3. Authentication

Computer Security Courses @ POLIMI

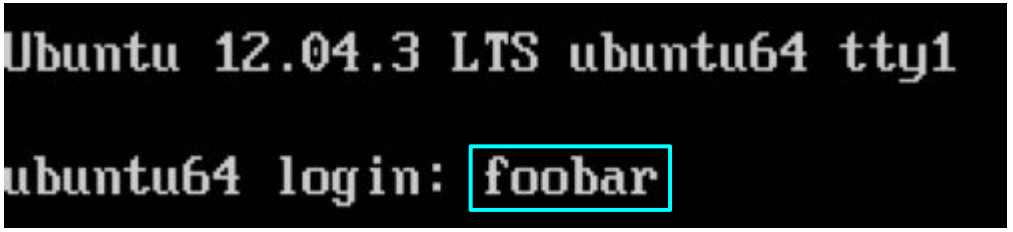
GOAL: VERIFY USER IDENTITY AND ITS AUTHORIZATIONS

Identification *vs.* Authentication

Identification: an entity declares its identifier

- **Examples**: *"I am Stefano"*, *"I am Michele"*

(USERNAMES)

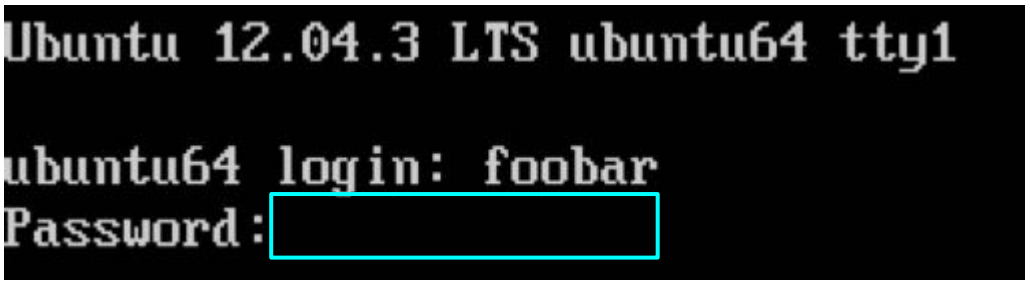


Authentication: the *entity* provides a *proof* that verifies its identity.

PROCESS OF VERIFYING THE IDENTITY CLAIMED BY SOME SYSTEM ENTITY

- **Examples**: *"Here is Stefano's ID card"*

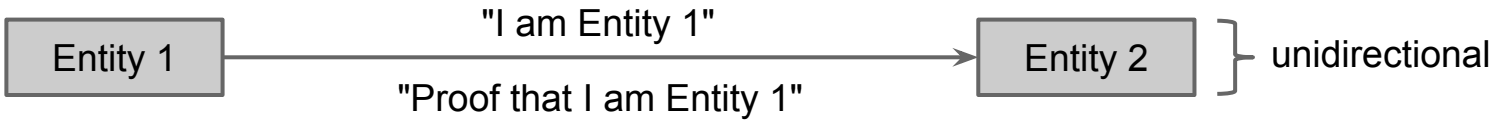
(PASSWORD)



WE MIGHT SAY THAT AUTHENTICATION COMES AFTER IDENTIFICATION

Authentication

Can be *unidirectional*



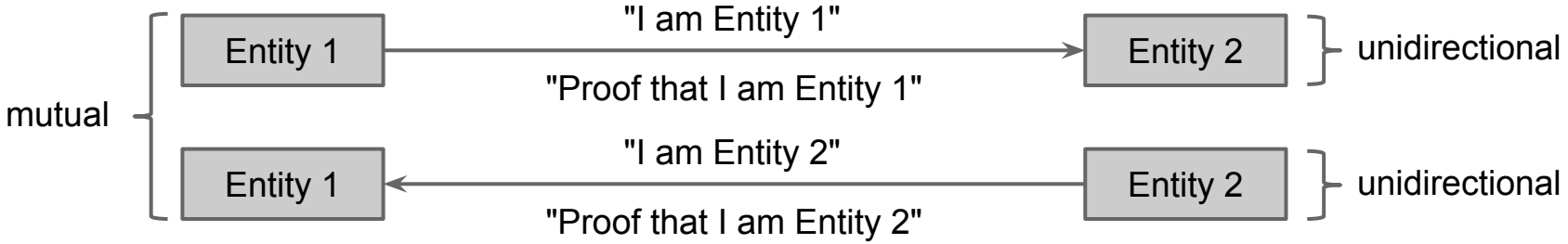
IN GENERAL, THEY ARE UNIDIRECTIONAL AUTHENTICATIONS. IN FACT, ONLY ENTITY 1 VERIFY HIS IDENTITY TO ENTITY 2

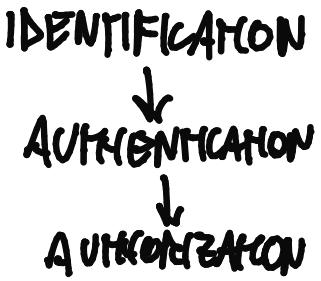
VULNERABLE TO SNOOPING ATTACKS

A SOLUTION IS TO USE BIDIRECTIONAL AUTHENTICATIONS.

Authentication

Can be *unidirectional* or *bidirectional (mutual)*.





Can happen between any entity:

- Human to human
- Human to computer
- Computer to computer

Foundation for the subsequent *authorization* phase

Three Factors of Authentication

Something that the entity ***knows*** (to know)

1. **Example:** password, PIN, secret handshake.

PROBLEM: ANYONE WHO KNOWS THOSE INFORMATIONS CAN AUTHENTICATE HIMSELF AS YOU

Something that the entity ***has*** (to have)

2. **Example:** Door key, smart card, token.

Something that the entity ***is*** (to be)

3. **Example:** Face, voice, fingerprints.

STRONGER AUTHENTICATION REDUCED PROBABILITY OF "UNAUTHORIZED AUTHENTICATION"

Humans: (3) more used than (2), more used than (1).

Machines: (1) more used than (2), more used than (3).

