Fachhochschule Wedel
Faculty of Computer Science

Authentication and Authorization

Submitted By:

Marc Mettke

IT-Security
FH Wedel
103829

Under the supervision of:

Prof. Dr. Gerd Beuster
FH Wedel

Hamburg, 2019-05-19

Contents

List of Figures

1 Introduction

Authentication and authorisation is the foundation for securing digital information. Their use is to verify that an entity is whom it claims to be and to limit its read and write access to resources required for its operation. It can be found everywhere in the digital world starting at the LAN level using Methods like IEEE 802.1X [NS] up to the server and service levels using Methods like SSH [SS1] and TLS [SS2].

Especially on the Internet, the authentication method of choice is the Password combined with a username which is known publicly most of the time. Its nature of being cheap and easy to develop makes it the number one choice even thou users have difficulties coming up with good and different passwords for each service.

With a continuing increase in services offering personalization and thus requiring authorization, users tend to use the same password more and more often which brings the danger of losing control of multiple accounts when passwords are leaked from a single service [PR]. This issue leads to research and development of new methods like Two-factor authentication to make the process of authentication more natural and convenient without introducing high development cost or removing ease of use.

Purpose and Scope of Seminar

This paper is about exploring the consequence and dangers of the frequent usage of passwords and taking a critical look at the recently developed methods of authentication. Its main focus is to question whether they actually solve the problems we have with password authentication.

2 Current Danger

The frequent usage of passwords around the internet is leading to more and more password reuse or centralized storage of passwords in Password Managers, which, in turn, leads to single-points of failure. This section will be about taking a look at the current dangers present in the usage of passwords.

2.1 Password Cracking

Storing passwords is a major concern in web services. While there are still services storing passwords in plain text or with rather basic methods like obfuscation or simple hashing, most of them at least try to keep up the arms raise of storing passwords securely.

But even with the development of new adaptive one-way functions for storing passwords [OPSCS] the problem of storing and verifying passwords is still nontrivial. These new functions require a server to invest more CPU power and memory to create a secure password in a process which is slow by design to counter fast cracking rigs. In contrast to the cracking rigs which benefit from a constant increase of performance in GPU processing power, servers are mostly staying at their level for years. Especially cheap or free services are not able to afford servers powerful enough to use reasonable settings for those new functions [TRB].

Figure 2.1: IBM Cracking Rig [PC]

Figure 2.1 contains a Cracking Rig from IBM made to perform password cracking tests within a few days only. With the use of multiple GPU Units, they are able to crack over 100 thousand passwords hashed using bcrypt, one of the newer password hashing functions [PC].

The performance increase of cracking rigs is, however, not the only problem. With a lot of users reusing passwords an attack on a lower cost service might prove similar worthy to an attack to a big target, but a lot easier to perform when it comes to cracking passwords. One password is therefore only as secure as the weakest storage function of the services it is used in [PRD].

2.2 Phishing

While password cracking is already a very sophisticated attack yielding great results especially when done right with some time at hand, phishing and especially spear phishing is an even more dangerous attack allowing the attacker to target a specific group or person.

Instead of trying to guess a person's password, phishing is about making the user to reveal his password voluntarily. There are numerous ways to trick a person like impersonating Service-Personal in a company, create a fake website for snooping credentials or planing malware into the victim's computer.

In June 2015 the Ubiquiti Networks Inc. lost 46.7 million Dollar due to a spear phishing attack target only to the finance department of the company [SPE]. While there was no compromise of the network itself, employees were tricked to transfer the money using executive mail addresses and carefully crafted alternative domains.

Unfortunately, attacks like these are not scarce but happen regularly. Especially as the protocol for mail delivery was never made to prevent the spoofing of mail addresses with little countermeasures available to date.

2.3 What it means

Losing one's password or more precisely having another person being able to access ones services in best cases lead to financial losses and in worst cases to one getting into legal problems with the government. One example of this is a woman whose identity was used in the case of murder [IT]. The main reason why she was not convicted was the fact that the real murderer used a fake id with her information but his photograph.

This is by far not the only example of identify theft. Other examples include the dangerous field of social media. Nowadays a post shared on social media can have the same effect as a word said in an interview leading to bans from online service and real-life events [AASF][OB].

