



# **3. Authentication**

Computer Security Courses @ POLIMI

# Identification vs. Authentication

1) **Identification:** an entity declares its identifier

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

```
Ubuntu 12.04.3 LTS ubuntu64 tty1
ubuntu64 login: foobar
```

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

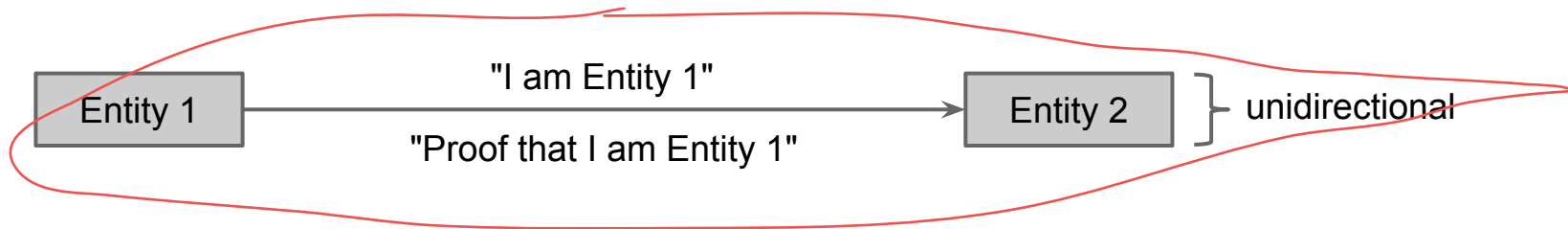
- **Examples:** *"Here is Stefano's ID card"*

```
Ubuntu 12.04.3 LTS ubuntu64 tty1
ubuntu64 login: foobar
Password: 
```

3) Authorization

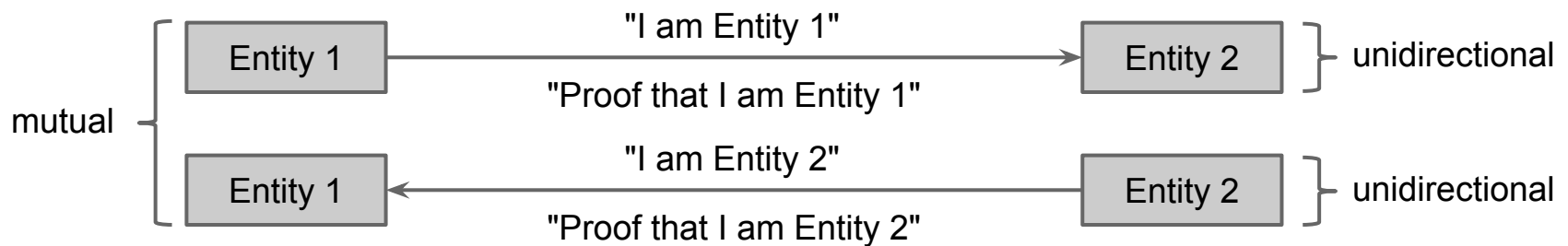
# Authentication

Can be *unidirectional*



# 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.

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.

**Humans:**

**Machines:**

# Three Factors of Authentication

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

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

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.

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

**Machines:**

# Three Factors of Authentication

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

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

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.

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

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

# Three Factors of Authentication

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

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

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.

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

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

*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

# 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

- [stolen/snooped](#)
- guessed
- cracked (enumerated)



SEPPA MAZZINI



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)



# 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**

*complexity*

*change*

*being related to users*

Against **cracking**

*complexity*

*change*

*being related to users*

Against **guessing**

*complexity*

*change*

*not being related to users*

# 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**

**complexity**

**change**

**being related to users**

Against **cracking**

*complexity*

*change*

being related to users

Against **guessing**

*complexity*

*change*

*not being related to users*

# 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**

**complexity**

**change**

**being related to users**

Against **cracking**

**complexity**

**change**

**being related to users**

Against **guessing**

*complexity*

*change*

*not being related to users*

# 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**

**complexity**

**change**

**being related to users**

Against **cracking**

**complexity**

**change**

**being related to users**

Against **guessing**

**complexity**

**change**

**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 loooooong enoooogh

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



# Password Meters

.....|



Password must:

