GameSS

AUTHENTICATION

WHO IS THE MATERIAL FOR?

This material provides an introduction to the concepts of Authentication. It is suitable for professionals such as

- Office employees
- Developers
- Managers

WHO MADE THIS MATERIAL?

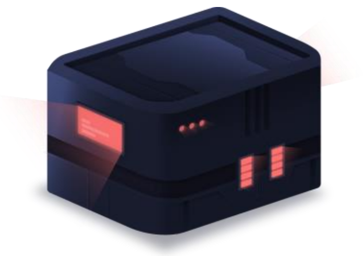
Peter Mayer

Assistant Professor at the Department of Mathematics and Computer Science at the University of Southern Denmark (SDU)

mayer@imada.sdu.dk

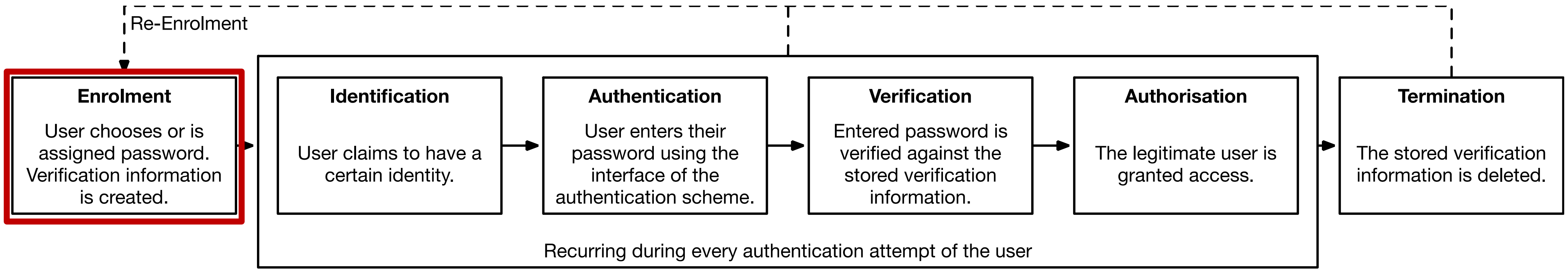
WHAT ARE THE MAIN TAKEAWAYS FOR THIS CONTENT?

- Office employees
- Developers
- Managers



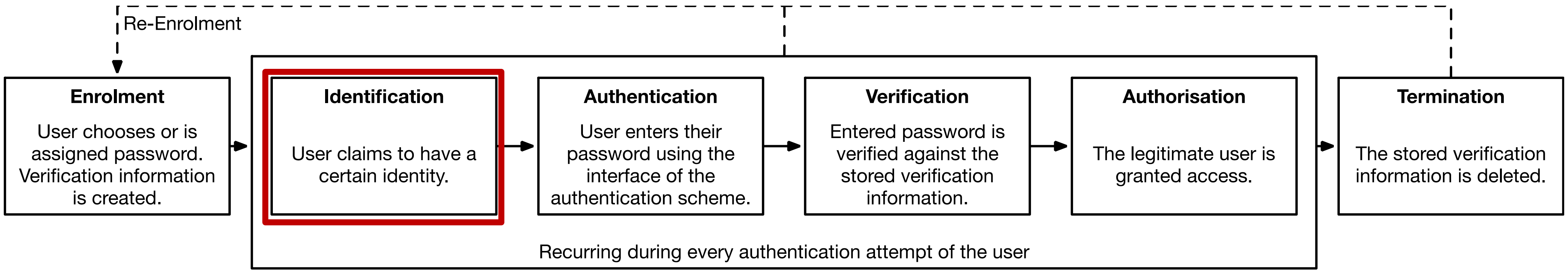
WHAT IS USER AUTHENTICATION?

THE USER ACCESS CONTROL PROCEDURE



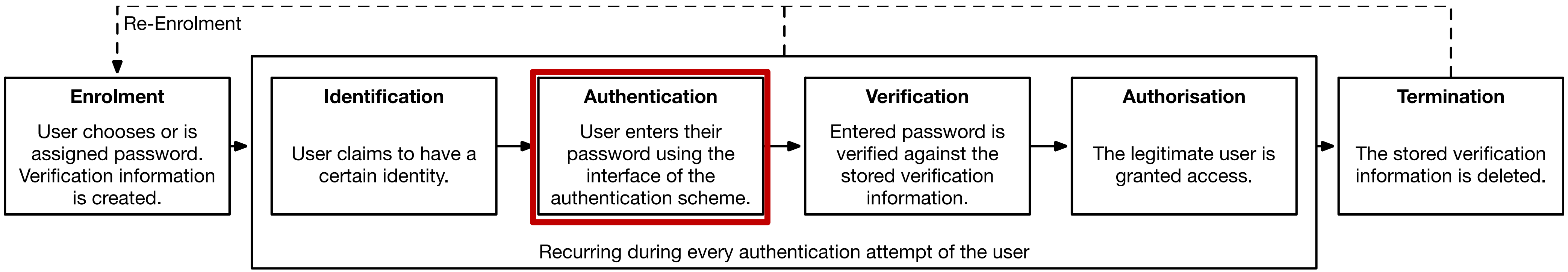
Goal: Only the genuine user can prove their identity and gain access to protected resources

THE USER ACCESS CONTROL PROCEDURE



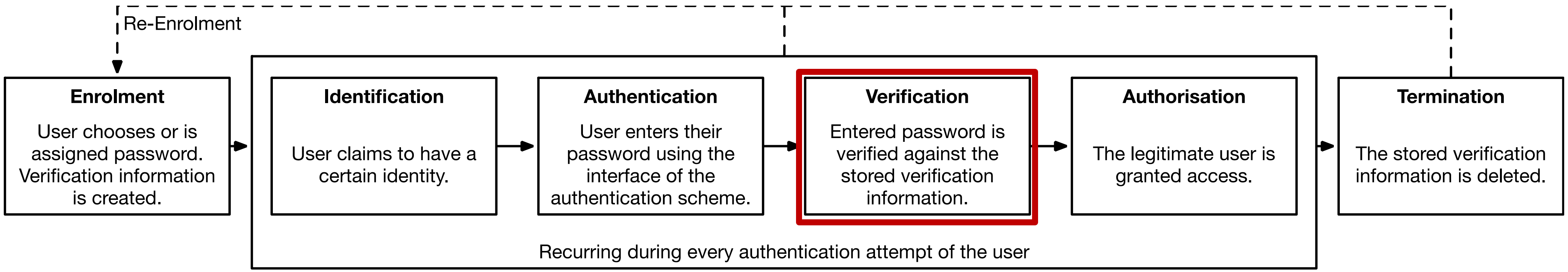
Goal: Only the genuine user can prove their identity and gain access to protected resources

THE USER ACCESS CONTROL PROCEDURE



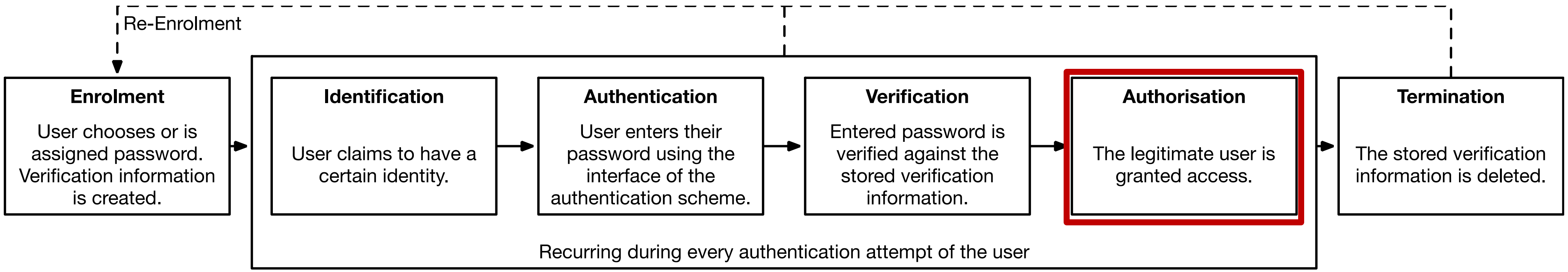
Goal: Only the genuine user can prove their identity and gain access to protected resources

THE USER ACCESS CONTROL PROCEDURE



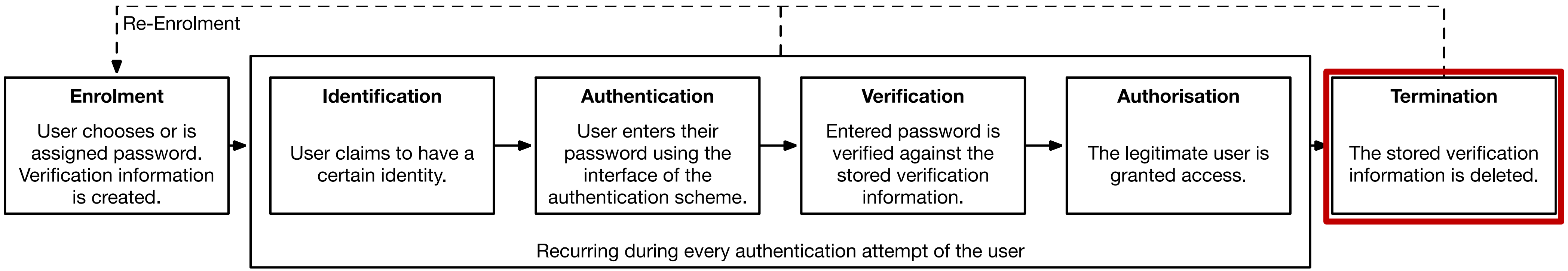
Goal: Only the genuine user can prove their identity and gain access to protected resources