3 Alternative Methods

Due to the mentioned problems about password authentication, there is ongoing research for new authentication methods to replace or improve password authentication. The following list contains an overview of currently available alternatives. Whether they are suitable for the job at hand is based on whether they offer unique authentication for each service, are reasonable to implement on server and client side and whether a common user would be able to use them.

3.1 What is authentication

Authentication is the process of verifying whether an entity is whom it claims to be. To do that, the authentication process itself uses one or more of the following authentication factors [AD].

Knowledge factor The "Knowledge factor" also known as "Something you know" uses information that only the user is supposed to know like passwords or pin codes. Most of the web services are using it due to its cheap implementation.

Possession factor The "Possession factor" stands for "Something you have" and referees to a device which provides information like a one-time password. Examples for those devices are Smart Cards and Hardware or Software Token Generators.

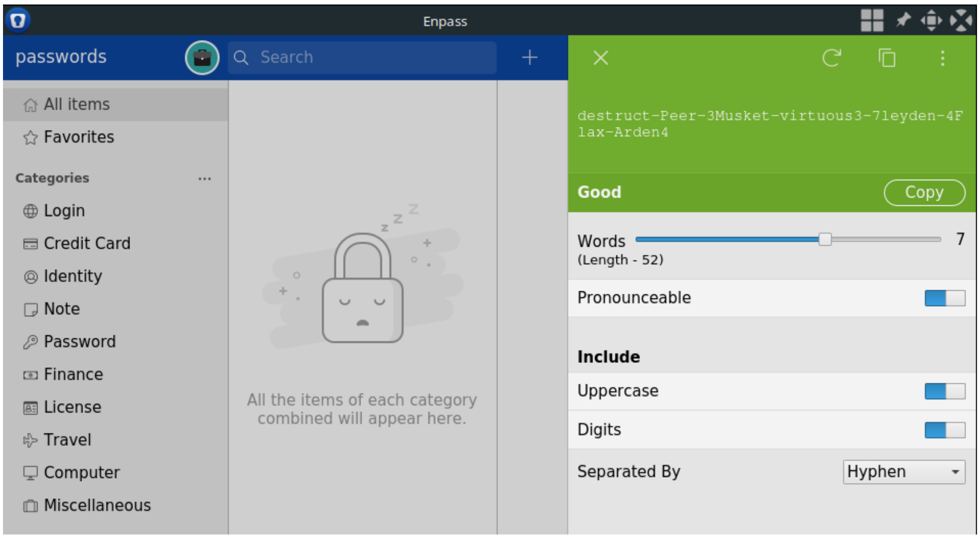
Inherence factor The "Inherence factor" is known as the "Something you are" and is one of the more dangerous ways to authenticate. It is incorporating physical body parts like fingerprints or the iris of an eye. While it is possible to change things we know and have, it is very hard to change our body. There is a follow up on this topic in the Biometric Section later on. (Biometrics on Page 8)

Location factor The "Location factor" is based on "Where you are" and cannot be used for sufficient authentication on its own. It is, however, a supplement and questions the likelihood whether a user would be able to be at the given location given the time and location of its last login.

Time factor The "Time factor" also known as "When you are authenticating" is also not sufficient as an authentication on its own. But it can be used for example to limit the authentication to a given time at a day like the usual working hours.

3.2 Password Manager

The first solution is using a password manager which allows storing the information for the “Knowledge factor”. They have advanced quite a lot in the last years and are either available as online (e.g. LastPass [LP]) or offline service (e.g. enpass) for all major operating systems including the mobile market. They also perform very well when it comes to the questions at hand:



All the items of each category combined will appear here.

```
destruct-Peer-3Musket-virtuous3-7leyden-4F
lax-Arden4
```

Good

Words 7
(Length - 52)
Pronounceable

Include
Uppercase
Digits
Separated By Hyphen

Figure 3.1: Enpass Password Manager

- **Authentication for each service**
  Password managers store one password for each service. While it is up to the user to use a different one for each service, a Password Manager makes it rather easy to do so as they come with a password generation tool.

