analyzed.

5.2.1 Algorithm

As both the HTOP and the TOTP are based on the HMAC algorithm by building the OTP over the HMAC function of the secret key and the counter with a truncation, the underlying HMAC algorithm needs to be evaluated.
The important part here is the chosen cryptographic hash algorithm. Mostly Secure Hash Algorithm (SHA)-1 is used, since it's the default of the RFC. Given that both SHA-1 and MD5 are considered insecure one has to ask if they are still considered secure in the OTP context.
Because the collision resistance of the chosen cryptographic hash algorithm is not important for the security of the OTP generation those algorithms do not expose a threat.
The Federal Office for Information Security (BSI) lists these algorithms as secure for HMAC[21]

Citations: []

It is more important that the algorithm is implemented correctly, in the past e.g. Google did not issue OTP values with a leading zero. Besides that, the minimum length of the OTP values are six digits, meanwhile the RFC supports up to 10.
For example Steam, decided to use a different alphabet and character length.

A theoretical vulnerability is to use the time sync offset feature because it enables an attacker to use a token that's much longer valid than it should be. (as discussed in section xx - time sync/drift)

5.2.2 Transportation

Given that the generation of the OTP is considered secure the more important region to analyze is the transportation of these OTP. In this section the transportation mediums SMS, E-Mail and App are considered.

SMS

The biggest advantage of SMS as a transportation medium is every mobile, ranging from an old Nokia to a new iPhone XS, is capable of receiving SMS. All major mobile phone operation systems come with a SMS application pre-installed, so no

[21] Sic19.

external apps are required.
SMS are around 1999 and highly accepted and easy to use.

While there are some key advantages with SMS transportation it also comes with a lot of downsides. Besides the cost aspect of SMS traffic, both for the sender and potentially for the receiver due to roaming fees, too, the current state of SMS traffic is considered insecure.
The SMS traffic relies on the Signalling System No. 7 (SS7) network which was developed in the 1970s. It has multiple security flaws that allows an attacker to eavesdrop or modify the in- and out-coming traffic.[22]

In contrast to the web and email the user is not very aware of phishing attacks in the SMS context. Studies however show that a new technique called forward phishing is already in use. In this scenario the attacker sends the victim a (spoofed) SMS from the fakes service provider to reply with the OTP code for security measures.[23]

Another negative aspect of SMS transportation is the routing. Many companies rely on third-party providers in order to send the SMS to the user. Often these providers like name some are using countries where SMS are very cheap, but on the other hand the SS7 security measures like SMS home routing and not enforced. This results in a higher security risk of the SMS being compromised while reaching the user. Also, the third party providers are given access to the OTP which enables the risk of a malicious insider because the security measures might be weaker than the original company.

Especially for Android there exists multiple SMS trojans which are capable of intercepting the SMS, too.

Given all these facts SMS transportation should be avoided at all costs[24], since there are multiple flaws in the SS7 network itself and the process how the SMS reaches the user. It's also not resistant against phishing or mobile phone trojans.

cons

1. Delivery time
2. SIM Swapping, cloning, hijacking, ...

[22] Wel17; HO17; Puz17.
[23] Jak18; Sia+17.
[24] Jak18.