Usability

Multi-factor authentication uses two or three factors.

The "to know" Factor

Passwords and PINs

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)


LA 7 HD

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)

Countermeasures *(i.e., costs)*

Enforce passwords that

- *change/expire* frequently
- are *long* and have a *rich character set*
- are *not related to the user*

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords

Can't pick *unlimited* countermeasures

  - how to choose?

The *"to know"* Factor: Passwords

Advantages

- Low cost,
- ease of deployment,
- low technical barrier.

Disadvantages

Secrets can be

a. stolen/snooped
b. guessed
c. cracked (enumerated)

Countermeasures *(i.e., costs)*

Enforce passwords that

- *change/expire* frequently
- are *long* and have a *rich character set*
- are *not related to the user*

Estimate the most likely attack in the scenario. ⟺ Accordingly choose the countermeasure(s) that are worth asking users to adhere to (remember indirect costs?)

Why are Countermeasures Costs?

Humans are not machines

- Inherently *unable to keep secrets*
- Hard to *remember* complex passwords

Can't pick *unlimited* countermeasures

- how to choose?

Countermeasure guideline: important, may help, unimportant

Against **snooping**	Against **cracking**	Against **guessing**
complexity	*complexity*	*complexity*
change	*change*	*change*
being related to users	being related to users	*not being related to users*

User Education and Password Complexity

User education: "human" == "weak link".

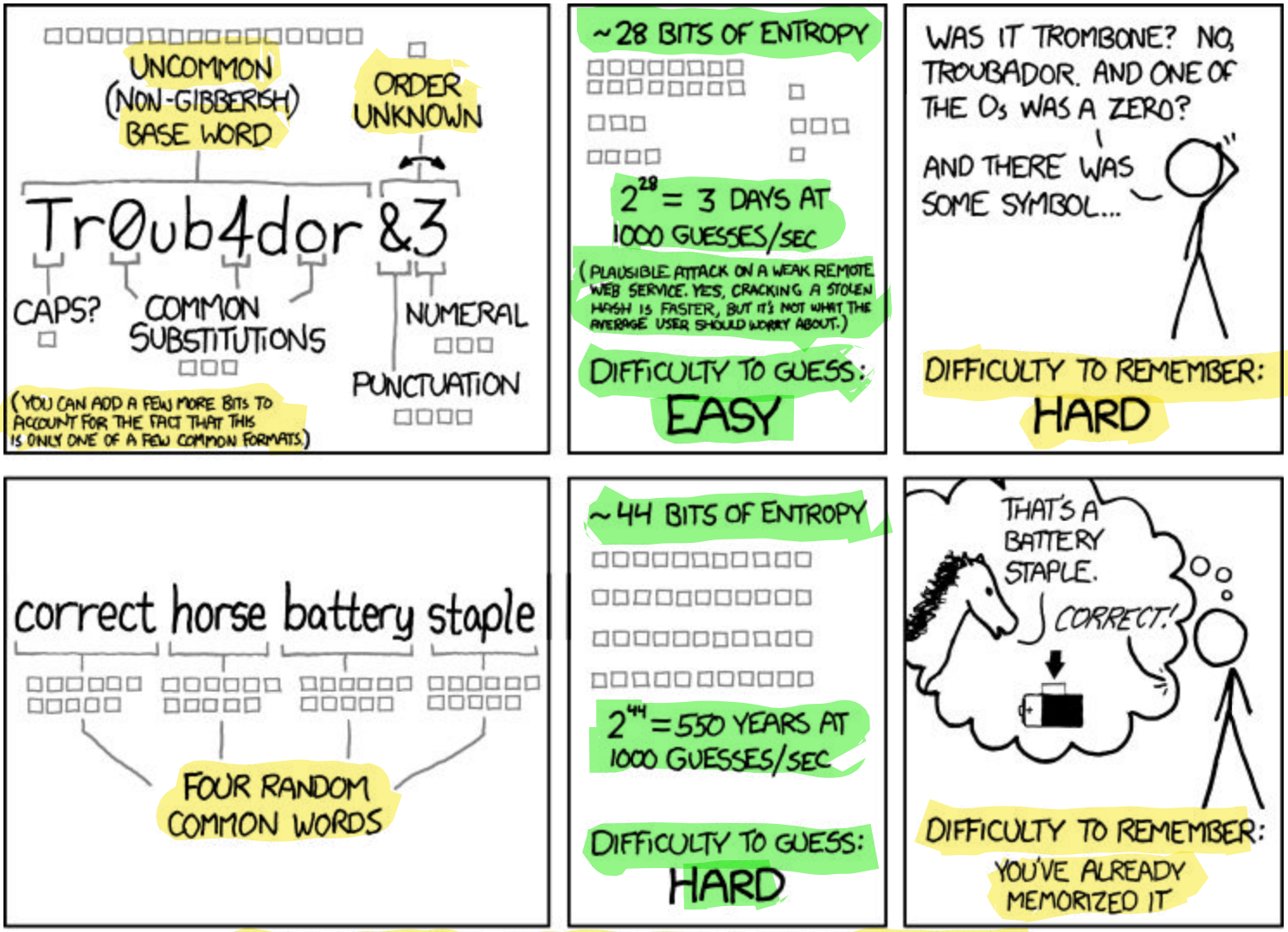
- enforce *strong* passwords in the process.
- enforce password *expiration/change* policies

Password complexity

- must h4v3 4 r1ch, ch4r4ct3r, s3t!
- mUsT hAvE a MiXeD cAsE
- muuuuust beeeeeeee looooooong enoooogh

Use **password meters** to balance *usability*.