THE USER ACCESS CONTROL PROCEDURE



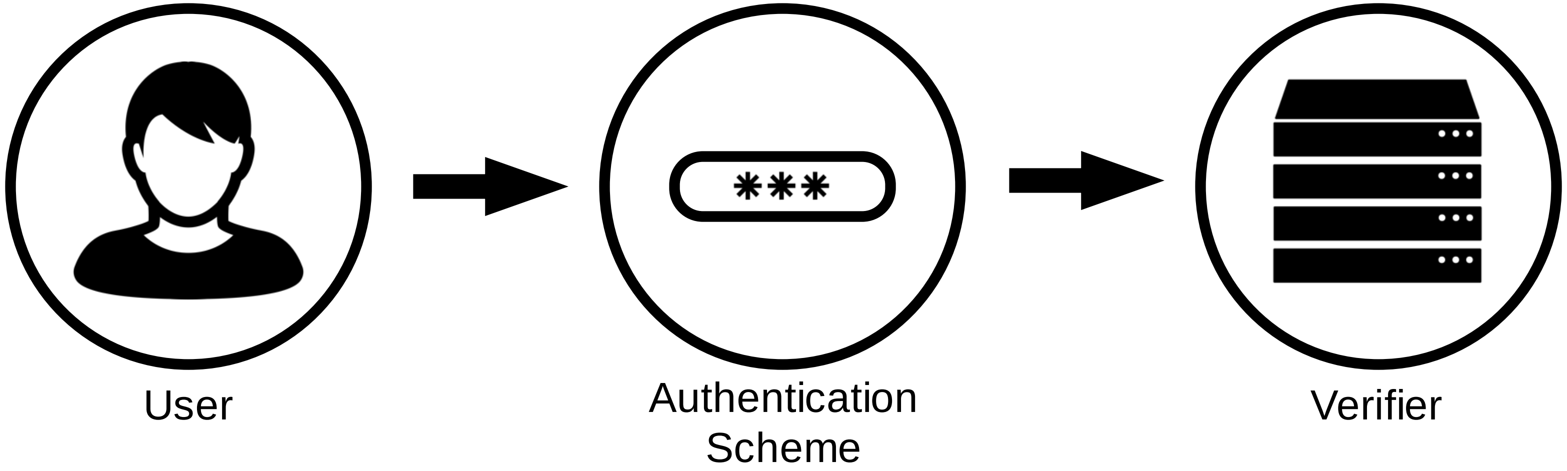
Goal: Only the genuine user can prove their identity and gain access to protected resources

THE USER ACCESS CONTROL PROCEDURE



Goal: Only the genuine user can prove their identity and gain access to protected resources

THE ENTITIES OF THE AUTHENTICATION PROCEDURE



TYPES OF USER AUTHENTICATION SCHEMES

- Authentication by **inherence** (Biometrics)

- Authentication by **possession** (Tokens)

- Authentication by **knowledge** (Cognometrics)

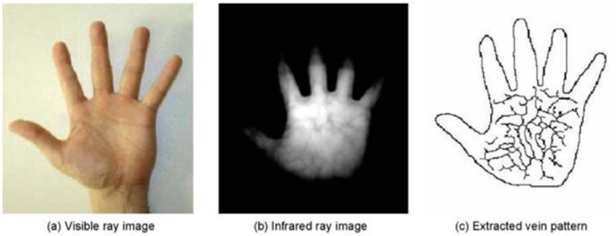
Multi-factor authentication (MFA)
Multiple factors can be combined to achieve

- ► Higher security
- ► Redundancy

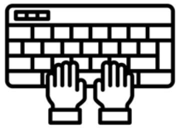
Q: What are the trade-offs for these two options?

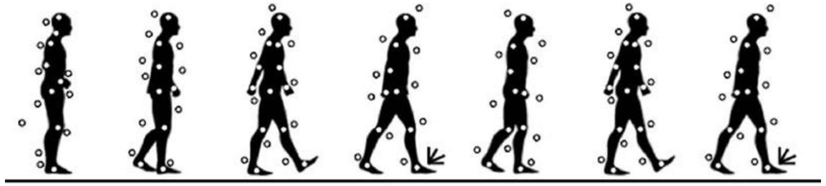
TYPES OF BIOMETRICS

- Based either on “static” property of person, e.g,
  - Fingerprint
  - Face
  - Palm-vein pattern
  - …



- Or on behaviour, e.g.,
  - Typing pattern
  - Walking gait
  - Eye saccades
  - …





TYPES OF TOKENS

- Dedicated security-devices, e.g,
  - Yubikey
  - OTP Token
  - Smart card
  - …

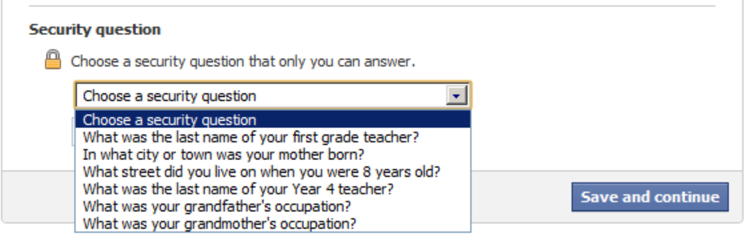
- Multi-purpose device also used for authentication, e.g.,
  - Smartphone
  - Wearables

TYPES OF COGNOMETRICS

- (Free) Recall-based, e.g, Text passwords

https://idp.scc.kit.edu

- Cued Recall-based, e.g., Security questions



Source: https://ux.stackexchange.com/questions/14016/what-is-the-benefit-of-fixed-security-questions

- Recognition-based, e.g., the graphical scheme Things
  - No systems in this category are widely deployed, but they have some very favourable properties

SINGLE SIGN-ON

- Can be used in conjunction with any authentication scheme
- Only one verifier for multiple accounts/services, e.g.,
  - SDU ID to access email, itslearning, VPN, etc.
  - Login on websites with Google/Facebook, etc.
- Advantages
  - User has to manage only one secret for the one verifier
  - Must not “trust” service with a password
- Disadvantages
  - Single point of failure
  - Privacy issues, must “trust” SSO service in this regard

WHAT ARE THE TYPICAL PROBLEMS WITH TEXT PASSWORDS?

COMMON ISSUES WITH PASSWORDS

- Secure passwords hard to generate and remember

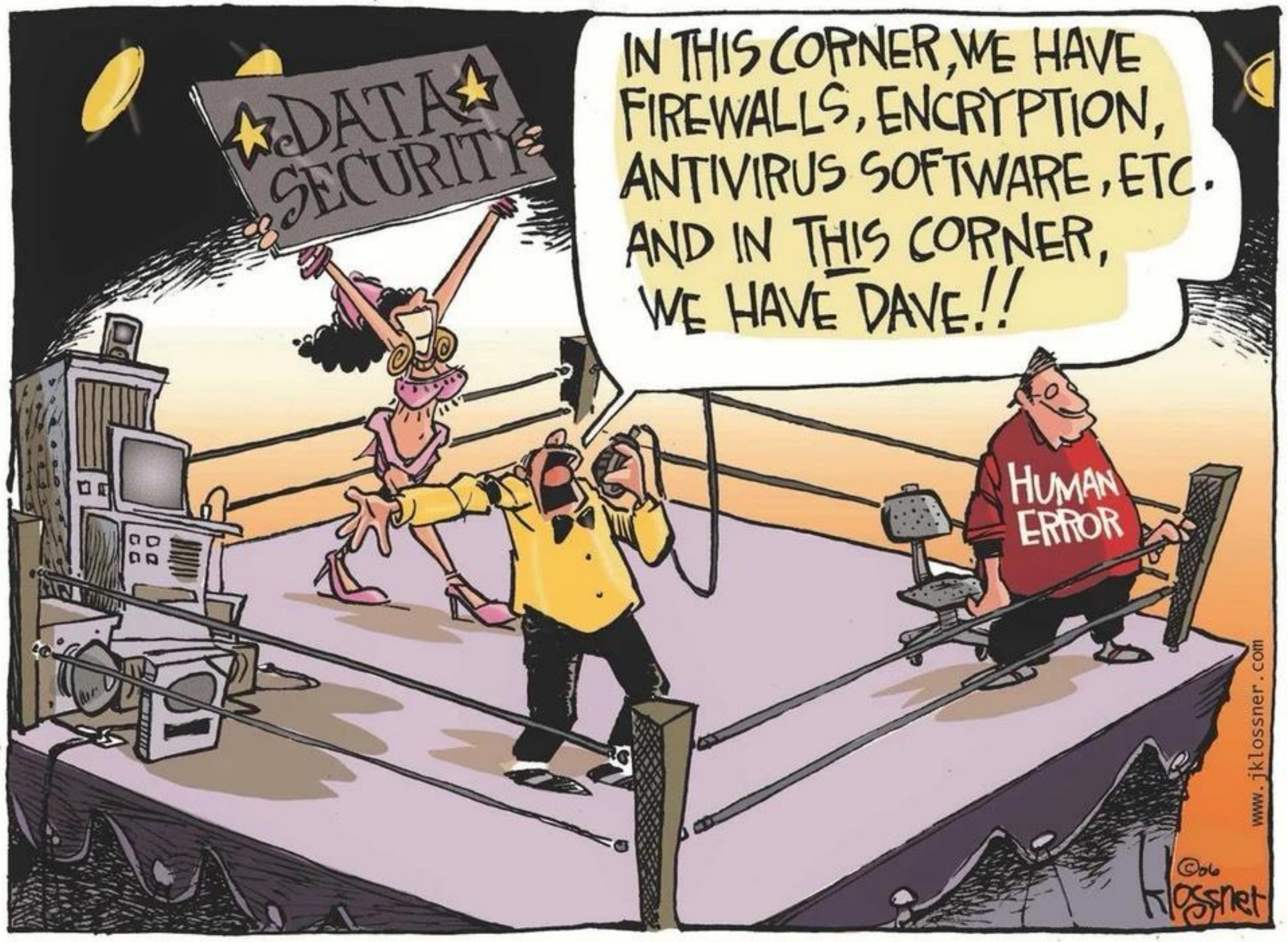
- People can reliably remember ~4 passwords
  - Weak passwords are chosen
  - Password reuse common

Oh no — pwned!
This password has been seen 37,359,195 times before
This password has previously appeared in a data breach and should never be used. If you've ever used it anywhere before, change it!