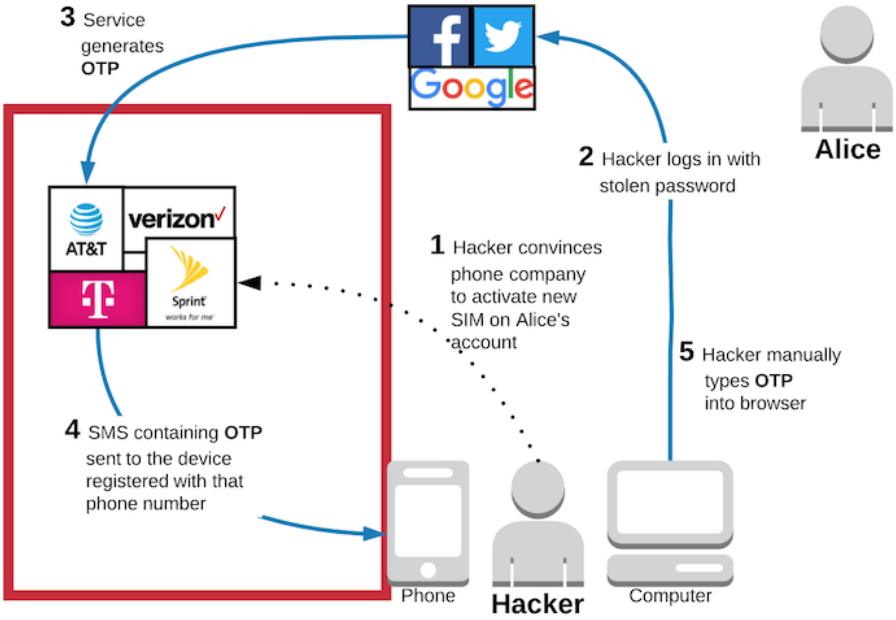


Figure 5.2:

App

pros

1. Works offline
2. cheaper

cons

1. Secret can be phished while setup (either on phone or computer)
2. Trusted apps? OSS?[25]
3. Vulnerabilities –> e.g. Authy[26]

E-Mail

pros

cons

[25] Ste19.
[26] Hom15.

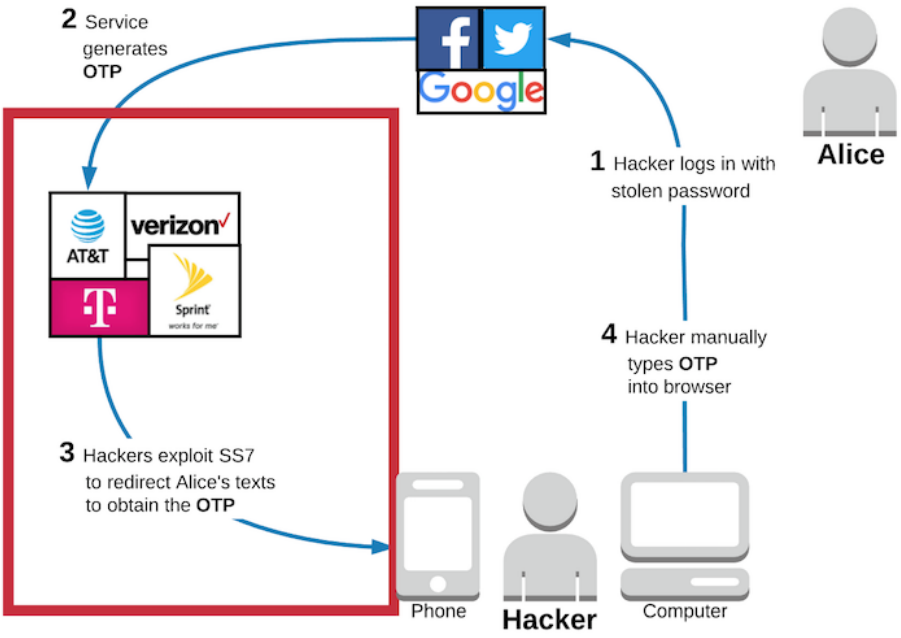

2 Service generates OTP
Google
Alice
1 Hacker logs in with stolen password
AT&T
verizon
Sprint
4 Hacker manually types OTP into browser
3 Hackers exploit SS7 to redirect Alice's texts to obtain the OTP
Phone
Hacker
Computer

Figure 5.3:

6 WebAuth

6.1 History and evolution

Fast IDentity Online (FIDO) World Wide Web Consortium (W3C) Universal Second Factor (U2F) Universal Authentication Framework (UAF) API

6.2 Technical implementation and details

6.2.1 Browser support

Table 1 shows the support status of the Web Authentication API for the most common web browser, both desktop and mobile and if they support the API and if yes since which version and when the release date was. The following subsections will explain the browser support more detailed.

The global browser support as of August 2019 is 68%.