Password Meters → HELPS IN BUILDING STRONG PASSWORD



PROBLEM: THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

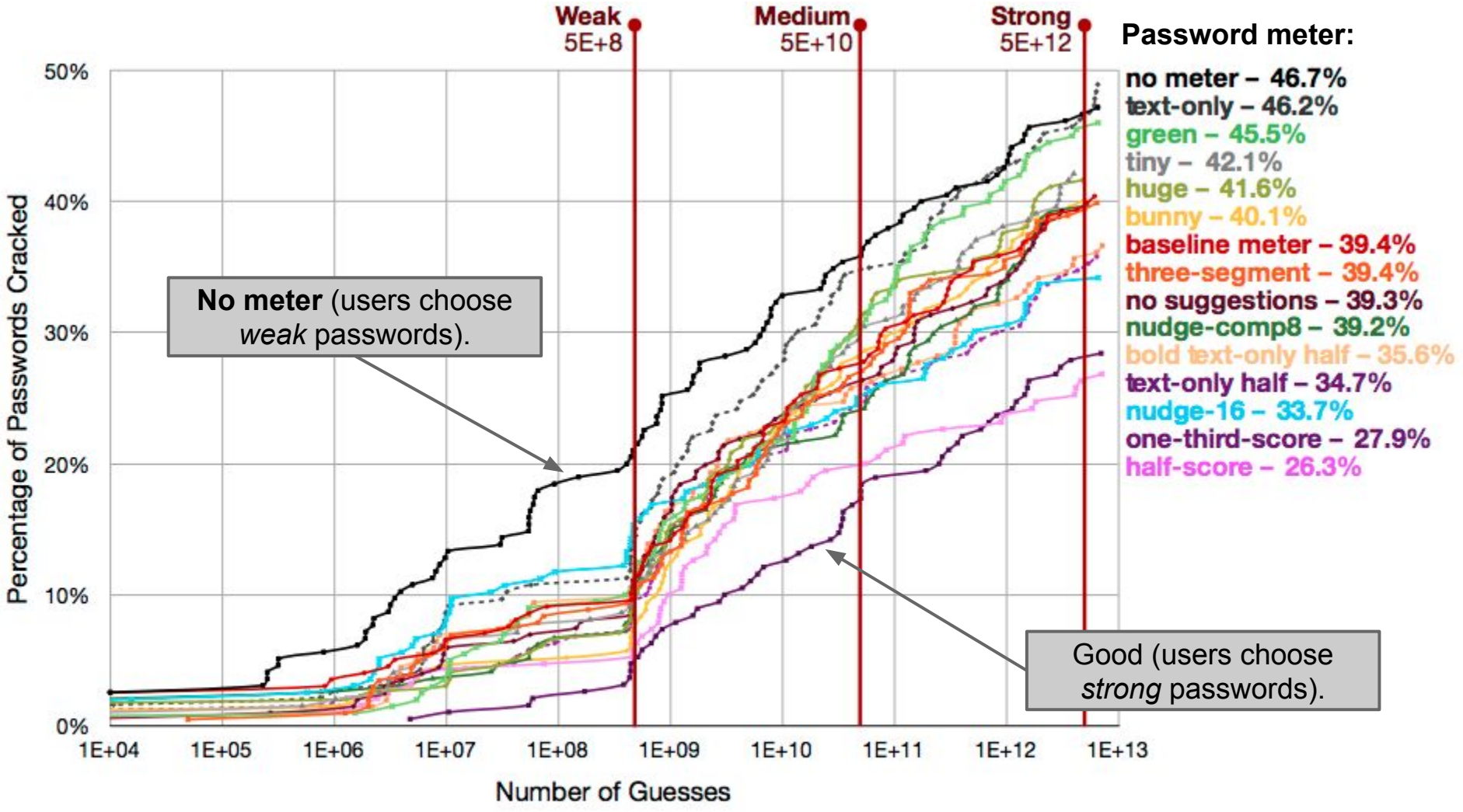


Figure 3: This graph contrasts the percentage of passwords that were cracked in each condition. The x-axis, which is logarithmically scaled, indicates the number of guesses made by an adversary, as described in Section 2.4. The y-axis indicates the percentage of passwords in that condition cracked by that particular guess number.

B. Ur, P.G. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L.F. Cranor. How does your password measure up? The effect of strength meters on password creation. USENIX Security 2012.


Enter a Password, and click Analyze
Analyze
Hide Examples
Show Options
Weak Passwords that pass typical policies:
qwerQWER12341@#$ - 11cracked - cracked7& -
Strong Passwords that fail typical policies:
udnkzdjeyhdowjpo - seattleautojesterarbol

passfault


This password needs more strength
Time To Crack:
less than 1 day
Total Passwords in Pattern:
8 Billion
HORIZONTAL
English
25%
of total strength
HORIZONTAL
English
25%
of total strength
HORIZONTAL
English
25%
of total strength
HORIZONTAL
English
25%
of total strength

Secure Password Exchange

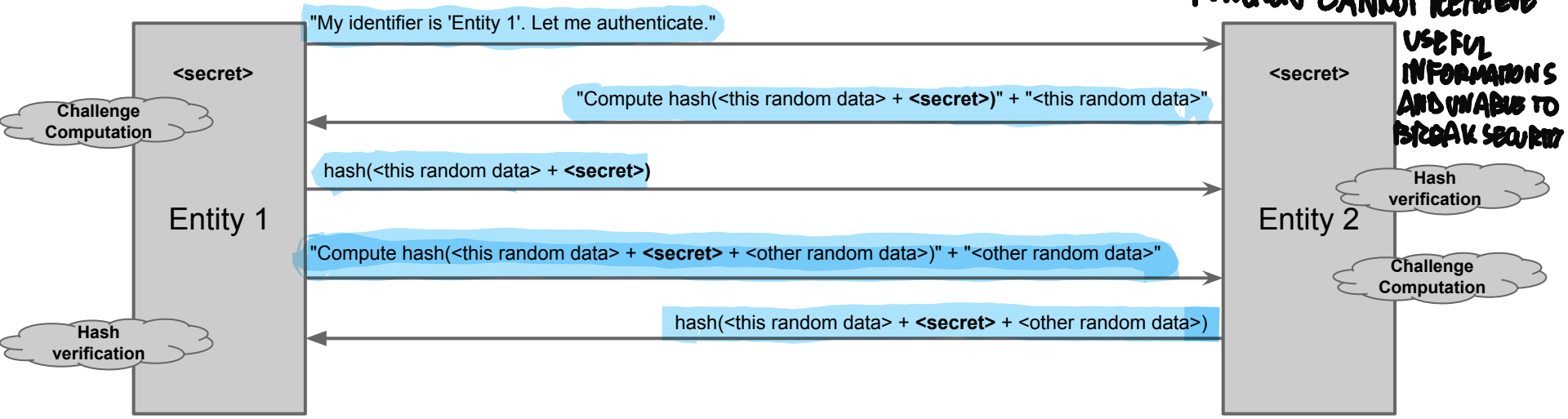
Authentication is about sharing a *secret*.