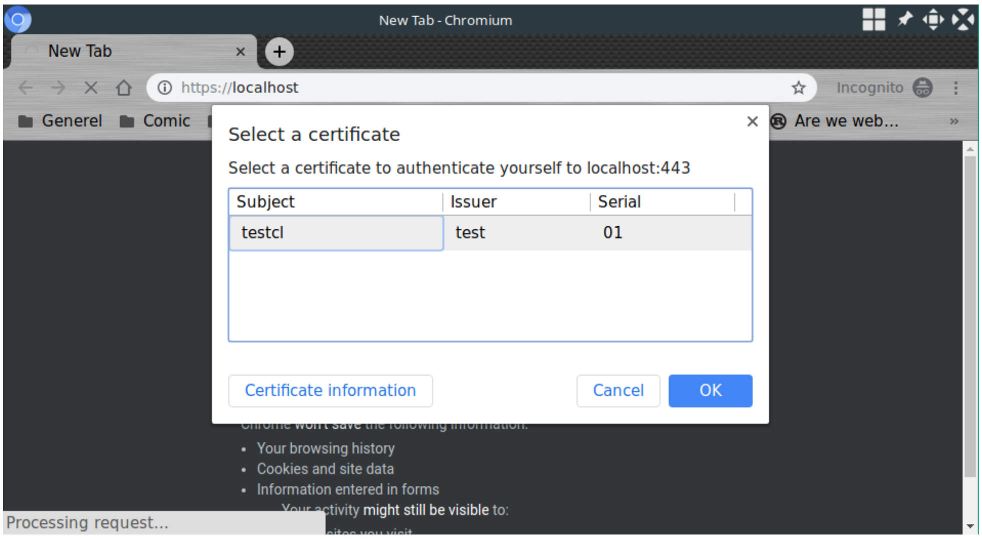
- **Ease of implementation**
  A Password Manager does not require any implementation afford on the server side.

- **Ease of use**
  Good Password Managers offer auto fill-in for password forms and thus make logging in easier for most services. There are, however, services which don’t work well with those fill-ins requiring the user to manually enter them. Especially on the mobile phone, this can become quite tedious.

Password Managers are therefore already quite good when it comes to providing more security in dealing with passwords. However, there is a disadvantage of using password managers. The fact that all passwords are stored in one place, on one hand, makes using them really easy and on the other hand, introduces a single point of failure. When an attacker is able to get the password file, they are only one password away from accessing every service a user is signed up to. This issue combined with the fact that it is hard if not impossible to verify whether the encryption of passwords and especially the transfer of the master password is secure presents a risk that must be considered before using them.

3.3 Client Certificate

Another solution is about using client certificates, which belong to the “Possession factor” of authentication:



Select a certificate

Select a certificate to authenticate yourself to localhost:443

Subject	Issuer	Serial
testcl	test	01

Chrome won't save the following information:

- Your browsing history
- Cookies and site data
- Information entered in forms

Your activity might still be visible to:

sites you visit

Figure 3.2: Client Certificate Authentication

- **Authentication for each service**
  Certificates cannot be reused for multiple services except if they are from the same company. This is due to the fact, that they require to be signed by a ca which must be in the possession of the company itself.

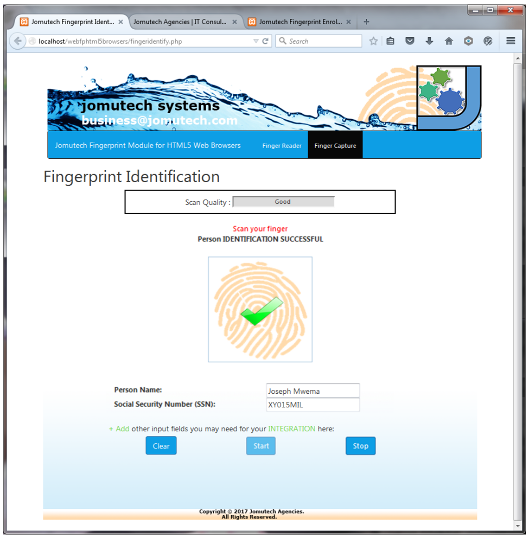
- **Ease of implementation**
  Certificate verification can be implemented on the application server only or on the application- and the web server. While the second is quite easy, the first depends on the language and frameworks used while developing. The first, however, would also allow using certificates as an alternative authentication method.

- **Ease of use**
  It is possible to install client certificates on all operating systems including the mobile market. Using them is only a matter of browsing the given site and selecting the certificate to use.