Search results for “123456” at haveibeenpwned.com

- Common guidelines/policies not helping
  - Passphrases are just as hard to remember
  - Password expiry weakens passwords and is discouraged
  - Any complexity rules beyond length and blocklist are discouraged

THE WEAKEST LINK



EVEN EXPERTS AT SECURITY FAIL WITH BEST PRACTICES

Cyber Tests Showed 'Nearly All' New Pentagon Weapons Vulnerable To Attack, GAO Says

October 9, 2018 · 5:14 PM ET

BILL CHAPPELL

Source: https://www.npr.org/2018/10/09/655880190/cyber-tests-showed-nearly-all-new-pentagon-weapons-vulnerable-to-attack-gao-says

- Even Pentagon has problems with password management

"…One test report indicated that the test team was able to guess an administrator password in nine seconds. Multiple weapon systems […] did not change the default password when the software was installed, which allowed test teams to look up the password on the Internet and gain administrator privileges for that software."

Source: https://www.gao.gov/products/GAO-19-128

AS DO TECH GIANTS, TELCOS, BROKERAGE FIRMS, ETC.

TECH / GOOGLE / SECURITY

Google stored some passwords in plain text for fourteen years / Only affects some G Suite customers

By Dieter Bohn
May 22, 2019 at 1:16 AM GMT+2

Facebook Stored Hundreds of Millions of User Passwords in Plain Text for Years

March 21, 2019 208 Comments

T-Mobile Austria is OK with Storing Passwords Partly in Clear Text

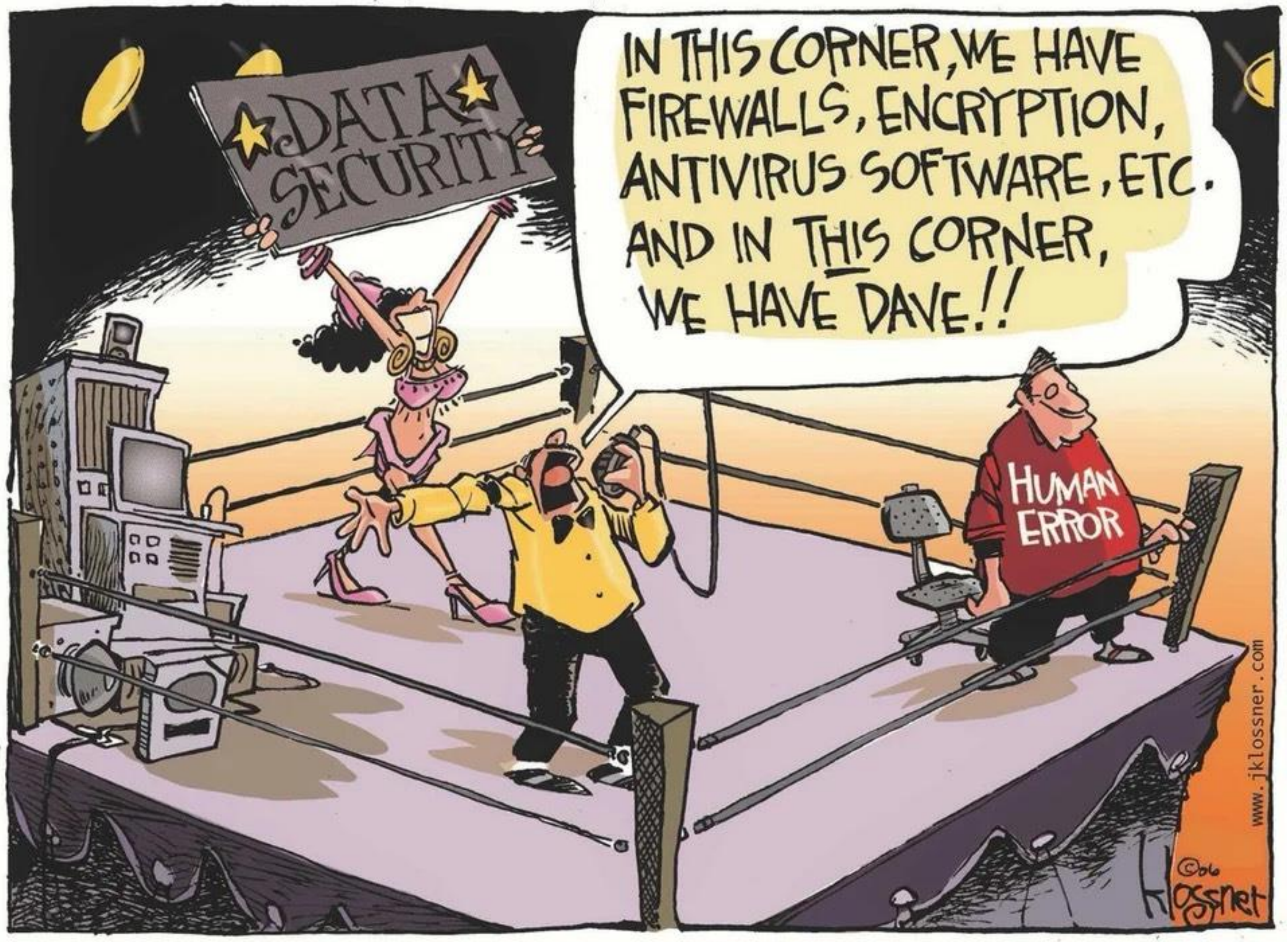
The admission came in Twitter exchange that left observers stunned. T-Mobile US hasn't followed its counterpart's example.

by Michael Kan Apr 07, 2018

Robinhood Brokerage Firm Alerts of Passwords Stored in Clear Text

By Lawrence Abrams July 24, 2019 05:57 PM 0

THE WEAKEST LINK



HOW TO COMPARE AUTHENTICATION SCHEMES?

A COMPARISON FRAMEWORK

- Widely accepted framework for rating authentication schemes (Bonneau et al. 2012)

- It defines three categories of metrics
  - **Security:** Authentication's primary purpose
  - **Usability:** Easy way for schemes to compromise security
  - **Deployability:** Will not be used if not easy to deploy in practice

- Each category has several metrics, that can fulfilled or some also quasi-fulfilled

- Comparison is not a trivial task of which one is the best
- Rather: Which one is the best for a certain application scenario

J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," IEEE Symposium on Security and Privacy, 2012, pp. 553–567.

A COMPARISON FRAMEWORK

- Widely accepted framework for rating authentication schemes (Bonneau et al. 2012)

			Usability								Deployability						Security										
Scheme	Described in section	Reference	*Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*

J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," IEEE Symposium on Security and Privacy, 2012, pp. 553–567.

SECURITY CRITERIA

- **Resilient to physical observation:** It is not possible to pass authentication after observation of password entry, quasi-resilience granted when possible only after 10-20 observations
- **Resilient to targeted impersonation:** It is not possible for someone with personal knowledge of the user to pass authentication
- **Resilient to throttled guessing:** An attacker who is rate-limited by the verifier cannot guess a significant number of secrets
- **Resilient to unthrottled guessing:** An attacker who is only limited by their own resources is not able to guess a significant number of secrets
- **Resilient to internal observation:** An attacker cannot pass authentication after observing the user's input from within the device, e.g., by using malware, quasi-resilience granted for [a] MFA where all factors need to be compromised, [b] authentication can be passed only after 10-20 observations

SECURITY CRITERIA (CONTINUED)

- **Resilient to leaks from other verifiers:** Verifiers can leak nothing that would kame an attack at another verifier easier
- **Resilient to phishing:** Simulating a valid verifier does not allow passing authentication (not including advanced man-in-the-middle attacks)
- **Resilient to theft:** Physical objects required to pass authentication can only be used by the genuine user, quasi-resilient if, e.g., PIN is required
- **No trusted third party:** Only user, scheme, and verifier involved in authentication
- **Requiring explicit consent:** The authentication process cannot be started without explicit consent by the user
- **Unlinkable:** Colluding verifiers cannot determine from the authenticator alone whether the same user is authenticating to both

DEPLOYABILITY CRITERIA

- **Accessible:** People that can use passwords can also use the scheme
- **Negligible cost per user:** The total cost per user of the scheme is negligible
- **Server-compatible:** Verifiers do not have to be changed if they support text passwords
- **Browser-compatible:** The scheme run in modern browsers without requiring add-ons

USABILITY CRITERIA

- **Memorywise effortless:** No secrets to remember/one secret for everything (quasi)
- **Scalable for users:** The burden on the user increases only negligibly with an increasing number of accounts
- **Nothing to carry:** No physical object needs to be carried around
- **Physically effortless:** Max physical effort is equivalent to pushing a button/ speaking (quasi)
- **Easy to learn:** It is easy to understand the scheme and recall that info later
- **Efficient to use:** Required time should be reasonably short (text pwd ~12 secs)
- **Infrequent errors:** Genuine users must be able to perform the required task easily without errors
- **Easy recovery from loss:** Recovery from loss must be itself usable

USABILITY CRITERIA

●= offers the benefit; ○= almost offers the benefit; *no circle* = does not offer the benefit.
|||= better than passwords; ≡= worse than passwords; *no background pattern* = no change.

			Usability								Deployability						Security										
Scheme	Described in section	Reference	*Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*
Web passwords	III	[13]			●		●	●	○	●	●	●	●	●	●	●		○						●	●	●	●

Difficult to remember, especially multiple

Super easy to implement and compatible

Quite bad security-wise

USABILITY CRITERIA

●= offers the benefit; ○= almost offers the benefit; *no circle* = does not offer the benefit.
|||= better than passwords; ≡= worse than passwords; *no background pattern* = no change.