How to minimize the risk that secrets get *stolen*?

- use *mutual authentication* if possible
- use a ***challenge-response*** scheme
  - use random data to avoid *replay attacks*

EXPLOITING A HASHING MECHANISM

MAKE HASH (RANDOM DATA) ALWAYS DIFFERENT SO THAT AN ATTACKER WHO ANALYSES THE FUNCTION CANNOT RETRIEVE USEFUL INFORMATIONS AND UNABLE TO BREAK SECURITY



Secure Password Storage

OS stores a file with usernames and passwords.

An attacker could try to compromise the confidentiality and integrity of this password file

How to minimize the risk that secrets get *stolen*?

Secure Password Storage

GENERALLY STORED ON SPECIFIC FILES MANAGED BY THE O.S.

OS stores a file with usernames and passwords.

An attacker could try to compromise the confidentiality and integrity of this password file

→ PROBLEM: USE AN EFFICIENT WAY TO PROTECT

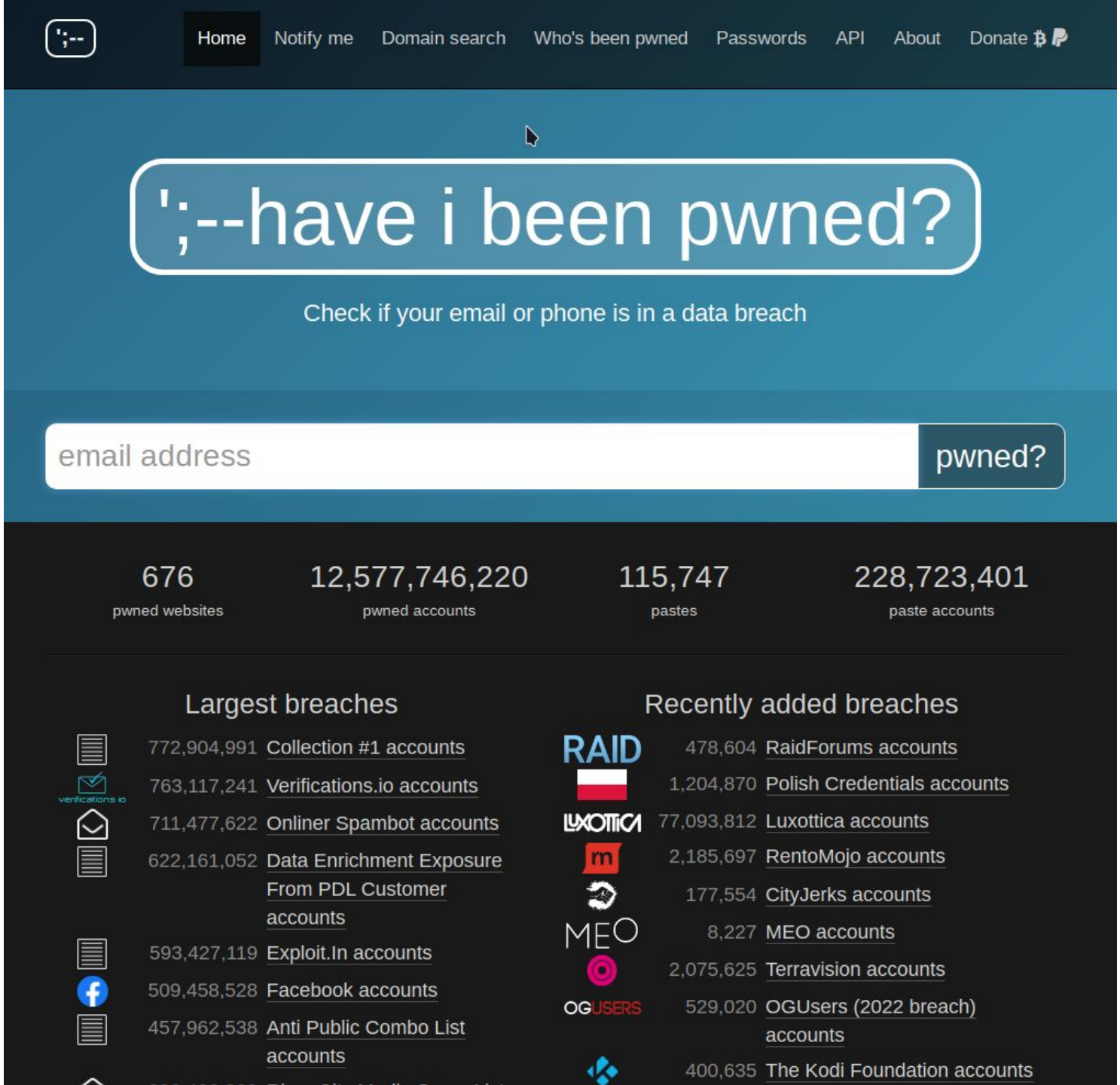
How to minimize the risk that secrets get *stolen*?

- Cryptographic protection → PROBLEM: USE AN EFFICIENT WAY TO PROTECT THE PASSWORD
- Never store passwords in clear (see next classes: hashing + salting to mitigate dictionary attacks) → WHY NOT USING HASH ALONE?
- Access control policies (privileges to w/r) CORRECT PERMISSIONS TO READ/WRITE THE FILE
- Never disclose secrets in password-recovery schemes.