	Web browser	Supported	Version	Release Date
Dekstop	Chrome	✓	67	May 2018
	Firefox	✓	60	May 2018
	Opera	✓	54	June 2018
	Internet Explorer	✗	-	-
	Edge	✓	18	November 2018
	Safari	(✓)	(13)	-
Mobile	Opera Mobile	✗	-	-
	IE Mobile	✗	-	-
	iOS Safari	✗	-	-
	iOS Safari	✗	-	-
Android	LineageOS Stock Browser	✗	-	-
	Chrome for Android	✓	70	October 2018
	Firefox for Android	✓	68	July 2019
	Opera	✗	-	-
	Opera mini	✗	-	-
	Edge	✗	-	-
	Samsung Internet	✗	-	-
	UC Browser	✗	-	-
	Mint Browser	✗	-	-
	360 Secure Browser	✗	-	-
	QQ Browser	✗	-	-
	Brave Browser	✗	-	-

Table 1: Browser support of the Web Authentication API[27]

[27] Sources: [BK18; JT18; Dav18; Ger18; Jon19]. A detailed analysis of Android browsers is available on the CD in the appendix.

Desktop support

The Web Authentication API is supported from Chrome 67 onwards, which was released in May 2018. Firefox added support for the Web Authentication API in May 2018, too, with version 60.
Microsoft added support for the Web Authentication API in Edge 13 which was released in November 2015. However, the implementation is based on an earlier draft version of the Web Authentication API. Support for the FIDO 2.0 specification was added in Edge 14 (released in December 2016). The feature is hidden behind a configuration option though and was enabled for all users with the release of Edge 17 in November 2018.

Browsers like Opera, Vivaldi or Brave and upcoming Edge versions, that are all based on Chromium, the browser and source code behind Google's Chrome browser, have support for the Web Authentication API, too.

As the development for the Internet Explorer halted and it's only receiving security updates, no support is available for new web APIs including the Web Authentication API, even though it's still used by 5% of all desktop browser users and remains supported for the operating system Windows 7, 8.1 and 10.[28] This is an important fact to take into account when evaluating the usability of the Web Authentication API since especially enterprise users often can't upgrade or switch their browser.

Safari added support for the Web Authentication API feature in December 2018 but only for the preview variant of the browser, called the Safari Technology Preview. It is expected to be available for all users with the release of Safari 13 in mid to end September. The support is limited to USB HID enabled authenticator though and only available for macOS Mojave and Catalina and yet unknown if older macOS version will receive an update to Safari 13.

Besides that Windows 10 also added support for MFA by incorporating the technology described in the FIDO standard. This allows biometric authentication with e.g. fingerprints when a reader is available or to use the facial recognition technology or iris scans. This feature is called »Windows Hello«. The credentials are only stored locally and are protected by asymmetric encryption. Besides biometric authentication Windows Hello also supports PINs, those are stored in the Trusted Platform Module (TPM).[29]

[28] See Sup19c.
[29] Bio16.

Mobile support

In contrast to the desktop support, the support for the Web Authentication API is worse for mobile web browsers. While Android supports the Web Authentication API since October 2018 for Chrome and in July 2019 Firefox for Android added support for it, iOS lacks support for the Web Authentication API. Even though in the iOS 13 beta versions the feature can be enabled in the »Experimental Features« section the API remains unsupported or at least there is no way to add an authenticator in the browser yet.
The only ray of hope is that the Brave browser for iOS incorporated support for the yet to be released security key »YubiKey 5Ci« which enables U2F and the Web Authentication API for iOS by using an Apple certified Lightning accessory. Unfortunately due to lack of availability this functionality could not be tested.[30]

It has to be noted though, that other Android browser vendors need to implement the functionality themselves. In other geographic regions browsers like the UC Browser, 360 Security Browser, Mint Browser from Xiaomi or the QQ Browser from Tencent are widely used. Neither them, nor browsers such as Samsung Internet, Opera (mini) for Android, Edge or the Android Stock browser are currently supporting the Web Authentication API.