Scheme	Described in section	Reference	Usability: *Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	Deployability: *Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*	Security: *Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*
Web passwords	III	[13]			●		●	●	○	●	●	●	●	●	●	●		○						●	●	●	●
OpenID	IV-C	[27]	○	●	●	○	○	●	●	●	●	●		●	●	●	○	○	○	○		●		●		●	
Facebook Connect		[44]	○	●	●	○	●	●	●	●	●	●		●	●		○	○	○	○		●		●			

Better in terms of usability, except for complexity to learn

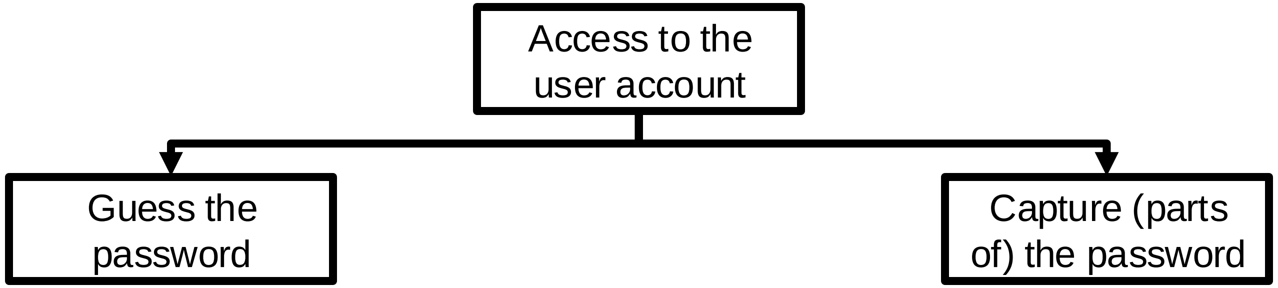
Strictly worse in terms of deployability

Better against several attacks, but might need to trust 3rd party, implicity & privacy tade-off

HOW ARE AUTHENTICATION SECRETS AND ACCOUNTS OF USERS ATTACKED?

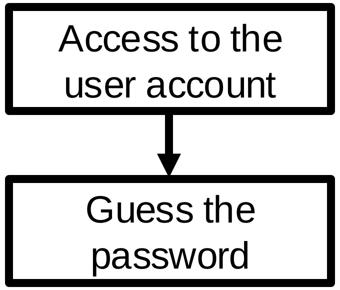
A THREAT MODELLING VIEW ON ATTACKS

- We will build up an attack tree with the attacks
- Our root node will be the goal, and we will branch from there
- We can categorize attacks according to how the password is obtained



- We first look at the guessing sub-tree

GUESSING BASICS – PASSWORD SPACE



Example
PIN of length 4 or 5
- $|A| = 10$
- $L = \{4,5\}$

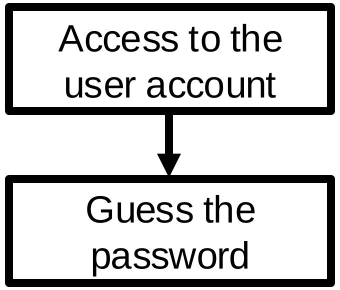
$$|P| = \sum_{l \in \{4,5\}} 10^l = 10^4 + 10^5 = 110000$$

- Difficulty of guessing passwords is ultimately decided by the number of passwords that need to be tested
- Totality of passwords to be tested is the password space P
- There are two password spaces
- **Theoretical password space:** All passwords that can be chosen in a password scheme. Size is determined by:
  - A = Alphabet (all elements that can be used in a secret)
  - L = Allowed lengths of the secrets

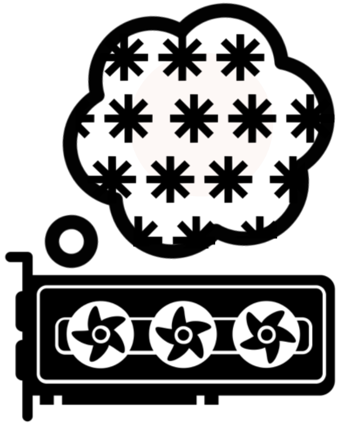
$$|P| = \sum_{l \in L} |A|^l$$

- **Practical password space:** All passwords that are typically chosen
  - Depends on user choice
  - Size must be empirically determined

GUESSING BASICS – BRUTE-FORCE GUESSING



- All possible password candidates are tested
  - Theoretically every password can be guessed
  - Attack on the theoretical password space



- Computational resources increase with strength of the password
  - With secure hash not viable after length ~10-12

Clearing up a common misconception

Short passwords can be as complex as possible, but will still be insecure. Password with 8 characters or less will be easily broken using today's hardware, even if they are composed using different character classes.

GUESSING BASICS – PASSWORD SPACE

Access to the user account

↓

Guess the password

- Attacker uses a wordlist for the attack, composed of e.g.,
  - Common passwords & passwords from leaks
  - Dictionary from different languages
  - Numbers with meaning such as dates
  - Entries specific for attack

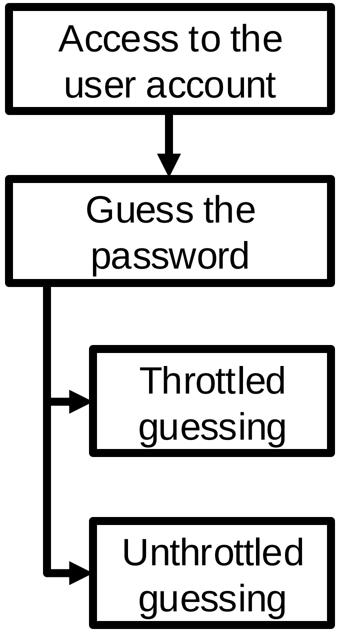
Palin E-Mail Hacker Says It Was Easy

As detailed in the postings, the Palin hack didn't require any real skill. Instead, the hacker simply reset Palin's password using her birthdate, ZIP code and information about where she met her spouse – the security question on her Yahoo account, which was answered (Wasilla High) by a simple Google search.

- Special case: Exploits knowledge about victim or device to create wordlist, e.g., default passwords for devices

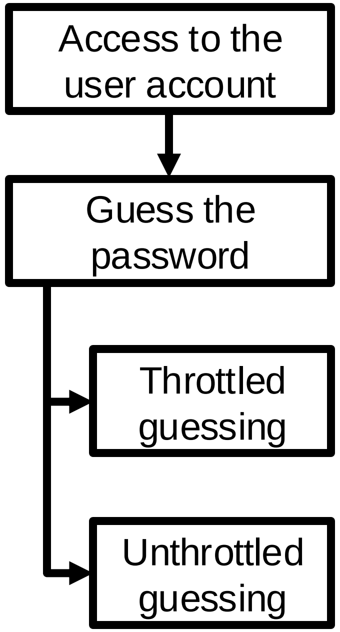
- Each entry in the wordlist is changed according to mangling rules, e.g., append !
  - Predictable changes to "make the password more secure" such as capitalising the first letter are know by attackers

GUESSING BASICS – THROTTLED VS UNTHROTTLED



- Throttled guessing
  - Needs to use verifier for the attack, e.g., login form
  - Rate-limited by verifier
  - Special named cases
  - Trawling attack: Untargeted attack using most common passwords at an arbitrary service
  - Password stuffing: Semi-targeted attack that uses credentials leaked in a data breach
- Unthrottled guessing
  - More powerful guessing attack, but requires compromising system and stealing password file
  - Limited only by the resources of the attacker
  - Attacker hashes password candidates and compares to stolen password data

GUESSING BASICS – THROTTLED VS UNTHROTTLED



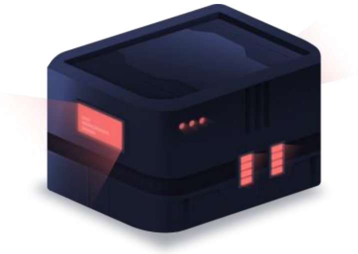
- Guessing attacks in practice with hashcat
- Hashcat
  - Password recovery tool that leverages GPUs to accelerate attacks
  - “the world’s fastest and most advanced password recovery tool”
  - Available for Windows, Linux, macOS (with limitations)
- Basic syntax

```
hashcat -a [attack mode] -m [hash type] [attack inputs]
```

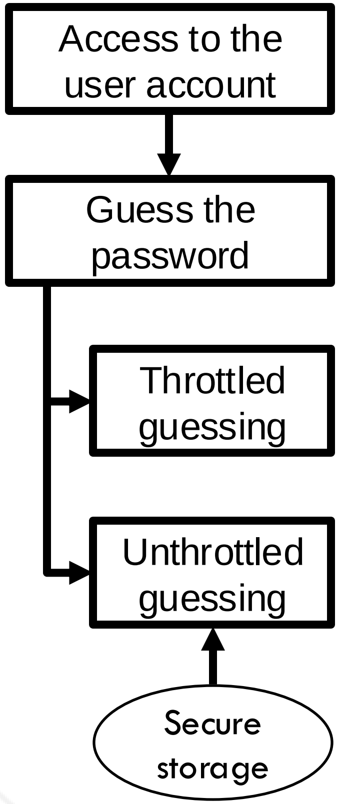
- Brute-force (mask attack)
- Wordlist-based (dictionary)
- Hybrid