- ☒ Have at least one letter
- ☐ **Have at least one capital letter**
- ☐ **Have at least one number**
- ☒ Not contain more than 3 consecutive identical characters
- ☒ Not be the same as the account name
- ☒ Be at least 8 characters

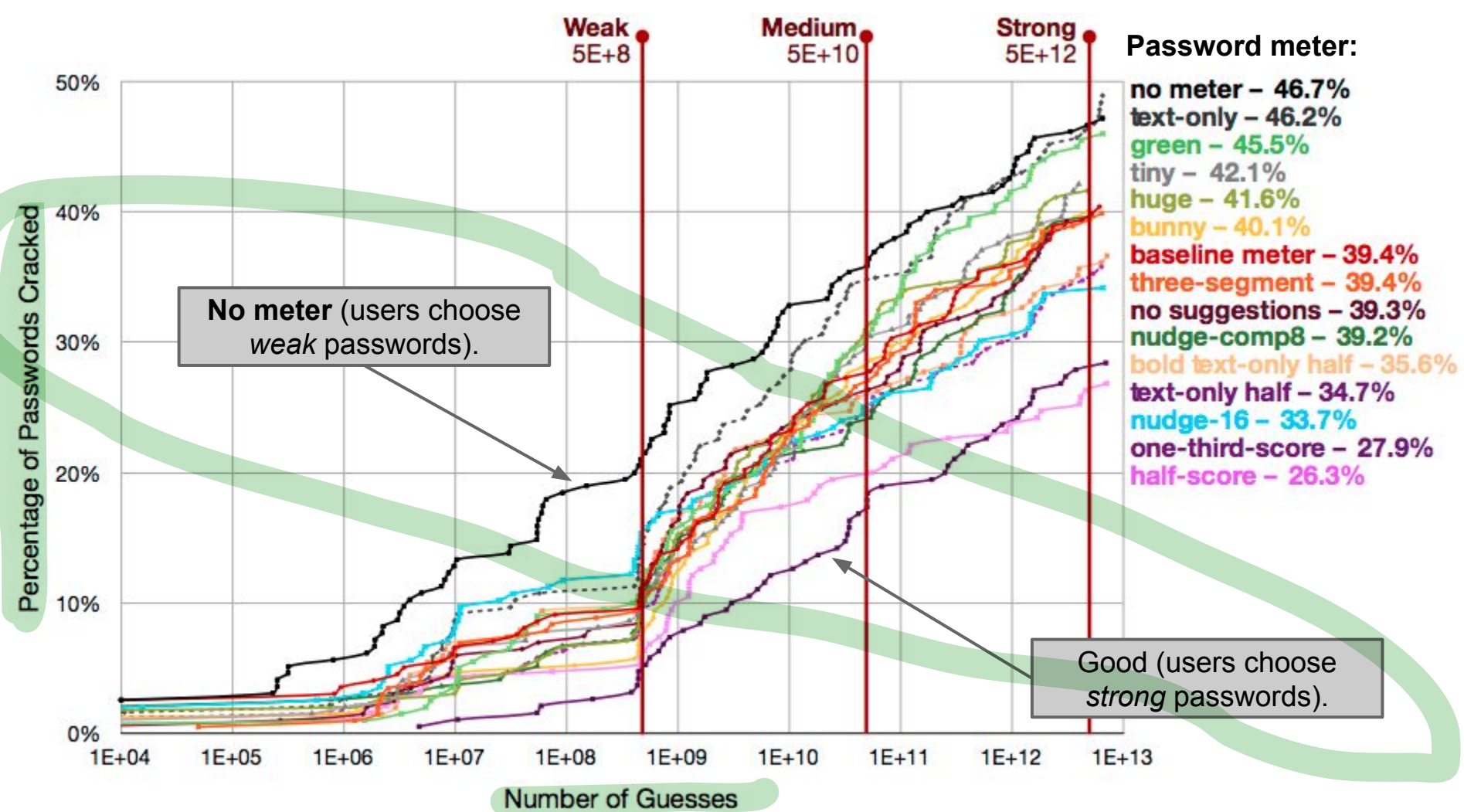
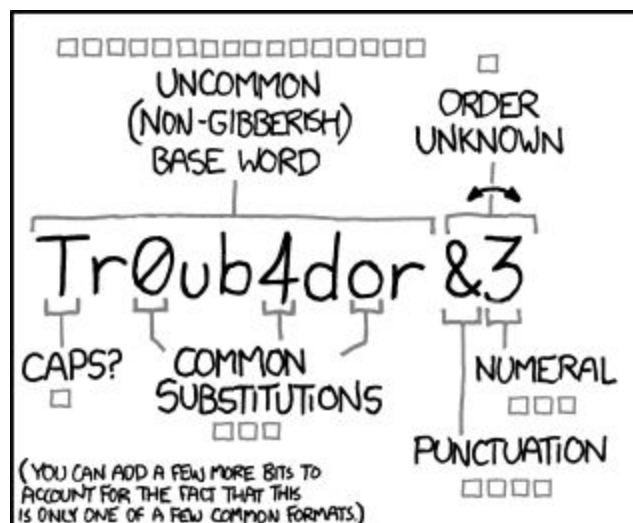