EXAMPLE: DIFFERENT USERS WITH THE SAME PASSWORD WILL HAVE THE SAME HASH

IT SHOULD BE AS SECURE AS PASSWORD MANAGEMENT ITSELF

Caching problem (information is held in intermediate storage locations)


';--
Home
Notify me
Domain search
Who's been pwned
Passwords
API
About
Donate
';--have i been pwned?
Check if your email or phone is in a data breach
email address
pwned?
676
pwned websites
12,577,746,220
pwned accounts
115,747
pastes
228,723,401
paste accounts
Largest breaches
772,904,991 Collection #1 accounts
763,117,241 Verifications.io accounts
711,477,622 Onliner Spambot accounts
622,161,052 Data Enrichment Exposure From PDL Customer accounts
593,427,119 Exploit.In accounts
509,458,528 Facebook accounts
457,962,538 Anti Public Combo List accounts
Recently added breaches
478,604 RaidForums accounts
1,204,870 Polish Credentials accounts
77,093,812 Luxottica accounts
2,185,697 RentoMojo accounts
177,554 CityJerks accounts
8,227 MEO accounts
2,075,625 Terravision accounts
529,020 OGUsers (2022 breach) accounts
400,635 The Kodi Foundation accounts

The "to have" Factor

Tokens, smart cards, smart phones.

The *"to have"* Factor

User must prove that it *possesses* something.

Advantages

- Human factor (less likely to hand out a key),
- relatively low cost,
- good level of security.

Disadvantages

- Hard to deploy, → GENERALLY REQUIRES "EXTRA" HARDWARE COMPONENT
- can be lost or stolen.

Countermeasures

- none
- use with second factor.

Example Classic Technologies

One-time password generators:

- *Secret key* + *counter* synchronized with the host.
- Client: MAC-compute(counter, key).
- Host: MAC-verify(counter, key).
- Check that the counter is the expected one.
- The counter changes every 30–60 seconds.

Application examples: online banking, admin console (e.g., Amazon AWS).

Smart cards (also w/ embedded reader in USB keys)

- CPU + non-volatile RAM with a *private key*.
- The smart card authenticates itself to host via a ***challenge-response*** protocol.
- Uses the *private key* to sign the challenge.
- The *private key* does not leave the device.
- Should be **tamper proof** to some extent.

Application examples: credit cards (+PIN).

Static OTP lists (cheaper alternative)

- Known to both client and host.
- Host chooses a challenge: random numbers (e.g., "second digit of the 14th cell").
- The client transmits the response (hopefully, over an encrypted channel).
- The host should not keep the list in clear (e.g., hashing).

NEW CHALLENGE: WHILE OTP AND CARDS PERFORMS ONLY THE OPERATIONS THEY ARE DESIGNED FOR, SMARTPHONES ARE HIGHLY EXPOSED TO POTENTIAL ATTACKS

"Modern" Technology: TOTP

Software that implements the same functionality of password generators

“Modern” Technology: TOTP

Software that implements the same functionality of password generators:

- Key difference
  - password generators are *closed, embedded* systems.
  - password-generation *apps* work on *general-purpose* sw/hw platforms.
- What if the device is infected by a malicious app: Dmitrienko et al., When More Becomes Less: On the (In)Security of Mobile Two-Factor Authentication, FC 2014

EUROPOL

SIM SWAPPING – A MOBILE PHONE SCAM

SIM swapping occurs when a fraudster, using social engineering techniques, takes control over your mobile phone SIM card using your stolen personal data.

HOW DOES IT WORK?

A fraudster obtains the victim's personal data through e.g. data breaches, phishing, social media searches, malicious apps, online shopping, malware, etc.

With this information, the fraudster dupes the mobile phone operator into porting the victim's mobile number to a SIM in his possession

The fraudster can now receive incoming calls and text messages, including access to the victim's online banking

The victim will notice the mobile phone lost service, and eventually will discover they cannot login to their bank account

https://www.europol.europa.eu/activities-services/public-awareness-and-prevention-guides/sim-swapping-%E2%80%93-mobile-phone-scam

The "to be" Factor.

Biometric authentication.

The *"to be"* Factor: Biometric.

HIGHEST LEVEL OF SECURITY

User must prove that it has some specific *characteristics*.

Advantages

- high level of security,
- requires no extra hardware to carry around.

Disadvantages → GENERALLY MORE EXPENSIVE

- Hard to deploy,
- probabilistic matching,
- invasive measurement,
- can be cloned,
- bio-characteristics change,
- privacy sensitivity,
- users with disabilities.

Countermeasures

- none
- none
- none
- none (see next slides)
- re-measure often,
- secure the process,
- need alternate (weaker?)

Technology examples

Extract the characteristics (i.e., features) of:

- Fingerprints (http://www.freedesktop.org/wiki/Software/fprint/)
- Face geometry (https://code.google.com/p/pam-face-authentication/)
- Hand geometry (palm print)
- Retina scan
- Iris scan
- Voice analysis
- DNA
- Typing dynamics (http://flyer.sis.smu.edu.sg/ndss13-tey.pdf)
- Grasp Smartphones (https://www.sciencedirect.com/science/article/pii/S1877050919313845)

Example: Fingerprint

- *Enrollment*: reference sample of the user's fingerprint is acquired at a fingerprint reader.
- Features are derived from the sample.
  - *Fingerprint minutiae*: end points of ridges, bifurcation points, core, delta, loops, whorls, ?
  - For higher accuracy, record features for more than one finger and different positions.
- Feature vectors are stored in a secure database.
- When the user logs on, a new reading of the fingerprint is taken; features are compared against (similarity) the reference features. User is accepted if match is above a predefined threshold.