- How password is stored

- Password file
- Wordlist
- Mangling rules

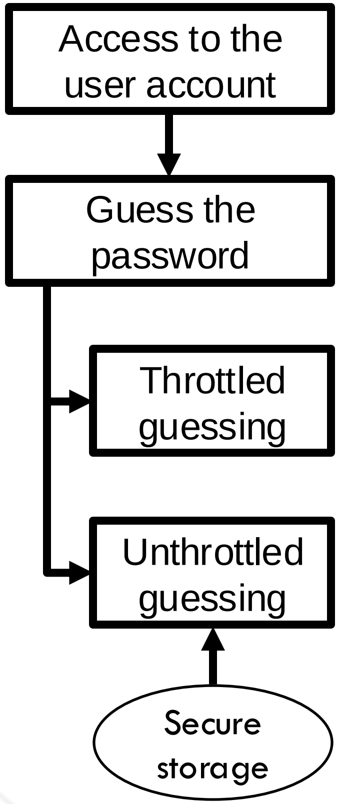


GUESSING ATTACKS – SECURE PASSWORD STORAGE

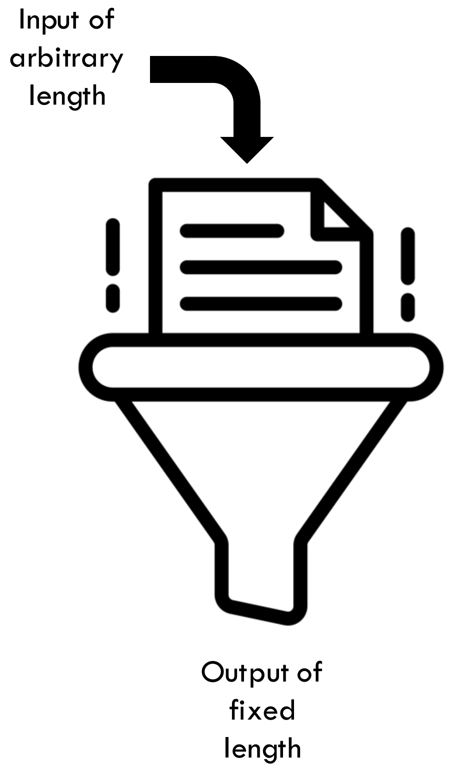


- Only helps against unthrottled guessing
  - Throttled guessing uses the verifier
  - No access to password data
- Secure special purpose hashing function

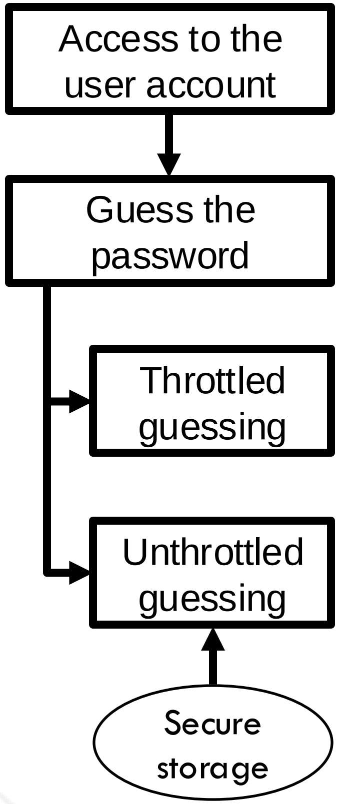
GUESSING ATTACKS – CRYPTOGRAPHIC HASHING FUNCTION



- The goal of all hash functions is to derive values of fixed length from inputs of arbitrary length
- General properties
  - One-way function
  - Collision-resistance
- Efficiency
  - When used for password storage
  - Efficiency is not really desirable because it makes attacks easier

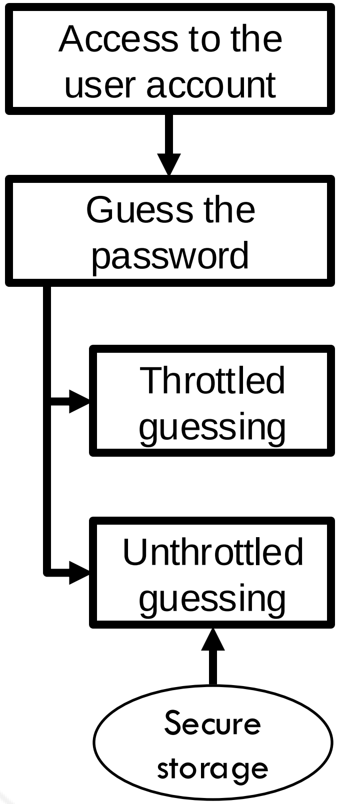


GUESSING ATTACKS – SECURE PASSWORD STORAGE



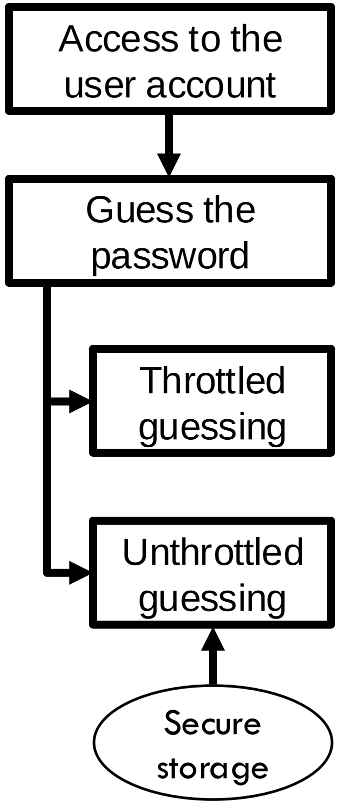
- Only helps against unthrottled guessing
  - Throttled guessing uses the verifier
  - No access to password data
- Secure special purpose hashing functions
  - Argon2
    - Modern memory-hard hashing function
    - Should preferably be used
  - PBKDF2
    - Mandated by some certification standards
    - If used should adhere to OWASP recommendations
      - 600.000 iterations for HMAC-SHA256
      - 210.000 iterations for HMAC-SHA512

GUESSING ATTACKS – SECURE PASSWORD STORAGE



- **Salt:** Random value stored alongside password
  - Individual to each user/password, and included in hashing
  - Otherwise same password would result in same hash
    - Would allow precomputing hashes of popular passwords
    - Computation once, attack as often as one likes
    - Note: lists of pre-computed hashes called rainbow tables
- **Pepper:** Special secret value unique for one system included in hashing
  - Stored in other place than passwords, e.g., in source of webapp
  - If password data is stolen, if pepper not included, no way to guess
- Only helps against unthrottled guessing
  - Throttled guessing uses the verifier
  - No access to password data

GUESSING ATTACKS – SECURE PASSWORD STORAGE



- How does secure password storage then look like?

config.py

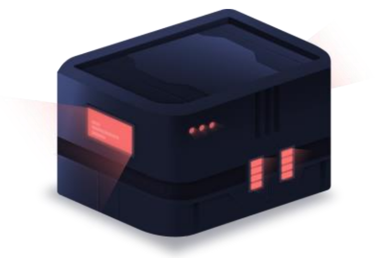
```
PEPPER = 'b04ab0b2d1e69cfa54f49f34…
```

pwd-storage.py

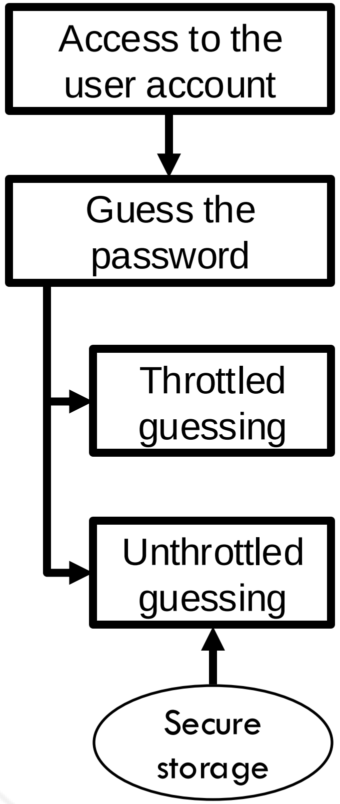
```
from argon2 import PasswordHasher

def storeHash(userinput)
  ph = PasswordHasher()
  hash = ph.hash(userinput + PEPPER)

  # Then write to DB
```

Using suitable libraries makes life easier.
Already includes defaults for:

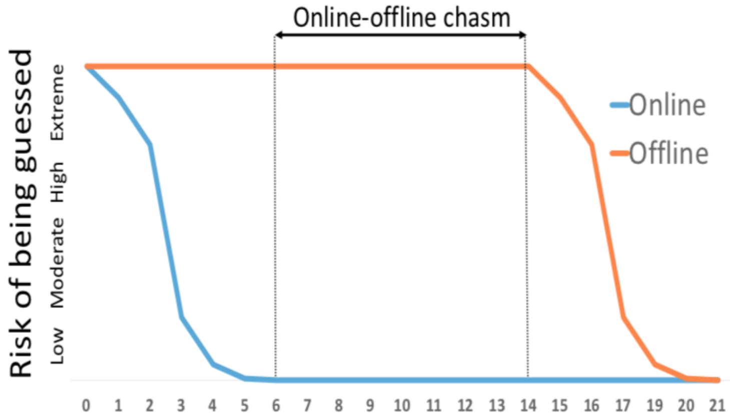
- Salt length
- Cost factors



GUESSING ATTACKS – PASSWORD COMPOSITION



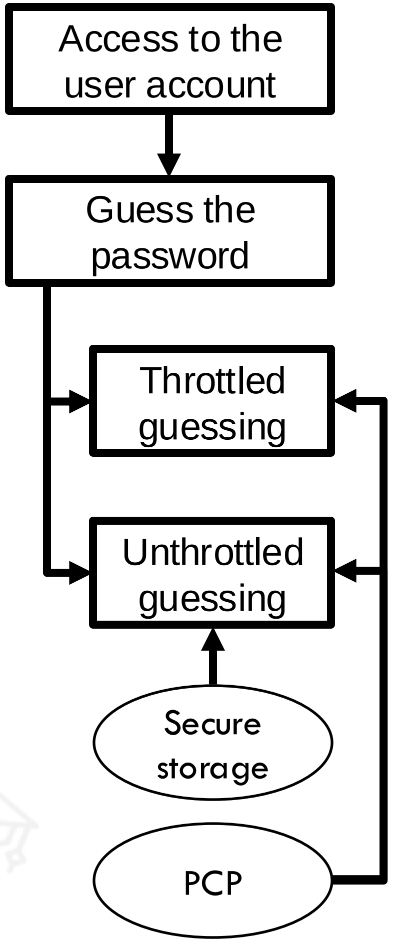
- Can enforce a password composition policy (PCP)
- Important to be mindful about the desired security level
- PCP in “Online-offline chasm” only burdens user but does not help in security