Client Certificates solve the problem of reuse quite sufficient as it is impossible by design to reuse them. However, there are only two ways to store them on all devices required to authenticate. The first is plain, which requires greater security at the device level, as an attacker would be able to log into every service as soon as a device is either left unlocked or is unprotected. The second way is using a password to encrypt the certificate. This, however, introduces the problem of password reuse once again only removing the problem of having passwords stored on a third party server.

3.4 Biometrics

Biometrics is a way to use the "Inherence factor" for authentication. It includes physiological patterns like fingerprints and behavioral patterns like the way of writing. Those patterns are as good as unique for each person allowing an authentication without having the user to know or have anything but themselves.


Jomutech Fingerprint Ident...
Jomutech Agencies | IT Consul...
Jomutech Fingerprint Enrol...
localhost/webfphtml5browsers/fingeridentify.php
Search
jomutech systems
business@jomutech.com
Jomutech Fingerprint Module for HTML5 Web Browsers
Finger Reader
Finger Capture
Clear
Start
Stop

Fingerprint Identification

Scan Quality : Good

Scan your finger

Person IDENTIFICATION SUCCESSFUL

Person Name: Joseph Mwema

Social Security Number (SSN): XY015MIL

+ Add other input fields you may need for your INTEGRATION here:

Copyright © 2017 Jomutech Agencies.
All Rights Reserved.

Figure 3.3: Jomutech offering biometric authentication [BA]

- **Authentication for each service**
  Authentication variety is very limited. While there are different patterns to use, the pattern itself always stays the same.

- **Ease of implementation**
  Biometrics belong to the hardest authentication when it comes to implementation. It requires a secure transfer, analyses, and storage of the biometric data coming from the users' devices. Services like these are not openly available and it might be necessary to implement them completely.

- **Ease of use**
  Biometric also belongs to the hardest to implemented on the users' side. It requires

advanced scanners which not only are quite expensive but also hard to carry with, especially in good quality.

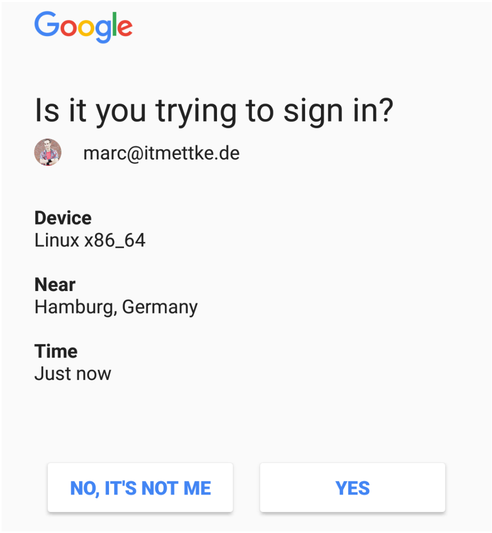
In recent years, more mobile phones were equipped with fingerprints scanners to allow easier and faster authentication for users. While those scanners are fast to use and offer decent security when it comes to the everyday thief, their lack of quality allows advanced attackers to spoof fingerprints rendering the security useless [FDH][FH].

However fingerprint hacking is not the only problem when it comes to biometrics. Especially for security against governments or criminal organizations, biometrics will not be sufficient as one can be forced to unlock a device/service using the body part in question, even thou there is still debate whether this should be allowed or not [CFF].

Another risk worth considering is the fact that a biometric pattern only exists once on a given body. While it is possible to use different patterns like different fingers, once one of them is leaked it is gone for good and cannot be changed like a password. Furthermore, as it is a piece of personal information, it also allows to connecting a given username of a service to a real person.

3.5 Peripheral device recognition

Another form of "Possession factor" authentication is the peripheral device recognition. After signing up, a second device like a mobile phone may be registered to confirm new logins.



Is it you trying to sign in?

marc@itmettke.de

Device
Linux x86_64

Near
Hamburg, Germany

Time
Just now

Figure 3.4: Google's peripheral device recognition

- **Authentication for each service**
  While it would be possible to use different devices for different services, it is rather unlikely that users carry multiple network-connected devices with them.