Other mobile operation systems like Windows Phone 8, BlackBerry 10 or KaiOS do not support the Web Authentication API.

6.2.2 Usability

6.3 Security aspects

Problems: [Sta18]

6.3.1 Problems

- Identify theft if not as 2FA and key is lost (e.g. Yubikey without fingerprint sensor)

[30] See Bra19.

7 Comparison

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special contents, but the length of words should match the language.

8 Conclusion and Outlook

OAuth 2.0, KERERBOS, radius based, LDAP, AD, OpenID Connect, SAML

Bibliography

[And08] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. 2nd ed. Indianapolis, USA: Wiley, 2008. ISBN: 978-0-470-06852-6.

[BB17] Lee Brotherston and Amanda Berlin. *Defensive Security Handbook: Best Practices for Securing Infrastructure*. Sebastopol, CA, USA: O'Reilly Media, 2017. ISBN: 978-1-491-96038-7.

[Bio16] "Microsoft Windows Hello biometric login now works with websites." In: *Biometric Technology Today* 2016.4 (2016), p. 12. ISSN: 0969-4765. DOI: 10.1016/S0969-4765(16)30071-6.

[DLE19] Thomas Dressel, Eik List, and Florian Echtler. "SecuriCast: Zero-touch Two-factor Authentication Using WebBluetooth." In: *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. EICS '19. Valencia, Spain: ACM, 2019, 6:1–6:6. ISBN: 978-1-4503-6745-5. DOI: 10.1145/3319499.3328225.

[DRN17] Dipankar Dasgupta, Arunava Roy, and Abhijit Nag. "Multi-factor authentication." In: *Advances in User Authentication*. Cham, Switzerland: Springer International Publishing, 2017, pp. 185–233. ISBN: 978-3-319-58808-7. DOI: 10.1007/978-3-319-58808-7_5.

[Eck14] Claudia Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. 9. Auflage. Munich, Germany: De Gruyter, 2014. ISBN: 978-3-486-77848-9.

[FKH14] Tobias Fiebig, Jan Krissler, and Ronny Hänsch. "Security Impact of High Resolution Smartphone Cameras." In: *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. San Diego, CA, USA: USENIX Association, Aug. 2014.

[FSS18] Julian Fietkau, Starbug, and Jean-Pierre Seifert. "Swipe Your Fingerprints! How Biometric Authentication Simplifies Payment, Access and Identity Fraud." In: *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD, USA: USENIX Association, Aug. 2018.

[Gri17] R.A. Grimes. *Hacking the Hacker: Learn From the Experts Who Take Down Hackers*. Indianapolis, IN, USA: Wiley, 2017. ISBN: 978-1-119-39621-5.

[HO17] Silke Holtmanns and Ian Oliver. "SMS and One-Time-Password Interception in LTE Networks." In: *2017 IEEE International Conference on Communications (ICC)*. May 2017, pp. 1–6. DOI: 10.1109/ICC.2017.7997246.

[Jak18] Markus Jakobsson. "Two-factor inauthentication – the rise in SMS phishing attacks." In: *Computer Fraud & Security* 2018.6 (2018), pp. 6–8. ISSN: 1361-3723. DOI: 10.1016/S1361-3723(18)30052-6.

[JRN11] Anil K. Jain, Arun A. Ross, and Karthik Nandakumar. *Introduction to Biometrics*. New York, NY, USA: Springer, 2011. ISBN: 978-0-387-77325-4. DOI: 10.1007/978-0-387-77326-1.

[Kei17] Konstantinos Markantonakis Keith Mayes. *Smart Cards, Tokens, Security and Applications*. 2nd ed. Cham, Switzerland: Springer International Publishing, 2017. ISBN: 978-3-319-50498-8.

[KSM14] Gurudatt Kulkarni, Ramesh Sutar, and Sangita Mohite. ""RFID security issues challenges"." In: *2014 International Conference on Electronics and Communication Systems (ICECS)*. Feb. 2014, pp. 1–4. DOI: 10.1109/ECS.2014.6892730.