Main issue: false positives and false negatives

Consumer-level Biometric Auth

http://cryptome.org/fake-prints.htm

Manufacturer	Model	Technology	Date	Difficulty
Identix	TS-520	Optical	Nov. 1990	First attempt
Fingermatrix	Chekone	Optical	Mar. 1994	Second attempt
Dermalog	DemalogKey	Optical	Feb.1996	First attempt
STMicroelectronics	TouchChip	Solid state	Mar. 1999	First attempt
Veridicon	FPS110	Solid state	Sept.1999	First attempt
Identicator	DFR200	Optical	Oct. 1999	First attempt

20 september 2013 RELEASED

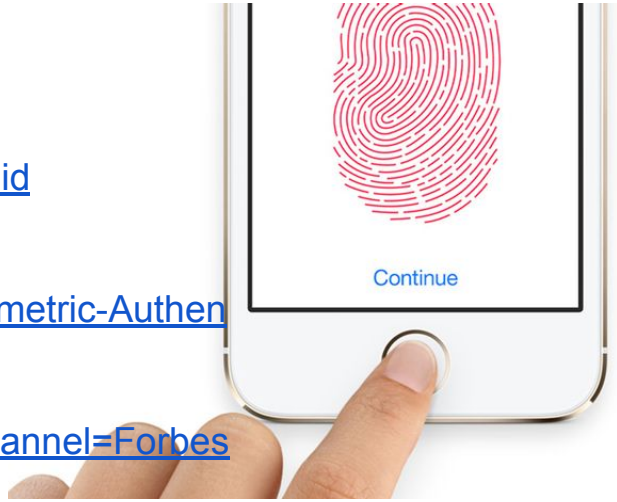
21 september 2013 CRACKED

http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid

https://www.ccc.de/en/updates/2017/iriden

https://i.blackhat.com/USA-19/Wednesday/us-19-Chen-Biometric-Authentication-Under-Threat-Liveness-Detection-Hacking.pdf

https://www.youtube.com/watch?v=ZwCNG9KFdXs&ab_channel=Forbes



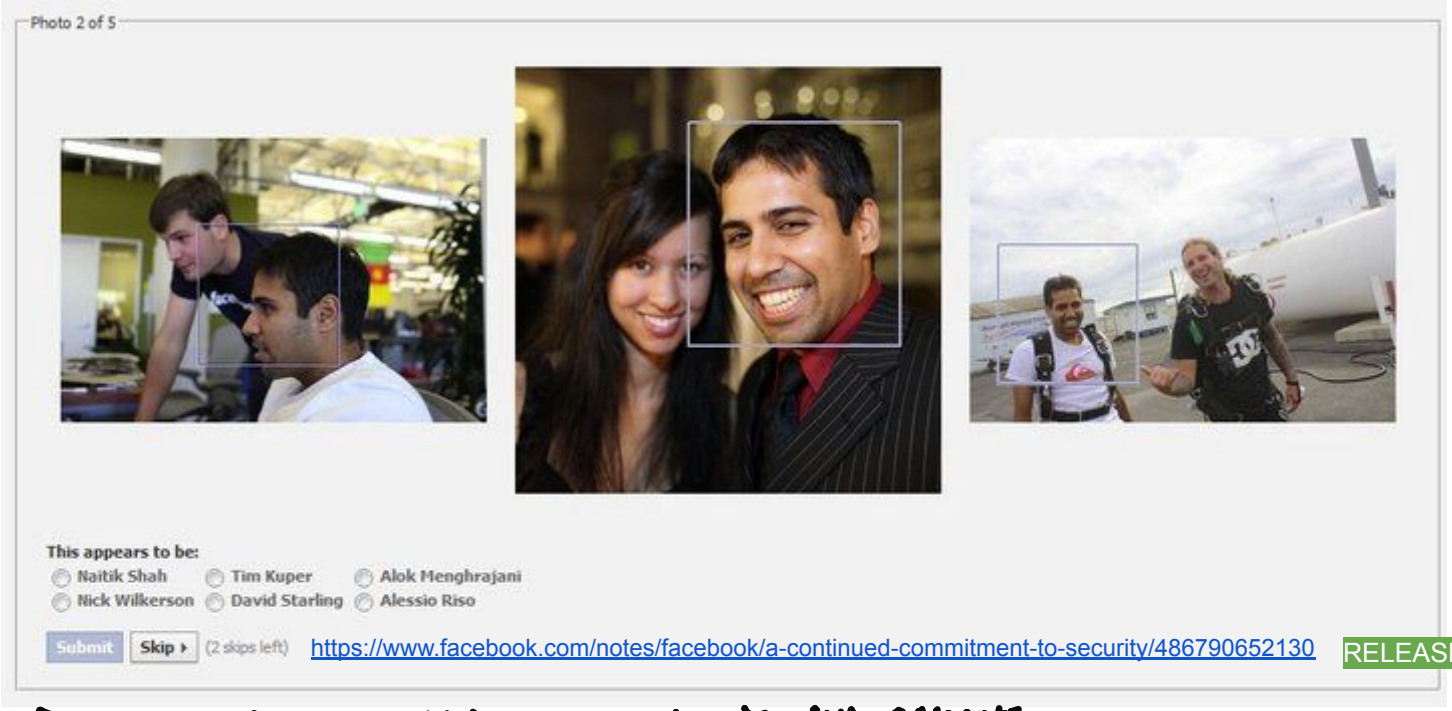
Novel (and Experimental) factors of authentication

NEW PROPOSED SOLUTIONS

POTENTIAL THIRD LEVEL AUTHENTICATION

The *"social"* Factor: Who you know.

Alice must prove that she *knows someone.*



RELEASED

Papers PROBLEM: THOSE INFORMATIONS CAN BE FOUND ONLINE

- H. Kim, J. Tang, and R. Anderson. *Social authentication: harder than it looks.* In Proceedings of the 2012 Financial Cryptography and Data Security conference.

CRACKED J. Polakis, M. Lancini, G. Kontaxis, F. Maggi, S. Ioannidis, A. Keromytis, S. Zanero, *All Your Face Are Belong to Us: Breaking Facebook's Social Authentication.* In Proceedings of 2012 Annual Computer Security Applications Conference.

Single Sign On

Problem: managing and remembering multiple passwords is complex.

- Users re-use passwords over multiple sites,
- Password policies replicated ($$$).

SINGLE SIGN-ON SERVICE

Solution: 1 identity, 1-2 auth. factors, 1 trusted host.