- **Ease of implementation**
  Implementation is rather difficult as there are multiple issues to consider. The implementation must make sure, that only a particular device is able to authorize a login and an attacker is not able to transfer the permission to its own device. The device also must be available at all times, as a login request can occur at any time. Further questions are what happens when a device is not available or even lost.

- **Ease of use**
  How easy it is to use peripheral device recognition depends on the implementation. Most of the time it is enough to authenticate oneself on the device in question to then use it to authorize other devices.

While peripheral device recognition is very easy and convenient to use, the implementation is very difficult and might open new attack vectors resulting in the second authentication being completely useless. The best case is that the user has its phone and is able to prevent another person from signing in by rejecting the request.

A skilled attacker, however, would know about the security method when attacking, making sure that the phone is either lost or deactivated before trying to authenticate. In this case, a service must either offer other authentication methods or fallback to passwords only giving the attacker the chance to login without requiring the user to allow the authentication. Thus a fallback is necessary, as a user must be able to recover the account when the phone is lost.

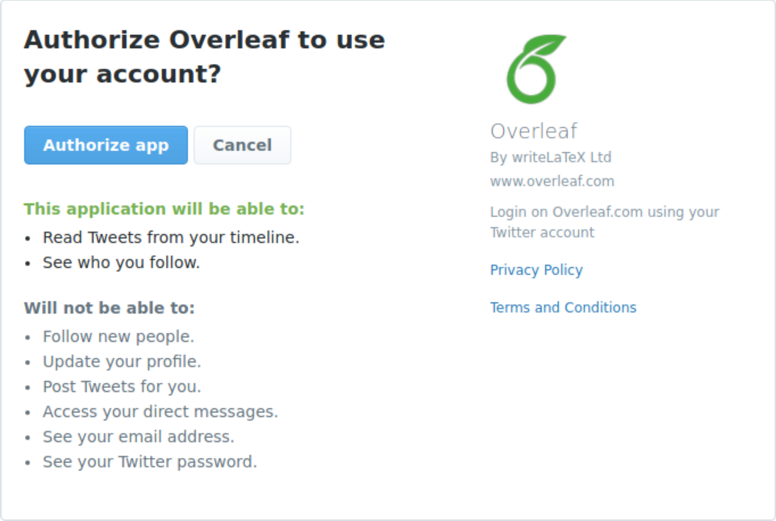
3.6 Single-Sign-On

Single-Sign-On is a form of the “Possession factor” authentication allowing one to access services by proving access to another service. This works by associating two services at the first login. Normally supporting services provide special login buttons, which redirect to the service providing the login information. After logging in, certain information is transferred to the service requiring authentication (like email and name). Both services must support this form of authentication.

- **Authentication for each service**
  The focus of Single-Sign-On is not to provide different authentication for different services, but to centralize them.

- **Ease of implementation**
  There are various libraries for various languages allowing authentication against other services or centralized company-owned services.

- **Ease of use**
  Single-Sign-On is very convenient for the user. After the first connect of two services, logging in becomes as simple as pressing a button. The more services use the same service for authentication the easier it becomes as credentials are saved and are used for each authentication request.



Authorize Overleaf to use your account?

This application will be able to:

- Read Tweets from your timeline.
- See who you follow.

Will not be able to:

- Follow new people.
- Update your profile.
- Post Tweets for you.
- Access your direct messages.
- See your email address.
- See your Twitter password.