[Mal+15] Aanchal Malhotra et al. *Attacking the Network Time Protocol*. Oct. 2015.

[Puz17] Sergey Puzankov. "Stealthy SS7 Attacks." In: *Journal of ICT Standardization* 5.1 (2017), pp. 39–52.

[Sho14] Adam Shostack. *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014. ISBN: 978-1-118-81005-7.

[Sia+17] Hossein Siadati et al. "Mind your SMSes: Mitigating social engineering in second factor authentication." In: *Computers & Security* 65 (2017), pp. 14–28. ISSN: 0167-4048. DOI: 10.1016/j.cose.2016.09.009.

[Sic19] Bundesamt für Sicherheit in der Informationstechnik. "Kryptographische Verfahren: Empfehlungen und Schlüssellängen." In: *Technische Richtlinie TR-02102-1, Bundesamt für Sicherheit in der Informationstechnik* 1 (2019).

[Tod07] Dobromir Todorov. *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*. Auerbach Publications, 2007. ISBN: 978-1-4200-5219-0.

[TW75] Rein Turn and W. H. Ware. "Privacy and Security in Computer Systems: The vulnerability of computerized information has prompted measures to protect both the rights of individual subjects and the confidentiality of research data bases." In: *American Scientist* 63.2 (1975), pp. 196–203. ISSN: 0003-0996.

[Wel17] Bill Welch. "Exploiting the weaknesses of SS7." In: *Network Security* 2017.1 (2017), pp. 17–19. ISSN: 1353-4858. DOI: 10.1016/S1353-4858(17)30008-9.

[ZKM12] F. Zhang, A. Kondoro, and S. Muftic. "Location-Based Authentication and Authorization Using Smart Phones." In: *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. June 2012, pp. 1285–1292. DOI: 10.1109/TrustCom.2012.198.

Internet sources

[BK18] Christiaan Brand and Eiji Kitamura. *Enabling Strong Authentication with WebAuthn*. May 29, 2018. URL: https://developers.google.com/web/updates/2018/05/webauthn (visited on 08/15/2019).

[Bra19] Brave. *Adding YubiKey Support to Brave for iOS*. June 24, 2019. URL: https://brave.com/ios-yubikey-support/ (visited on 08/17/2019).

[Col16] Katie Collins. *Facebook buys black market passwords to keep your account safe*. Nov. 9, 2016. URL: https://www.cnet.com/news/facebook-chief-security-officer-alex-stamos-web-summit-lisbon-hackers/ (visited on 08/18/2019).

[Dav18] Jon Davis. *Release Notes for Safari Technology Preview 71*. Dec. 5, 2018. URL: https://webkit.org/blog/8517/release-notes-for-safari-technology-preview-71/ (visited on 08/15/2019).

[Ger18] Chromium Gerrit. *Enable WebAuthN on Android by default (If95f7508) · Gerrit Code Review*. Aug. 16, 2018. URL: https://chromium-review.googlesource.com/c/chromium/src/+/1176736/ (visited on 08/15/2019).

[Goo] Google. *Google 2-Step Verification*. URL: https://www.google.com/landing/2step/ (visited on 08/01/2019).

[Hom15] Egor Homakov. *How "../sms" could bypass Authy 2 Factor Authentication*. Mar. 15, 2015. URL: https://sakurity.com/blog/2015/03/15/authy_bypass.html (visited on 08/09/2019).

[Jon19] J.C. Jones. *Web Authentication in Firefox for Android*. Aug. 5, 2019. URL: https://blog.mozilla.org/security/2019/08/05/web-authentication-in-firefox-for-android/ (visited on 08/15/2019).

[JT18] J.C. Jones and Tim Taubert. *Using Hardware Token-based 2FA with the WebAuthn API*. Jan. 16, 2018. URL: https://hacks.mozilla.org/2018/01/using-hardware-token-based-2fa-with-the-webauthn-api/ (visited on 08/15/2019).

[Kre14] Stefan Krempl. *31C3: CCC-Tüftler hackt Merkels Iris und von der Leyens Fingerabdruck*. Dec. 28, 2014. URL: https://heise.de/-2506929 (visited on 08/09/2019).