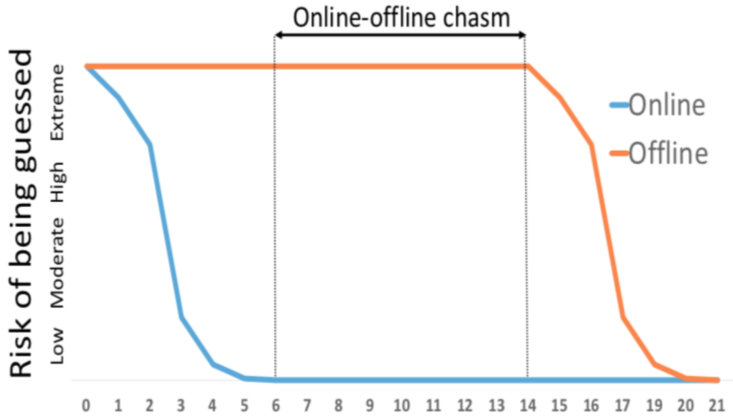
Note
Throttled = Online
Unthrottled = Offline
Are used synonymously

- Current best practice is to only enforce a length requirement and no other composition rules

GUESSING ATTACKS – SECURE PASSWORD STORAGE



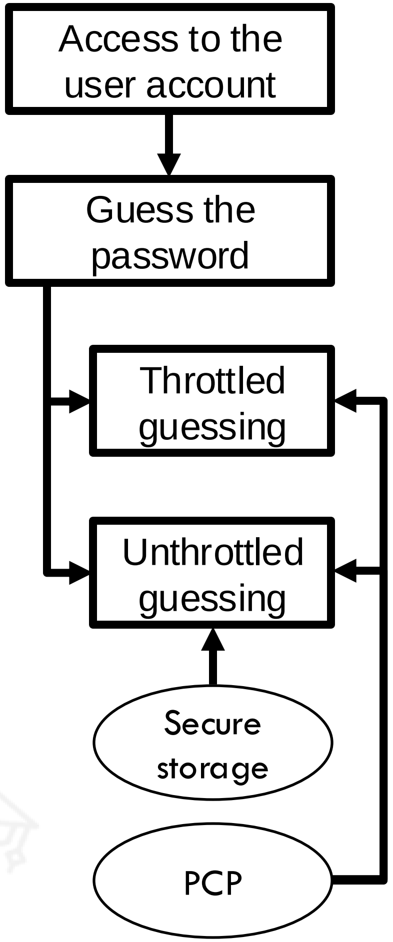
- Can enforce a password composition policy (PCP)
- Important to be mindful about the desired security level
- PCP in “Online-offline chasm” only burdens user but does not help in security



Note
Throttled = Online
Unthrottled = Offline
Are used synonymously

- Current best practice is to only enforce a length requirement and no other composition rules

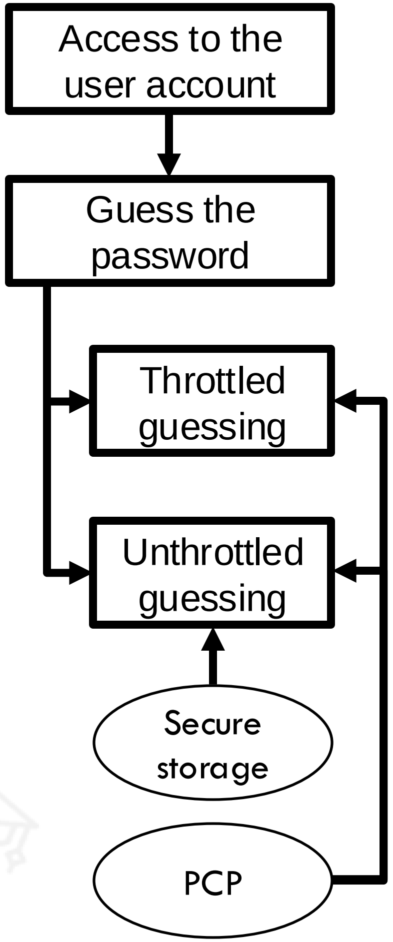
GUESSING ATTACKS – SECURE PASSWORD STORAGE



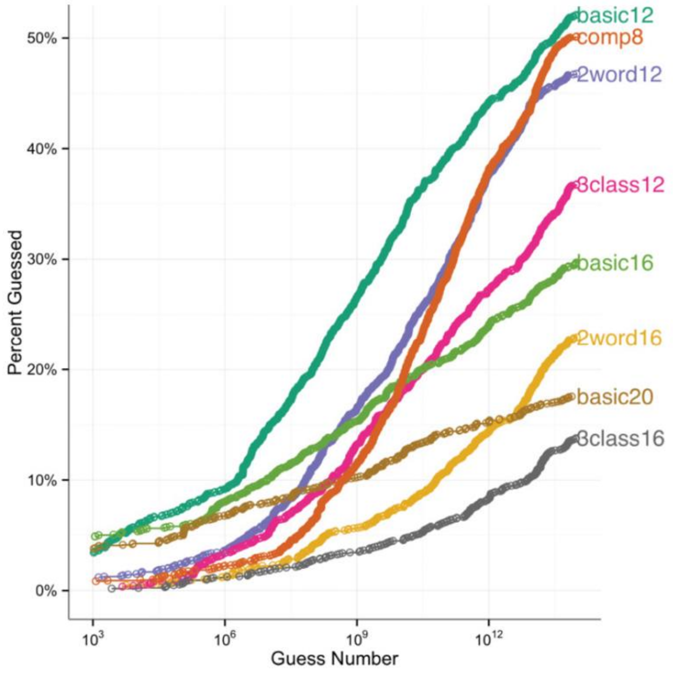
- Complexity rules are confusing (Choong et al. 2016)
  - N=60
  - Character-selection and compliance-checking tasks

"[Our] participants were extremely confused by the variety of terms for 'special character.' Seemingly small changes in language have large, observable impacts on users' understanding of password rules. Language in password requirements must be carefully constructed to ensure that users fully comprehend the allowable character space."

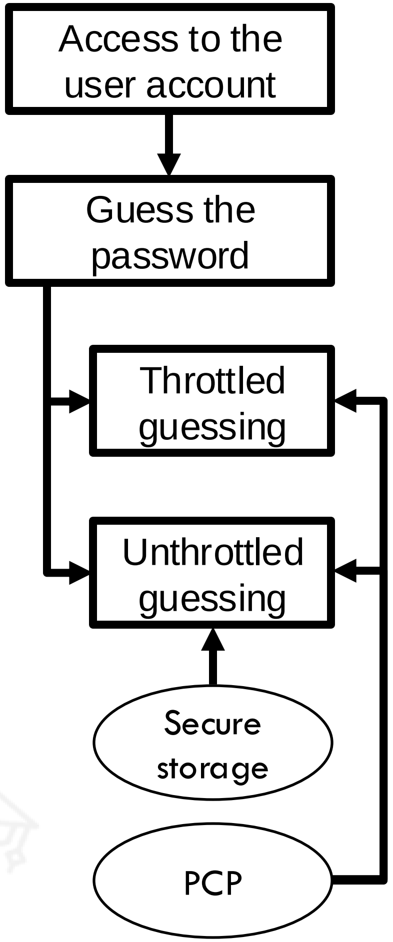
GUESSING ATTACKS – PROBLEMS OF COMPLEXITY RULES