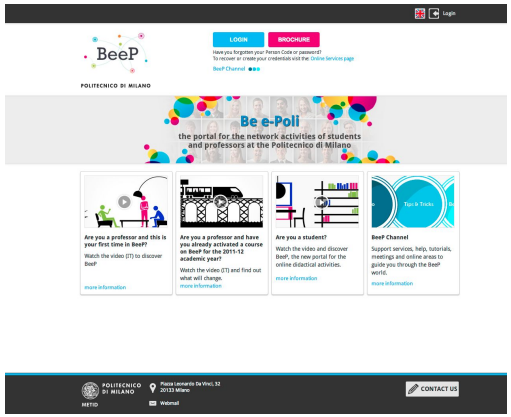
- elect a trusted host,
- users authenticate (sign on) on the trusted host,
- other hosts ask the trusted host if a user is authenticated.

Example: Shibboleth (AunicaLogin)

User (e.g., you)

1. I am "John G. Student"

6. You can access the service.



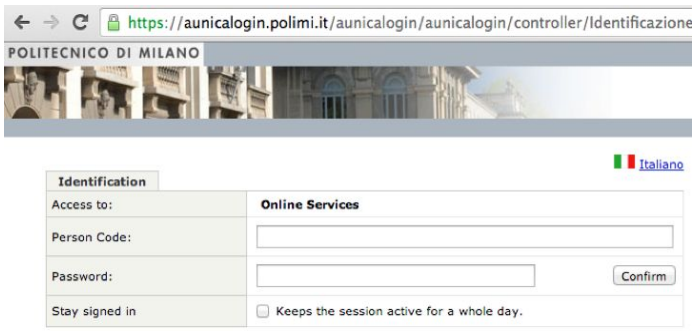
Service (e.g., BeeP)

2. Is "John G. Student" a student?

3. I need your credentials.

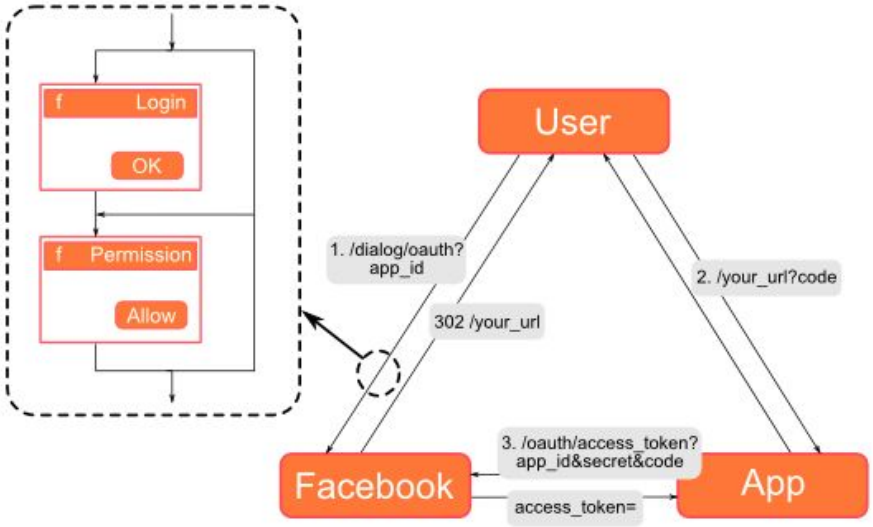
4. Here is my username and password

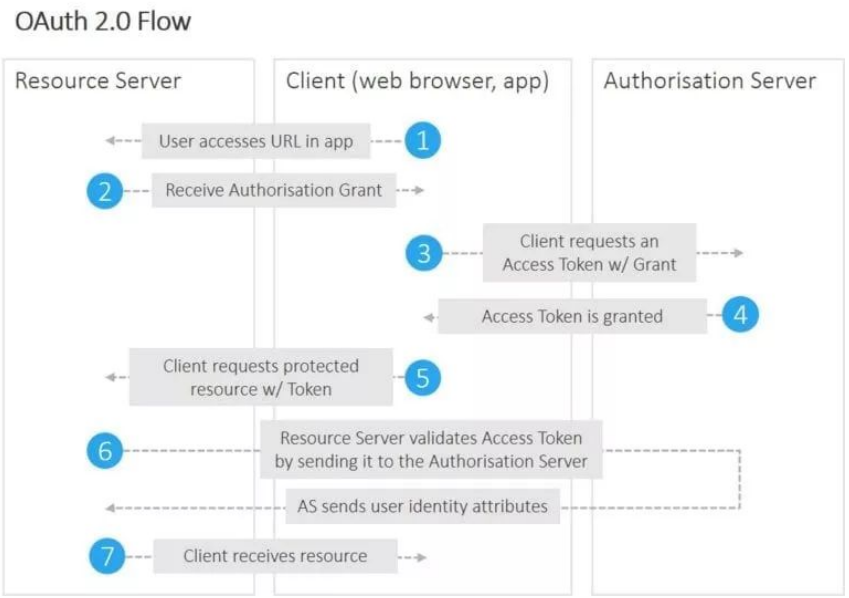
5. Yes. I confirm.



Identity Provider (e.g., AunicaLogin)

Example: OAuth2 Flow (Facebook)





Single Sign On: challenges

Single point of *trust*: the trusted server.

- If compromised, all sites are compromised.
- Password reset scheme must be bulletproof.
  - Email is the trusted element

Kontaxis G. et al., *SAuth: Protecting User Accounts from Password Database Leaks*. In Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS), 2013.

Difficult to get right for the developers.