[Mic19] Microsoft. *How to use two-step verification with your Microsoft account*. July 25, 2019. URL: https://support.microsoft.com/en-us/help/12408/microsoft-account-how-to-use-two-step-verification (visited on 08/01/2019).

[Pla] PlayStation. *2-Step Verification*. URL: https://www.playstation.com/en-us/account-security/2-step-verification/ (visited on 08/01/2019).

[Sta18] P.I.E. Staff. *Security Concerns Surrounding WebAuthn: Don't Implement ECDAA (Yet)*. Aug. 23, 2018. URL: https://paragonie.com/b/ya9unbDYhvmp2EUy (visited on 08/09/2019).

[Ste19] Lukas Stefanko. *Malware sidesteps Google permissions policy with new 2FA bypass technique*. June 17, 2019. URL: https://www.welivesecurity.com/2019/06/17/malware-google-permissions-2fa-bypass/ (visited on 08/09/2019).

[Sup19a] Apple Support. *Two-factor authentication for Apple ID*. July 30, 2019. URL: https://support.apple.com/en-us/HT204915 (visited on 08/01/2019).

[Sup19b] Apple Support. *Two-step verification for Apple ID*. May 29, 2019. URL: https://support.apple.com/en-us/HT204152 (visited on 08/01/2019).

[Sup19c] Microsoft Support. *Lifecycle FAQ—Internet Explorer and Edge*. June 12, 2019. URL: https://support.microsoft.com/en-us/help/17454/lifecycle-faq-internet-explorer (visited on 08/15/2019).

List of Figures

Listings

List of Tables

Glossary

S

SS7 A telephony signaling protocol.

T

TPM A secure chip designed to provide security functions such a secure random number generator, sealing, protection of cryptographic keys or remote attesstation to an operating system.

W

W3C The international standards organization for the World Wide Web.