Figure 3: This graph contrasts the percentage of passwords that were cracked in each condition. The x-axis, which is logarithmically scaled, indicates the number of guesses made by an adversary, as described in Section 2.4. The y-axis indicates the percentage of passwords in that condition cracked by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



~28 BITS OF ENTROPY

$2^{28} = 3 \text{ DAYS AT } 1000 \text{ GUESSES/SEC}$

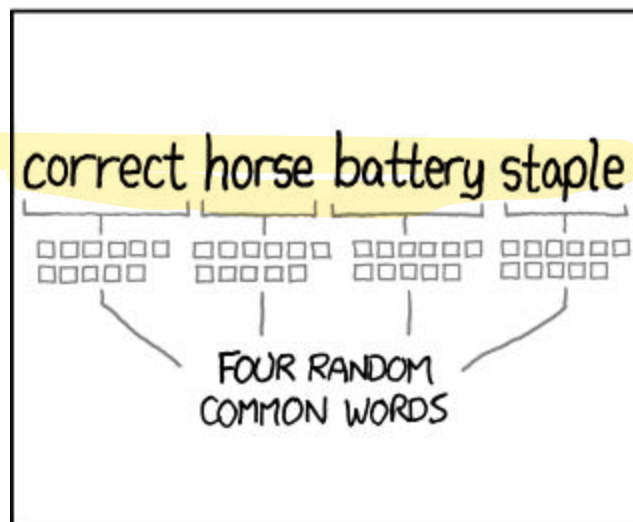
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)

DIFFICULTY TO GUESS: EASY

WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE 0s WAS A ZERO?

AND THERE WAS SOME SYMBOL...

DIFFICULTY TO REMEMBER: HARD



~44 BITS OF ENTROPY

$2^{44} = 550 \text{ YEARS AT } 1000 \text{ GUESSES/SEC}$

DIFFICULTY TO GUESS: HARD

THAT'S A BATTERY STAPLE.

CORRECT!

DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT

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.



Enter a Password, and click Analyze

.....

Analyze

Hide Examples

Show Options

Weak Passwords that pass typical policies:

qwerQWER1234!@#\$ - l1cracked - cracked7& -

Strong Passwords that fail typical policies:

udnkzdejhdowjpo - 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

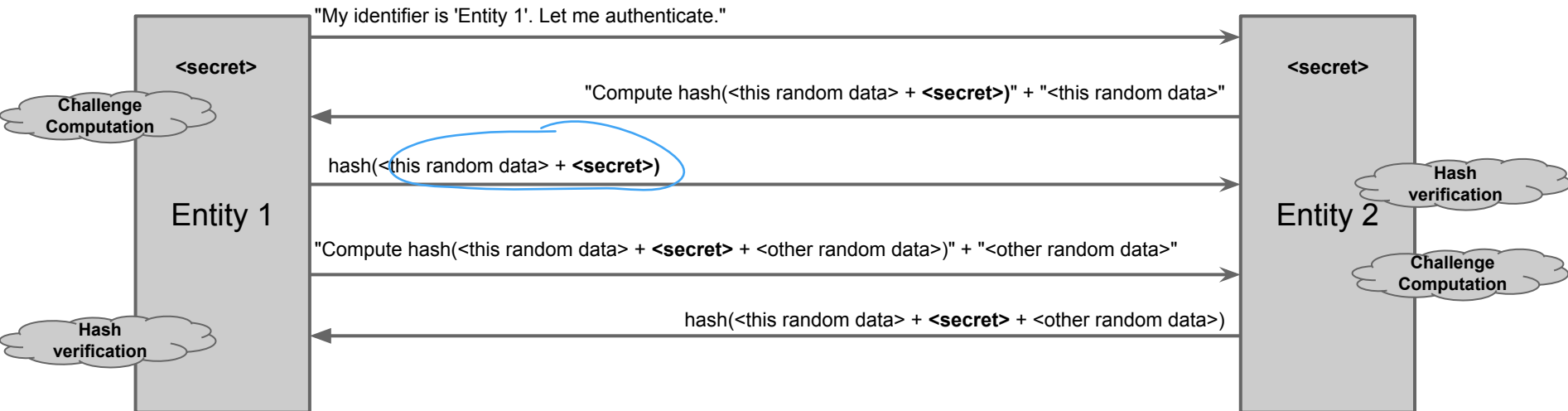


# Secure Password Exchange

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*



# 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

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*?

- Cryptographic protection
  - Never store passwords in clear (see next classes: hashing + salting to mitigate dictionary attacks)
- Access control policies (privileges to w/r)
- Never disclose secrets in password-recovery schemes.

Caching problem (information is held in intermediate storage locations)

[Home](#)[Notify me](#)[Domain search](#)[Who's been pwned](#)[Passwords](#)[API](#)[About](#)[Donate !\[\]\(0b5e7e25e8775f7e7e80906ada4f0021\_img.jpg\)](#)

# ';--have i been pwned?

Check if your email or phone is in a data breach

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](#)



verifications.io

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

RAID

478,604 [RaidForums accounts](#)



1,204,870 [Polish Credentials accounts](#)

LUXOTTICA

77,093,812 [Luxottica accounts](#)



2,185,697 [RentoMojo accounts](#)



177,554 [CityJerks accounts](#)

MEO

8,227 [MEO accounts](#)



2,075,625 [Terravision accounts](#)

OGUSERS

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

## Countermeasures

# 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,
- 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:  $\text{MAC-compute}(\text{counter}, \text{key})$ .  
*(Handwritten: NUMBER = counter)*
- Host:  $\text{MAC-verify}(\text{counter}, \text{key}, \text{NUMBER})$ .  
*(Handwritten: NUMBER)*
- 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)



► Esempio di codice personale



Esempio di codice personale

- 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).

# “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



# 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





# **The "to be" Factor.**

Biometric authentication.