- The flow is complex to implement.
- Libraries exist, but they can be bugged.

http://homakov.blogspot.it/2014/02/how-i-hacked-github-again.html

Password Managers [~Déjà vu]

SOFTWARE THAT STORES AND CAN GENERATE PASSWORDS

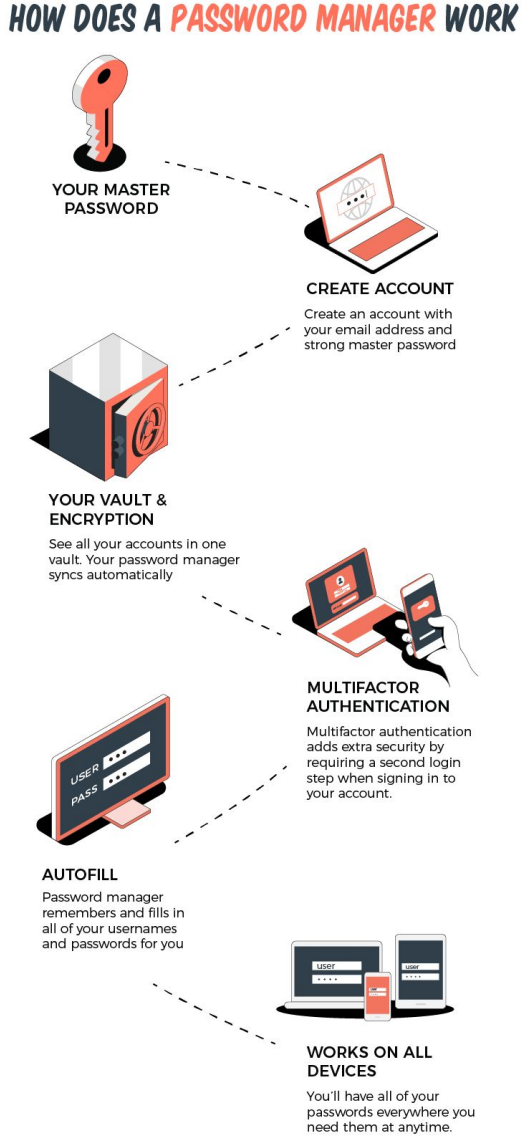
Problem: managing and remembering multiple passwords is complex.

- Users re-use passwords over multiple sites,
- Password policies replicated ($$$).

Solution: 1 identity, 1-2 auth. factors, 1 trusted host.

- elect a trusted host,
- users authenticate (sign on) on the trusted host with a <u>master password</u>

AVOID NON OPEN-SOURCE PASSWORD MANAGERS AND THE ONES THAT ARE NOT CLEAR ENOUGH ON HOW THEY STORE DATA



Password Managers

Pros

- No need to remember all passwords
- It allows generating robust passwords
- Usability
  - Auto..
    - Fill
    - Synch
  - Multiple Devices

Cons

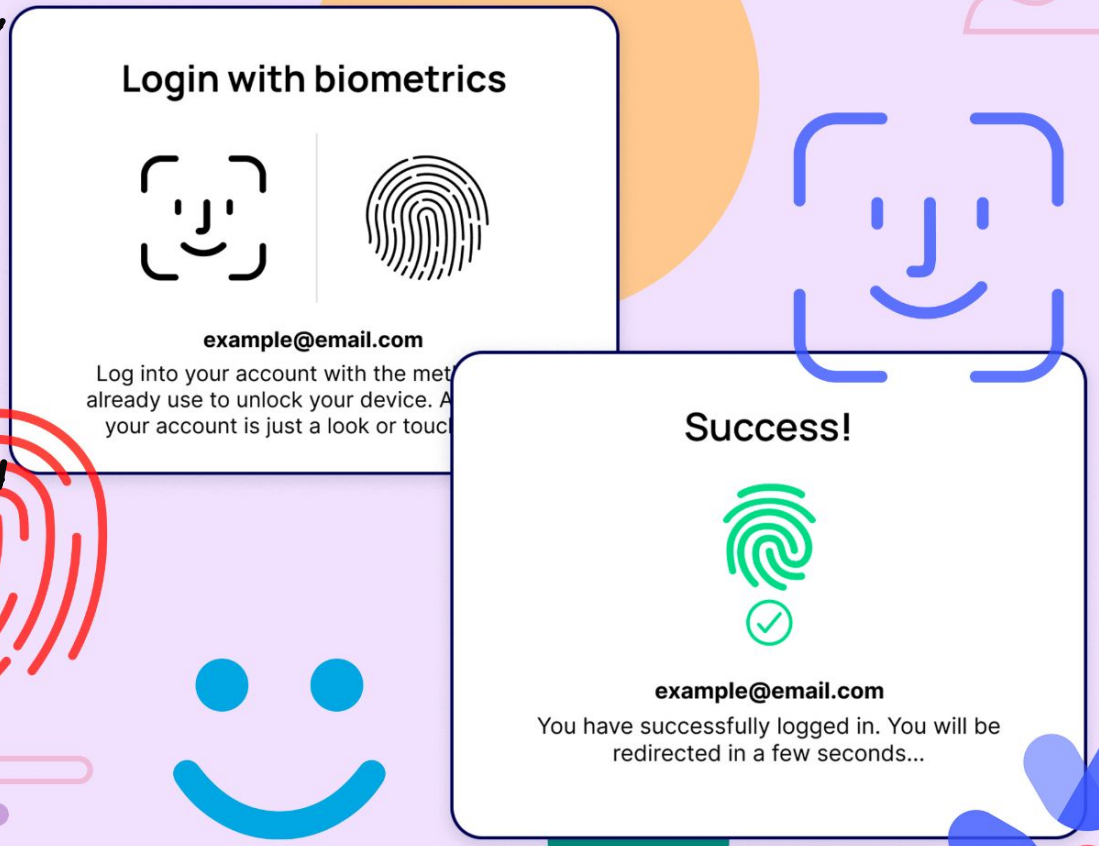
- Trusting single sign-on
- A single point of failure
- Larger attack surface
  - Password managers are softwares...

https://stuartschechter.medium.com/before-you-use-a-password-manager-9f5949ccf168

Passwordless Auth - Passkeys

ASK TO AN EXTERNAL APP/DEVICE TO DIGITALLY SIGN A CHANNEL AND AUTHENTICATE THE USER

UNLOCK PRIVATE KEY AND PERFORM LOGIN USING ASYMMETRIC ENCRYPTION



EXAMPLES:

- BIOMETRIC
- AUTHENTICATION EMAIL/SMS

Conclusions

Identification, *authentication* and *authorization* are three distinct, yet inter-dependent, concepts.

There are *three* types of authentication *factors*, which should be used in *combination*.

Passwords are increasingly showing their limits.

New authentication schemes are promising, but should be used with care.