Acronyms

Symbols

2FA Two-factor Authentication

A

API Application Programming Interface

B

BLE Bluetooth Low Energy

BSI Federal Office For Information Security

E

ECC Elliptic-curve Cryptography

F

FAR False Acceptance Rate

FIDO Fast IDentity Online

FRR False Rejection Rate

H

HMAC Keyed-Hashing For Message Authentication

HTOP HMAC-based One-time Password Algorithm

I

IT Information Technology

M

MAC Message Authentication Code

MFA Multi-factor Authentication

MITM Man-in-the-Middle

N

NFC Near-field Communication

NIST National Institute Of Standards And Technology

NTP Network Time Protocol

O

OATH Initiative For Open Authentication

OTP One-time Password

P

PIN Personal Identification Number

R

RFC Request For Comments

RFID Radio-frequency Identification

S

SHA Secure Hash Algorithm

SS7 Signalling System No. 7, *Glossary:* SS7

SSO Single Sign-on

T

TOTP Time-based One-Time Password Algorithm

TPM Trusted Platform Module, *Glossary:* TPM

U

U2F Universal Second Factor

UAF Universal Authentication Framework

W

W3C World Wide Web Consortium, *Glossary:* W3C

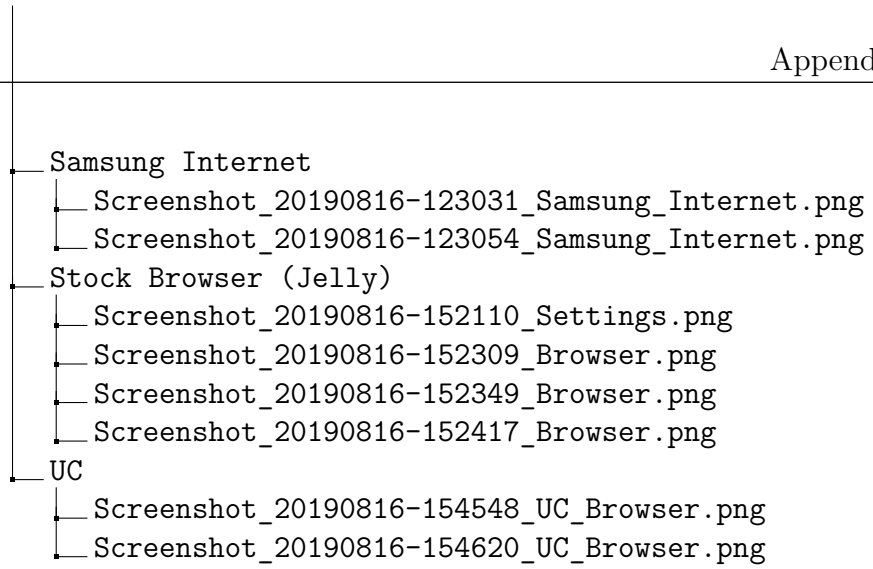
A Appendix

A.1 Test

B Annex

B.1 Table of Content of the CD-Rom

```
/Volumes/CWniwQ7mThQP_0/Masterthesis/LaTeX/cd_rom
└── Web Authentication API Support Test Android
    ├── 360
    │   ├── Screenshot_20190816-154301_360.png
    │   ├── Screenshot_20190816-154306_360.png
    │   └── Screenshot_20190816-154438_Settings.png
    ├── Brave
    │   ├── Screenshot_20190816-153730_Brave.png
    │   ├── Screenshot_20190816-153901_Brave.png
    │   ├── Screenshot_20190816-153910_Brave.png
    │   └── Screenshot_20190816-153936_Brave.png
    ├── Chrome
    │   ├── Screenshot_20190816-121238_Google_Play_services.png
    │   ├── Screenshot_20190816-121245_Google_Play_services.png
    │   └── Screenshot_20190816-121304_Chrome.png
    ├── Edge
    │   ├── Screenshot_20190816-135319_Edge.png
    │   ├── Screenshot_20190816-135334_Edge.png
    │   └── Screenshot_20190816-135347_Edge.png
    ├── Firefox
    │   ├── Screenshot_20190816-121429_Firefox.png
    │   └── Screenshot_20190816-122820_Firefox.png
    ├── Mint
    ├── Opera
    │   ├── Screenshot_20190816-155037_Opera.png
    │   ├── Screenshot_20190816-155044_Opera.png
    │   └── Screenshot_20190816-155108_Opera.png
    ├── Opera mini
    │   ├── Screenshot_20190816-155738_Opera_Mini.png
    │   ├── Screenshot_20190816-155750_Opera_Mini.png
    │   └── Screenshot_20190816-155815_Opera_Mini.png
    └── QQ
        ├── Screenshot_20190816-121712_QQ.png
        ├── Screenshot_20190816-121734_QQ.png
        └── Screenshot_20190816-122522_Settings.png
```


Samsung Internet
Screenshot_20190816-123031_Samsung_Internet.png
Screenshot_20190816-123054_Samsung_Internet.png
Stock Browser (Jelly)
Screenshot_20190816-152110_Settings.png
Screenshot_20190816-152309_Browser.png
Screenshot_20190816-152349_Browser.png
Screenshot_20190816-152417_Browser.png
UC
Screenshot_20190816-154548_UC_Browser.png
Screenshot_20190816-154620_UC_Browser.png

Declaration of Academic Integrity

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

Hamburg, August 19, 2019

Tim Brust

Theses

1. The status quo of password usage is bad, often chosen passwords are re-used and weak.

2. Humans are the weakest link.

3. 2FA is not phishing resistent, both the secret when setting it up and the second factor can be phished or stolen. Software solutions are more probable to be phished.

4. The biggest threat to 2FA is the transportation especially when using SMS or unencrypted e-mail traffic.

5. MFA can made be phishing resistant but it requires more effort to do so.

6. Web Authentication API is not yet usable enough nor widely adopted, this is especially true because iOS lacks support for it and the Internet Explorer is still widely used.

7. The user needs to be educated about passwords, the risk of password re-use, phishing and how to protect themselves against common (internet) threats.