# The "*to be*" Factor: Biometric.

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

## Advantages

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

## Disadvantages

- 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**

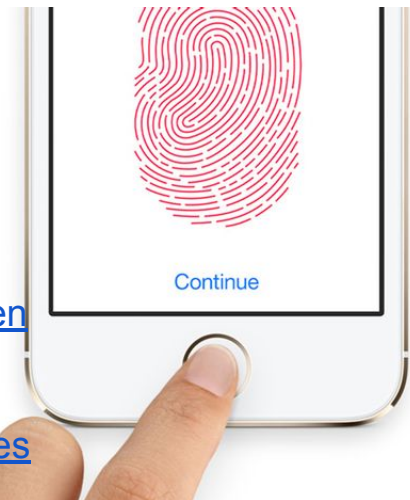
21 september 2013 **CRACKED**

<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>

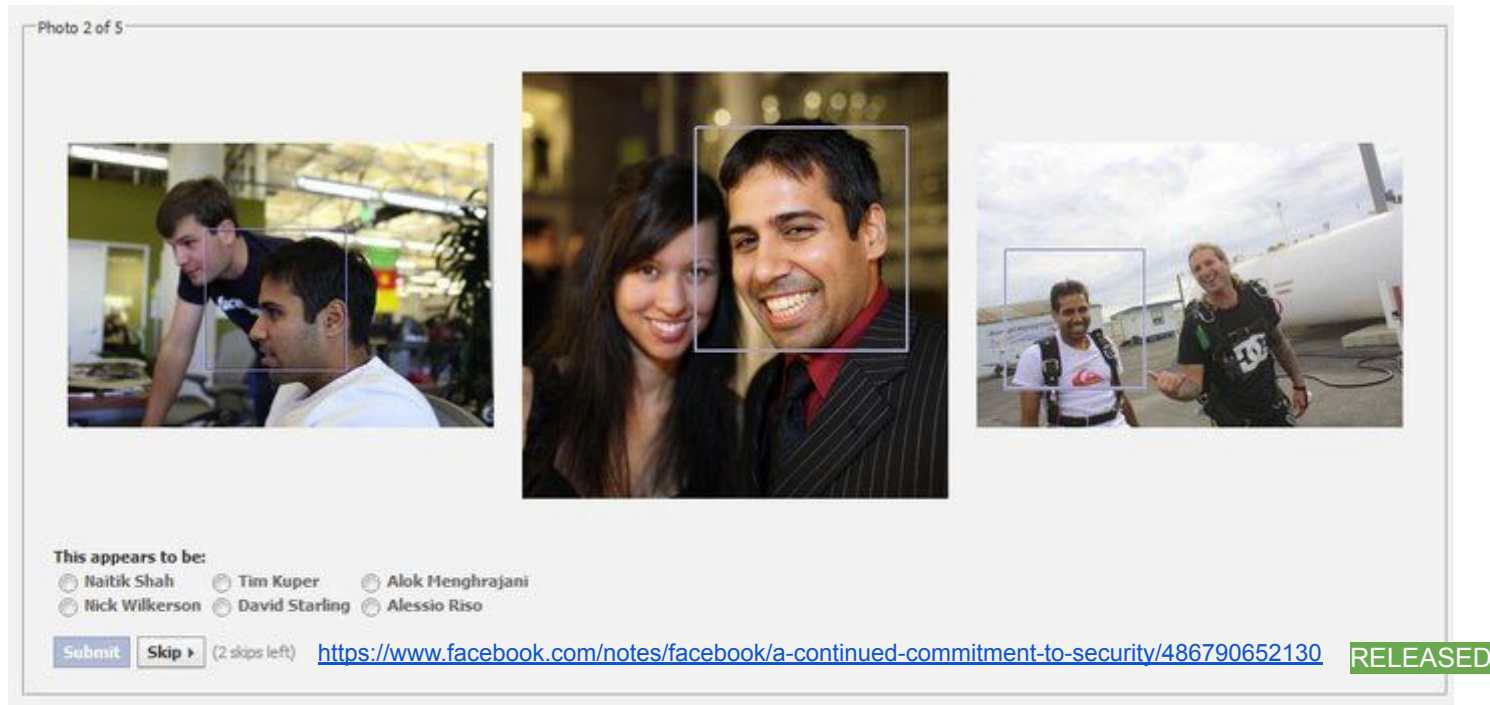
[https://www.youtube.com/watch?v=ZwCNG9KFdXs&ab\\_channel=Forbes](https://www.youtube.com/watch?v=ZwCNG9KFdXs&ab_channel=Forbes)



**Novel (and Experimental)  
factors of authentication**

# The "social" Factor: Who you know.

Alice must prove that she *knows someone*.



## Papers

- H. Kim, J. Tang, and R. Anderson. [Social authentication: harder than it looks](#). In Proceedings of the 2012 Financial Cryptography and Data Security conference.
- J. Polakis, M. Lancini, G. Kontaxis, E. 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.