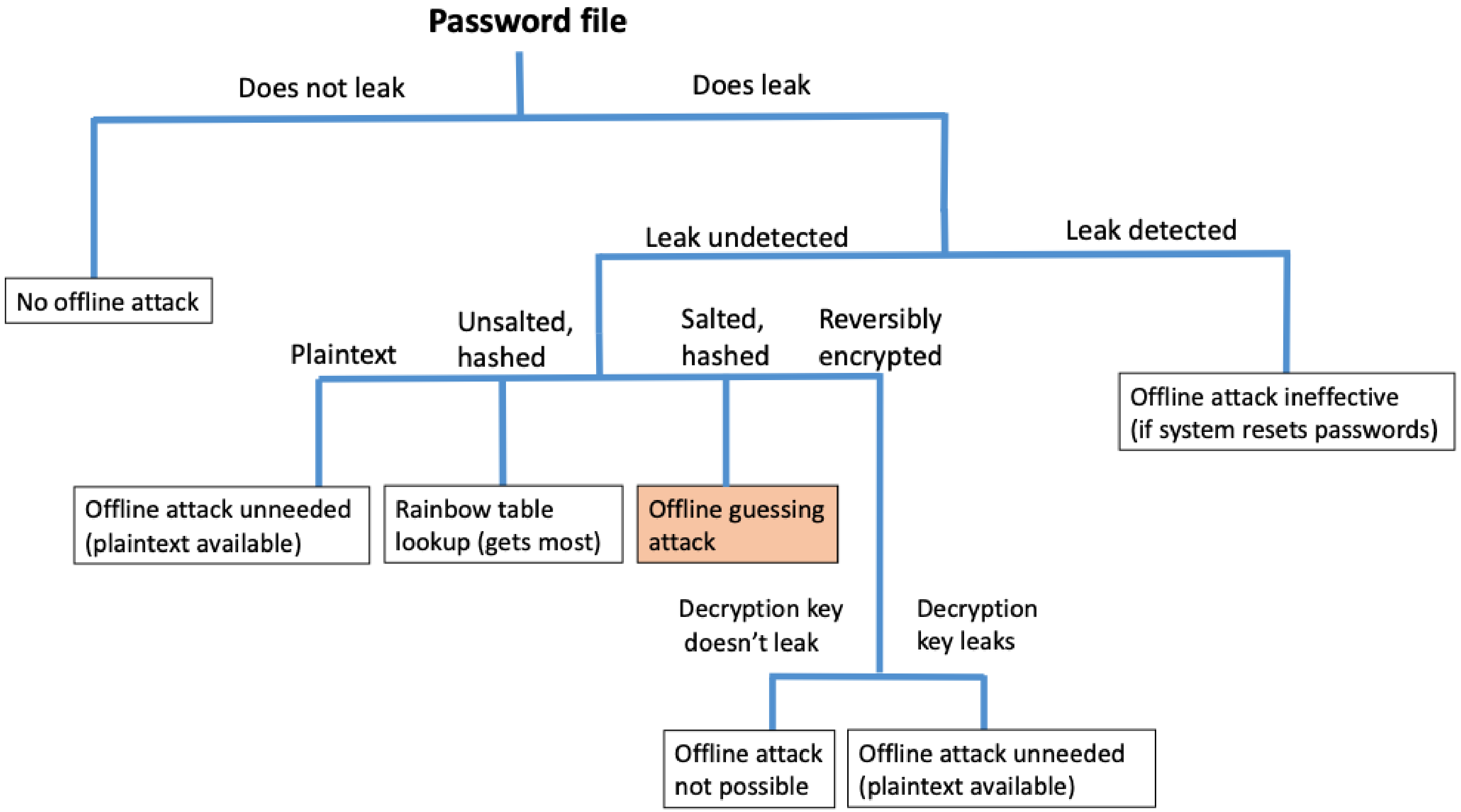


- Complexity rules do not have the impact one would imagine (Shay et al. 2016)
  - Two online studies, N=20000
  - Examine creation under 15 password policies
- “Our research has shown that there are other policies that are more usable and more secure. We found three policies — 2class12, 3class12, and 2word16—that we can directly recommend over comp8.”
- But also use blocklists or just length is problematic

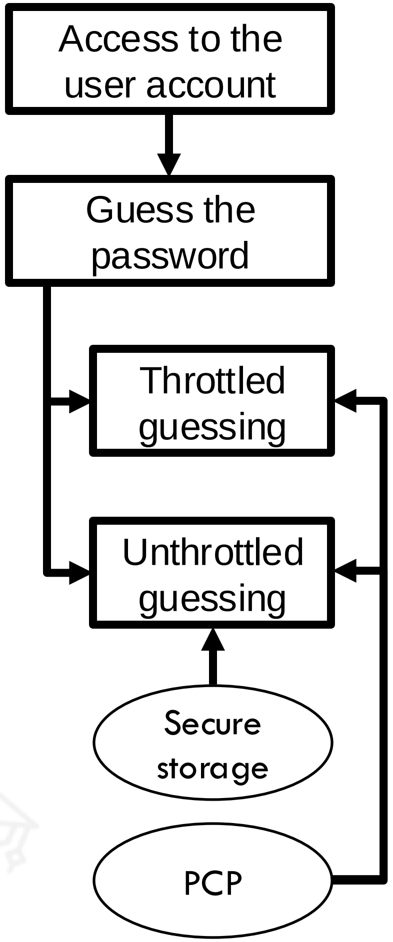


GUESSING ATTACKS – PROBLEMS OF COMPLEXITY RULES



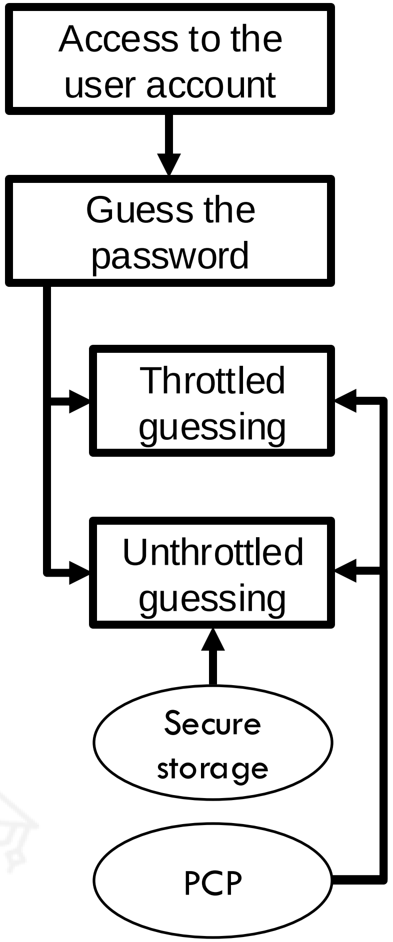


GUESSING ATTACKS – PROBLEMS OF PASSWORD EXPIRY



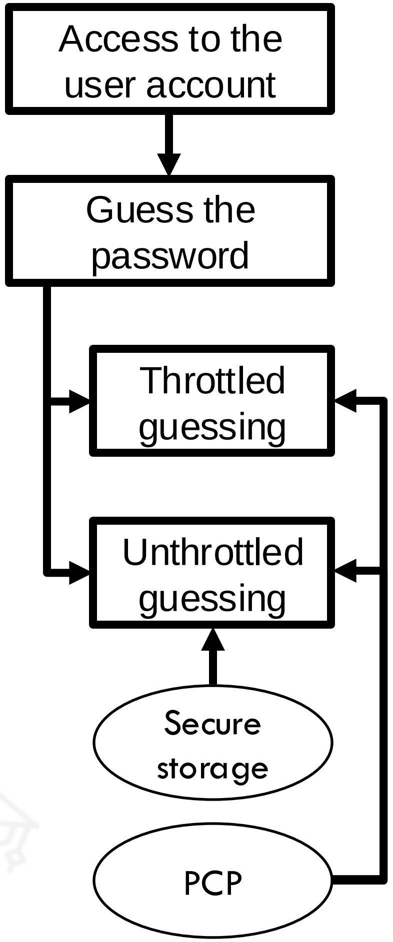
- Password expiry is tradeoff between
- The **security gained** by giving the attacker a "moving target"
- The **security lost** by forcing people to use coping strategies

GUESSING ATTACKS – PROBLEMS OF PASSWORD EXPIRY



- How much security is gained? (Chiasson & van Oorschot 2015)
  - Prevents attacker with access to use the same password for continued access
  - Does not prevent backdoors, persistent malware, etc.
- “In sum, these security-specific observations and the results in Sect. 3 suggest the security **benefit of password aging policies are at best partial and minor**.”
- ”the maximum advantage that a defender can hope to gain by a policy-driven password change is a reduction [in the expectation of attack success] from 1.0 [...] to a **probability [...] in no cases any lower than 0.632**”

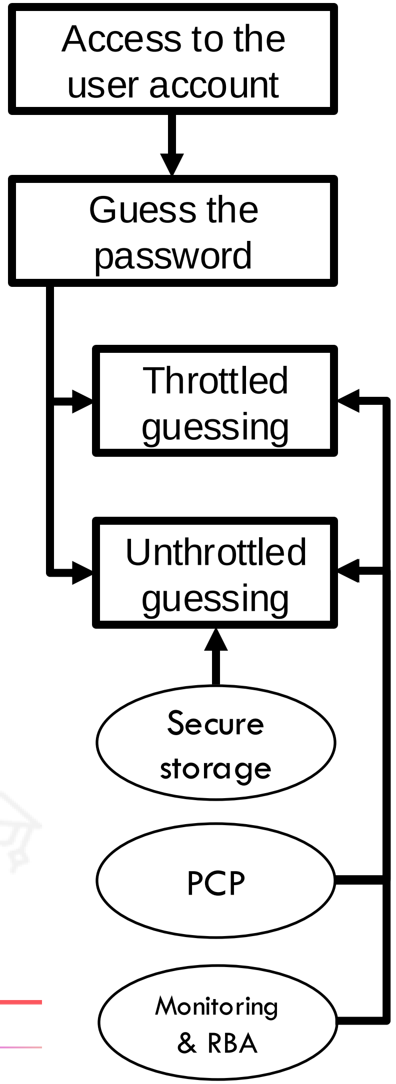
GUESSING ATTACKS – PROBLEMS OF PASSWORD EXPIRY



- How much security is lost?
  - Users required to change their passwords frequently produce less passwords in the first place and disclose them more often (Adams & Sasse 1999)
  - Users annoyed with password policies choose weaker passwords (Mazurek et al. 2013; Inglesant & Sasse 2010)

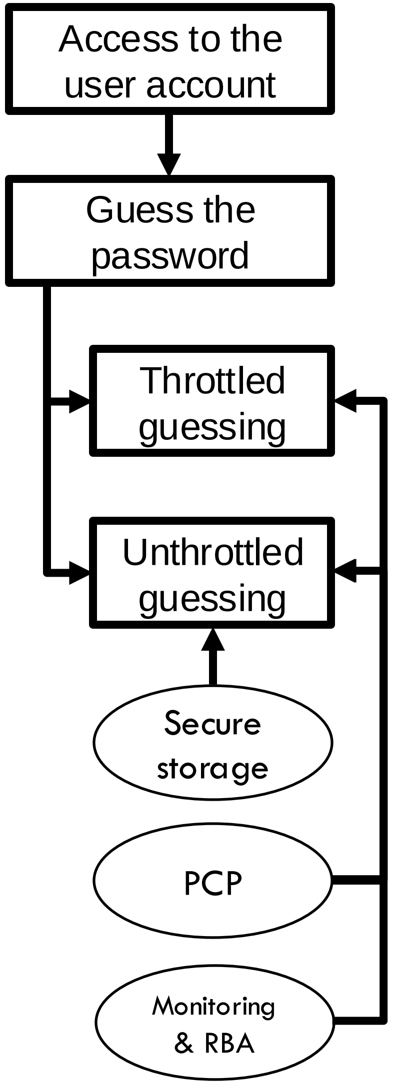
- Experiment with users previous passwords (Zhang et al. 2010)
  - "we can break 17% of accounts on average in an online attack, with fewer than 5 online guesses in expectation"
  - "we can break future passwords from past ones using the same class of transforms in 63% of accounts on average in an offline attack"