Overleaf
By writeLaTeX Ltd
www.overleaf.com

Login on Overleaf.com using your Twitter account

Privacy Policy

Terms and Conditions

Figure 3.5: Single-Sign-On with Twitter

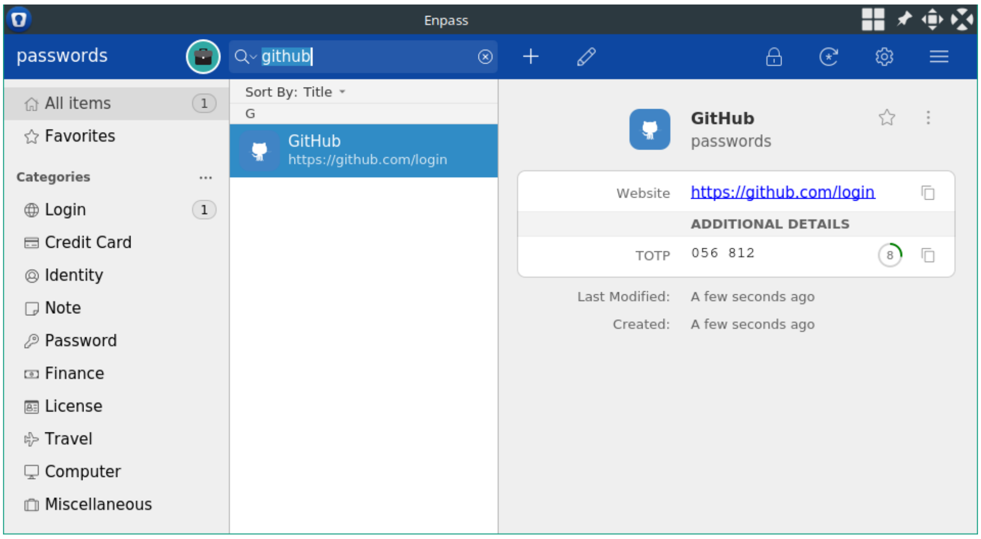
It is very hard to quantify how dangerous it is to use Single-Sign-On. On one hand, all the services are linked to each other making it very easy for an attacker to take over several systems as soon as the account used for authentication is compromised. On the other hand, centralizing authentication makes it very easy for users to secure their account. It is not necessary to know multiple passwords or remember to use different Forms of Authentication for each service. Instead, it is only required to make gaining access to one account as hard as possible to secure all the others.

The second argument is also backed up by the fact that we already have a strong centralization of authorization. Nearly every service offering user authentication provides the functionality to recover the password using the mail address linked to the account. When an attacker is able to compromise the mail account, every service connected to that mail account is compromised as well. Single-Sign-On, therefore, caught be considered an addition to that recovery functionality taking away the need for a service to store passwords and with that improve the overall security of that password by not being stored on different services.

Nevertheless, the decision of using Single-Sign-On should not be taken lightly. Especially as it might introduce a second service as a centralized authority. An Example for this is using a Mail Account for recovery and Facebook Connect as an authorization provider. While the centralization doesn't change as long as the mail account is used for authorization, it does change when a second service like Facebook Connect is used which wouldn't be able to provide access to a service otherwise.

3.7 One-time password

The last authentication method is a combination of the "Knowledge factor" and the "Possession factor" authentication allowing users to login using a temporary password. That temporary password is generated by using a pre-shared key and either the current time, a chain of passwords where every password must be used in order or using a nonce from the server.



Website https://github.com/login

TOTP 056 812

Last Modified: A few seconds ago

Created: A few seconds ago

Figure 3.6: One time password using Enpass

- **Authentication for each service**
  OTP uses a pre-shared key which must be known by the server and the client. Most of the time, the key is generated by the server which makes it impossible to reuse it for different services.

- **Ease of implementation**
  The generation of an OTP is based on very simple algorithms with libraries for various languages.

- **Ease of use**
  The usage requires a special application either for the desktop or mobile environment. Adding a key is just a matter of adding the pre-shared key which very often simplified by using a qrcode.

One-time passwords are a very good alternative to passwords as they take away the need to come up and transfer passwords between client and server for each request. Due to the nature of the pre-shared key being generated on the server, it is not possible for the user to reuse a given key, thus reducing the risk of the key being stored on the server-side.

There are, however, disadvantages to the system if used incorrectly. Using the password chain an attacker might be able to get a visual of the current OTP allowing authentication for a short amount of time if the password is already known. While the attack would become obvious as soon as the user tries to login it would already be too late with the attacker having access to the service. Something similar is possible when the time-based password generation is used. As it requires the current time, an attacker might try to get a future code by changing the device time to a future date. It would then be possible to login to the service with that code and the right time, given that the user password is already known. While these are rather theoretical attack vectors, it is necessary to secure the devices generating OTP.

Another great disadvantage of using OTP is the lack of good OTP Managers. While there are a lot of applications for various operating systems, only very few allow the synchronization of the pre-shared key making it very hard to use OTP on multiple systems.

4 Conclusion

Passwords are here to stay. While in theory, it would be possible for One-time passwords to replace passwords, the risk of losing access to services due to attackers getting a visual or tampering with the time-generation is too big to ignore.