CRACKED

# Single Sign On

**Problem:** managing and remembering multiple passwords is complex.

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

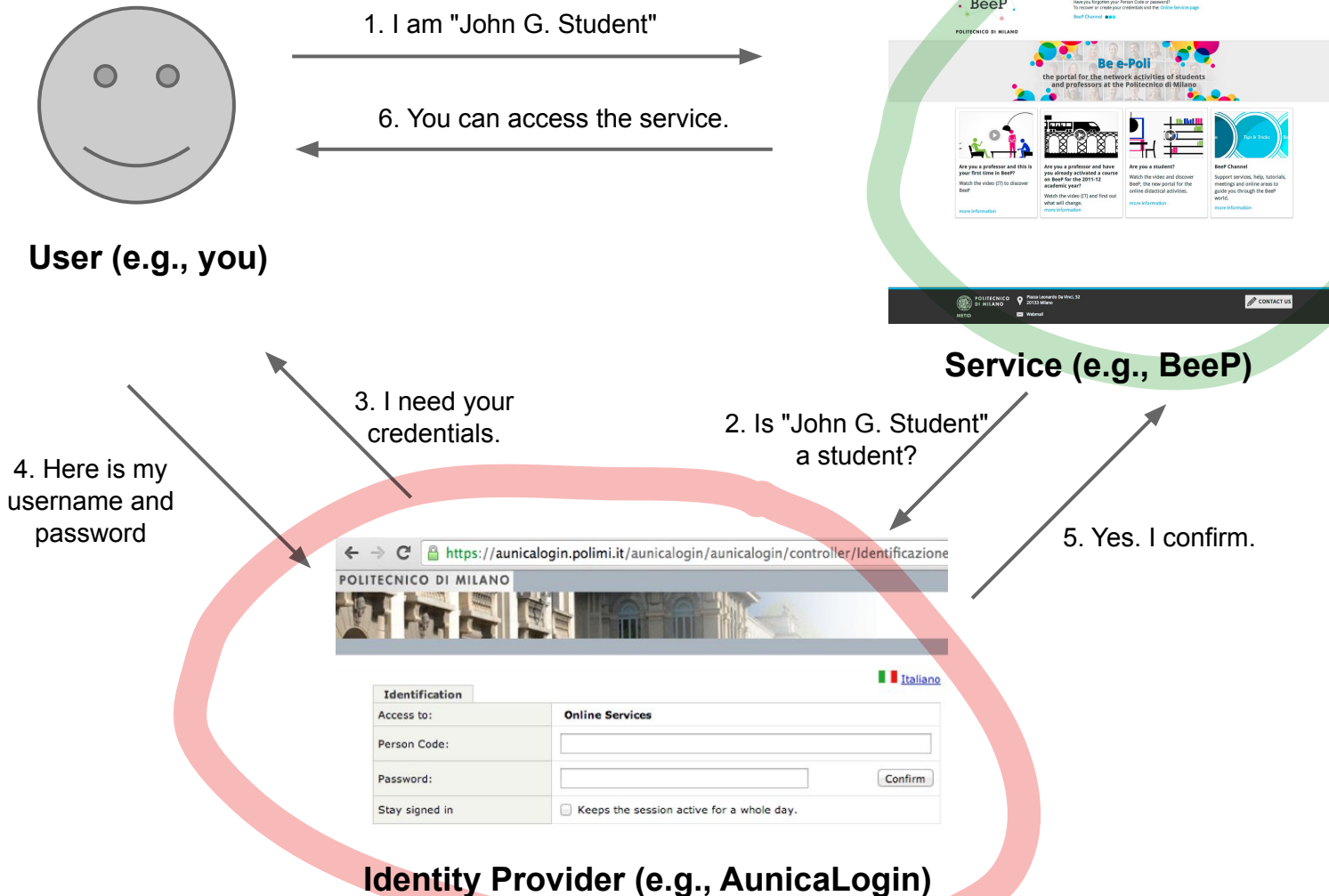
**Solution:** 1 identity, 1-2 auth. factors, 1 trusted host.

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

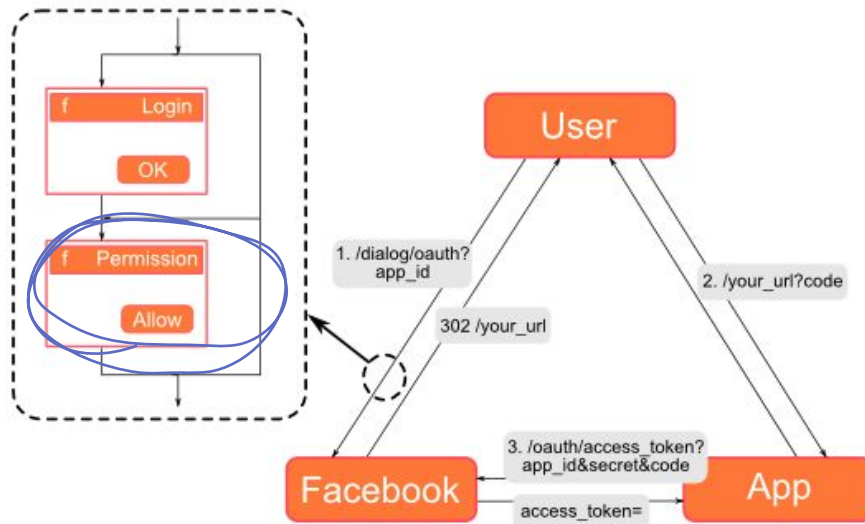




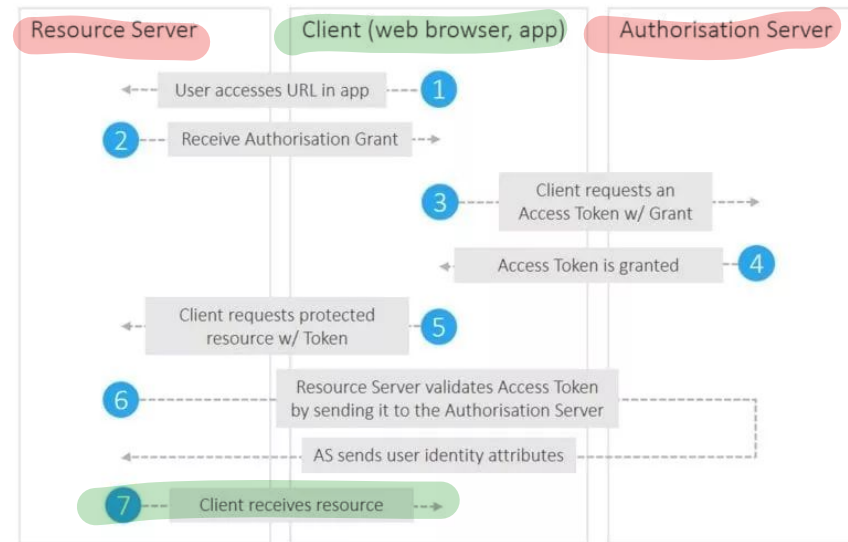
# Example: Shibboleth (AunicaLogin)



# Example: OAuth2 Flow (Facebook)



OAuth 2.0 Flow





# 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]

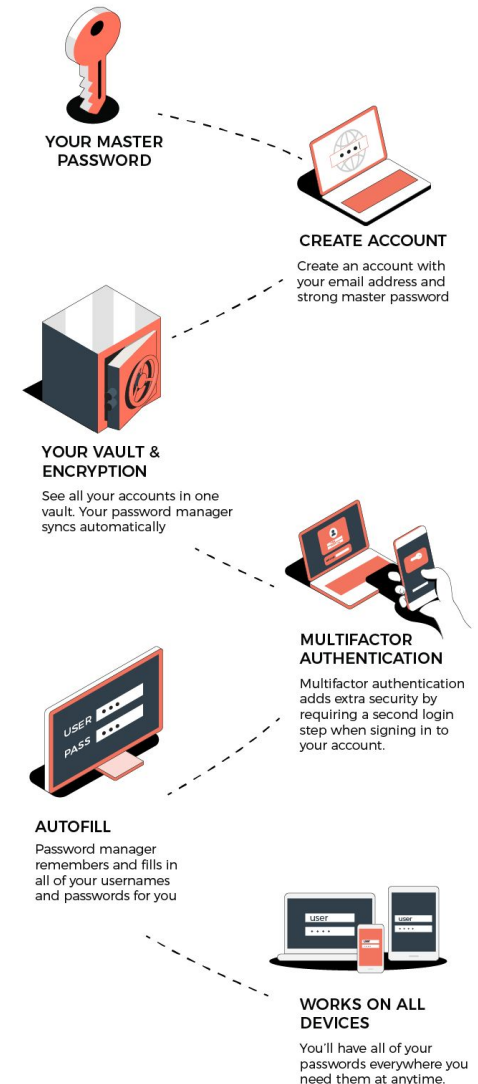
**Problem:** managing and remembering multiple passwords is complex.

- 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 master password

HOW DOES A **PASSWORD MANAGER** WORK



# 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...

# Passwordless Auth - Passkeys

Login with biometrics



example@email.com

Log into your account with the method you already use to unlock your device. Accessing your account is just a look or touch.

Success!



example@email.com

You have successfully logged in. You will be redirected in a few seconds...

# 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.