Instead, it is more likely that there will be a bigger shift to Single-Sign-On securing multiple accounts by making sure that it is as hard as possible to get into a special one. A very good example of this is the google account which a lot of services already use for authentication. As a lot of people use google mail for password recovery, the google mail account already is a very important service which must be secured as much as possible.

To secure the service, Google is offering a password, peripheral device recognition using a mobile phone or a security key, one-time passwords, backup codes, and SMS verification. These methods make it easy for a user to regain access when one method is lost without making it easy for an attacker to gain access to a given account. In addition, the user gets notified by mail and on every connected mobile phone when a new login occurs to allow a fast response should an attacker be able to gain access to the account.

With security placements like these, it is possible that in the future, users will be able to login to multiple services around the internet without having to know or enter passwords each time and without compromising their security by doing so. But till then, it is necessary to incorporate Multi-Factor Authentication wherever possible and to encourage users to use password managers with different passwords for each service and to separate login requirements by using different devices.

Bibliography

[FDH] Iphone/Samsung Fingerprint Duplicate Hack - R. Brandom [Zugriff 2019-05-03]
https://www.theverge.com/2016/5/2/11540962/iphone-samsung-fingerprint-duplicate-hack-security

[SPE] Spear Phishing real life examples - D. Brecht [Zugriff 2019-04-30]
https://resources.infosecinstitute.com/spear-phishing-real-life-examples/

[PC] The 'Cracken' in Action: A Password Cracking Adventure - D. Bryan [Zugriff 2019-04-14]
https://securityintelligence.com/the-cracken-in-action-a-password-cracking-adventure/

[NS] RFC3580 IEEE 802.1X - P. Congdon [Zugriff 2019-04-13]
https://tools.ietf.org/html/rfc3580

[PR] The Tangled Web of Password Reuse - A. Das [Zugriff 2019-04-13]
https://www.cs.ucy.ac.cy/courses/EPL682/papers/passwords-2.pdf

[AASF] How Apple and Amazon Security Flaws Led to My Epic Hacking - M. Honan [Zugriff 2019-04-30]
https://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/

[PRD] The Domino Effect of Password Reuse - B. Ives [Zugriff 2019-04-30]
http://130.18.86.27/faculty/warkentin/SecurityPapers/Merrill/IvesWalshSchneider2004_CACM44_4_Domino%20Effect%20of%20Password%20Reuse.pdf

[BA] Jomutech Biometric Authentication - jomutech.com[Zugriff 2019-05-01]
https://jomutech.com/html5webbiometricsauthentication/

[IT] Real Identity Theft Stories - B. Keylore [Zugriff 2019-04-30]
https://www.identityforce.com/blog/4-scary-real-identity-theft-stories

[LP] Lastpass.com [Zugriff 2019-09-09]
https://www.lastpass.com

[SS2] RFC8446 TLS - E. Rescorla [Zugriff 2019-04-13]
https://tools.ietf.org/html/rfc8446

[AD] Authentication - M. Rouse [Zugriff 2019-05-01]
https://searchsecurity.techtarget.com/definition/authentication

[OB] Swiss soccer player banned from Olympics for racist tweet - J. Saraceno [Zugriff 2019-04-30]
http://usatoday30.usatoday.com/sports/olympics/london/soccer/story/2012-07-30/swiss-athlete-banned-michel-morganella-olympics/56591966/1

[OPSCS] OWASP Password Storage Cheat Sheat - J. Steven [Zugriff 2019-04-30]
https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/Password_Storage_Cheat_Sheet.md

[TRB] The Red Book - Syssec [Zugriff 2019-09-09]
http://www.red-book.eu/m/documents/syssec_red_book.pdf

[FH] Hack Fingerprint Scanner Using GLUE/ - sw4p [Zugriff 2019-05-03]
https://www.instructables.com/id/Hack-Fingerprint-Scanner-Using-GLUE/

[SS1] RFC4253 SSH Transport Layer Protocol - T. Ylonen [Zugriff 2019-04-13]
https://tools.ietf.org/html/rfc4253

[CFF] Judge: Police Can't Force You to Unlock Phone With Fingerprint or Face ID - R. Whitwam [Zugriff 2019-05-03]
https://www.extremetech.com/mobile/283795-judge-police-cant-force-you-to-unlock-phone-with-fingerprint-or-face-id