GUESSING ATTACKS – MONITORING



- Especially in business environment, systems should be monitored for unusual activity
- Intrusion Detection Systems
  - Network activity
  - Process activity
  - Access patterns
  - …
- Honey pots
  - Services only for security purposes
    - If accessed ⇢ raise alarm
  - Specific user input
    - If detected ⇢ raise alarm

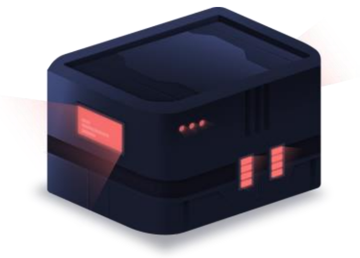
GUESSING ATTACKS – MONITORING



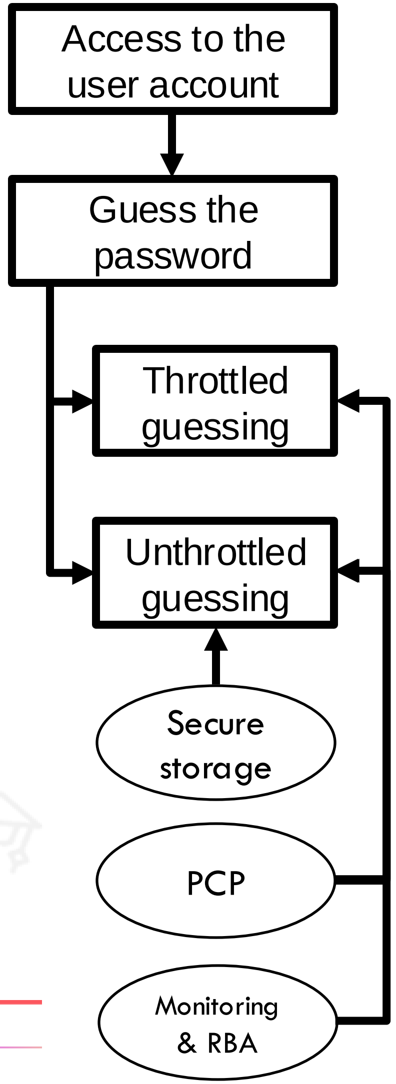
- Potential implementation "honey passwords" (Juels & Rivest 2013)

pwd-verify.py

```
from argon2 import PasswordHasher

honeyPasswords = […]

def checkPwd(userinput)
    ph = PasswordHasher()
    if userinput in honeyPasswords:
        raiseAlert()
        return False
     else:
        # Get hash from DB
        return ph.verify(hash, userinput+PEPPER)
```

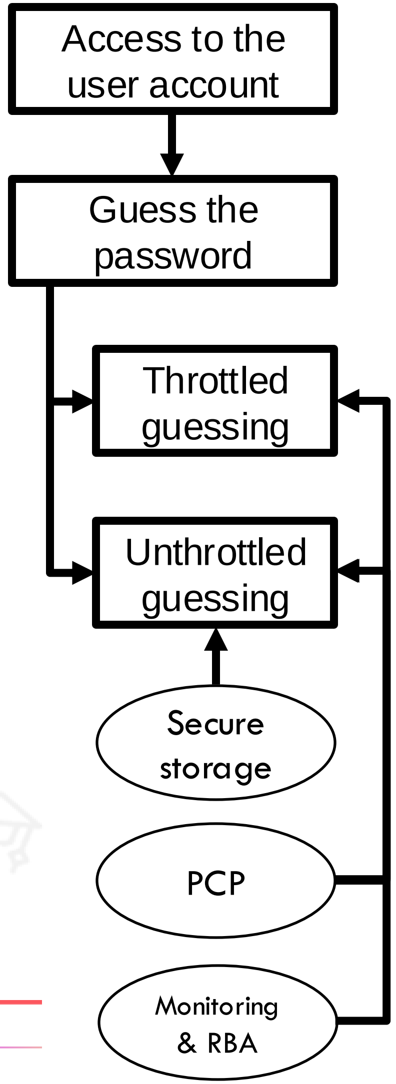


GUESSING ATTACKS – MONITORING

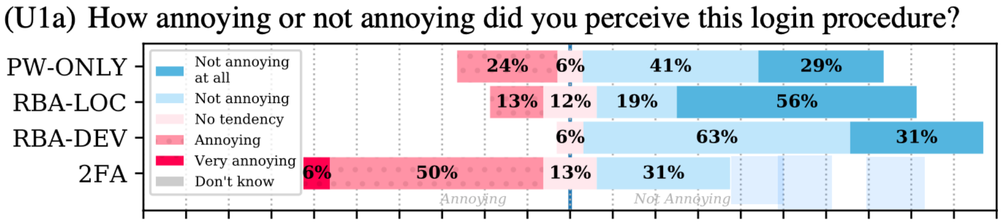


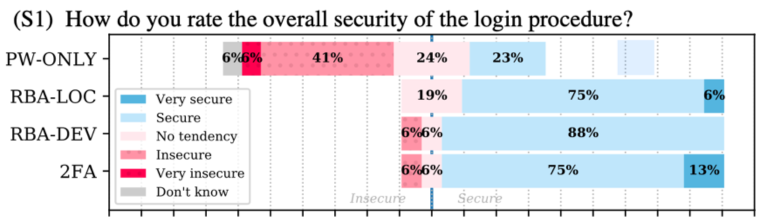
- Risk-based authentication (RBA)
  - Require additional factor in certain circumstances
  - Based on monitoring of clients and login attempts
  - Includes adaptive rate-limiting
- Can help against throttled and unthrottled guessing
- Can recognize break-ins and invalidate password data if necessary
  - Only in this case must passwords be changed
  - Regular password changes have been proven to burden users but not help security much

GUESSING ATTACKS – MONITORING

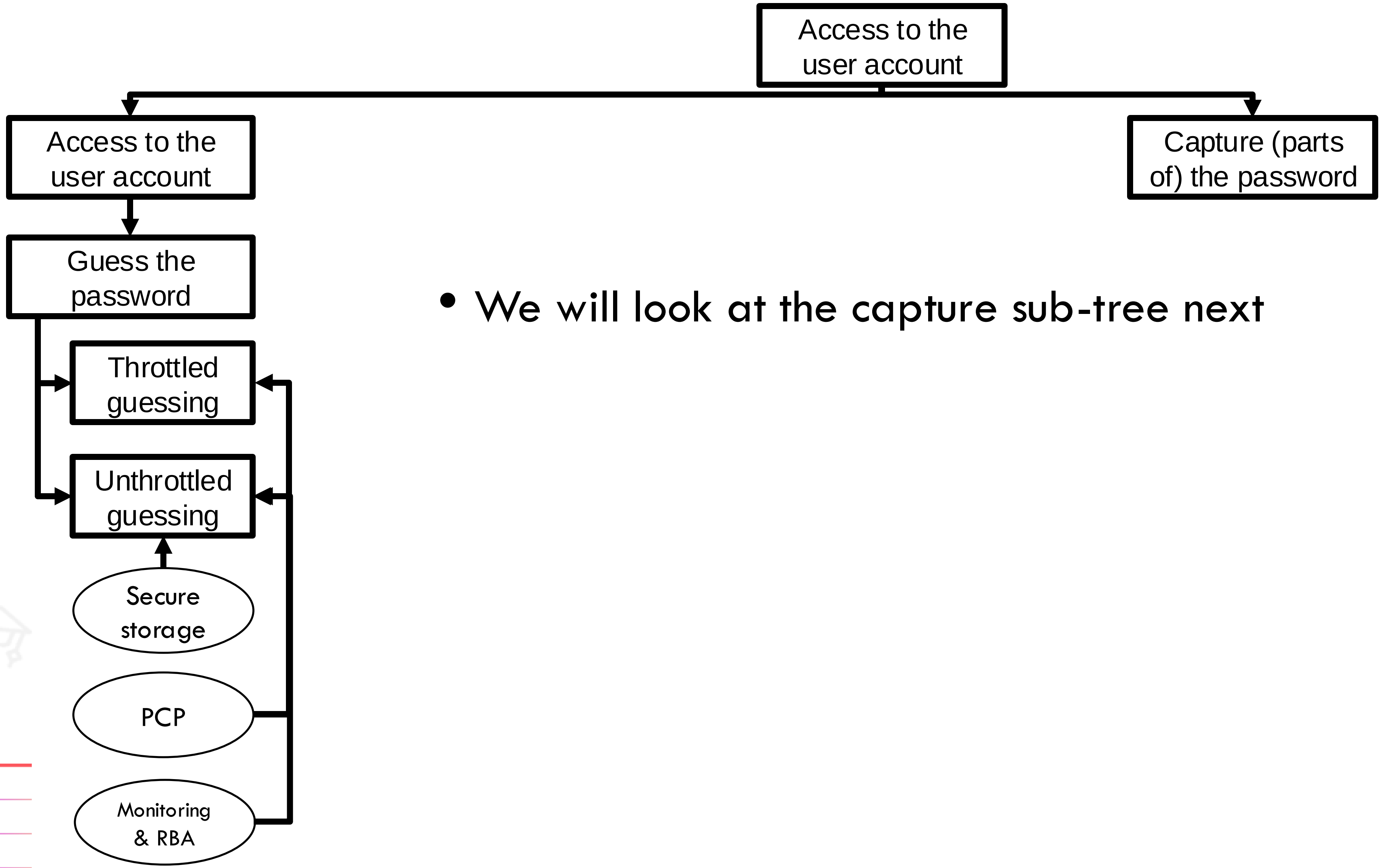


- RBA increases usability: burden adapted to context
- Preferable over 2FA (Wiefling et al. 2020)
  - N=60, four conditions with 2FA, 2xRBA, password only
  - No sig. difference between password only and other groups for satisfaction
- But
  - RBA sig less annoying than 2FA and
  - Sig. differences for security perception



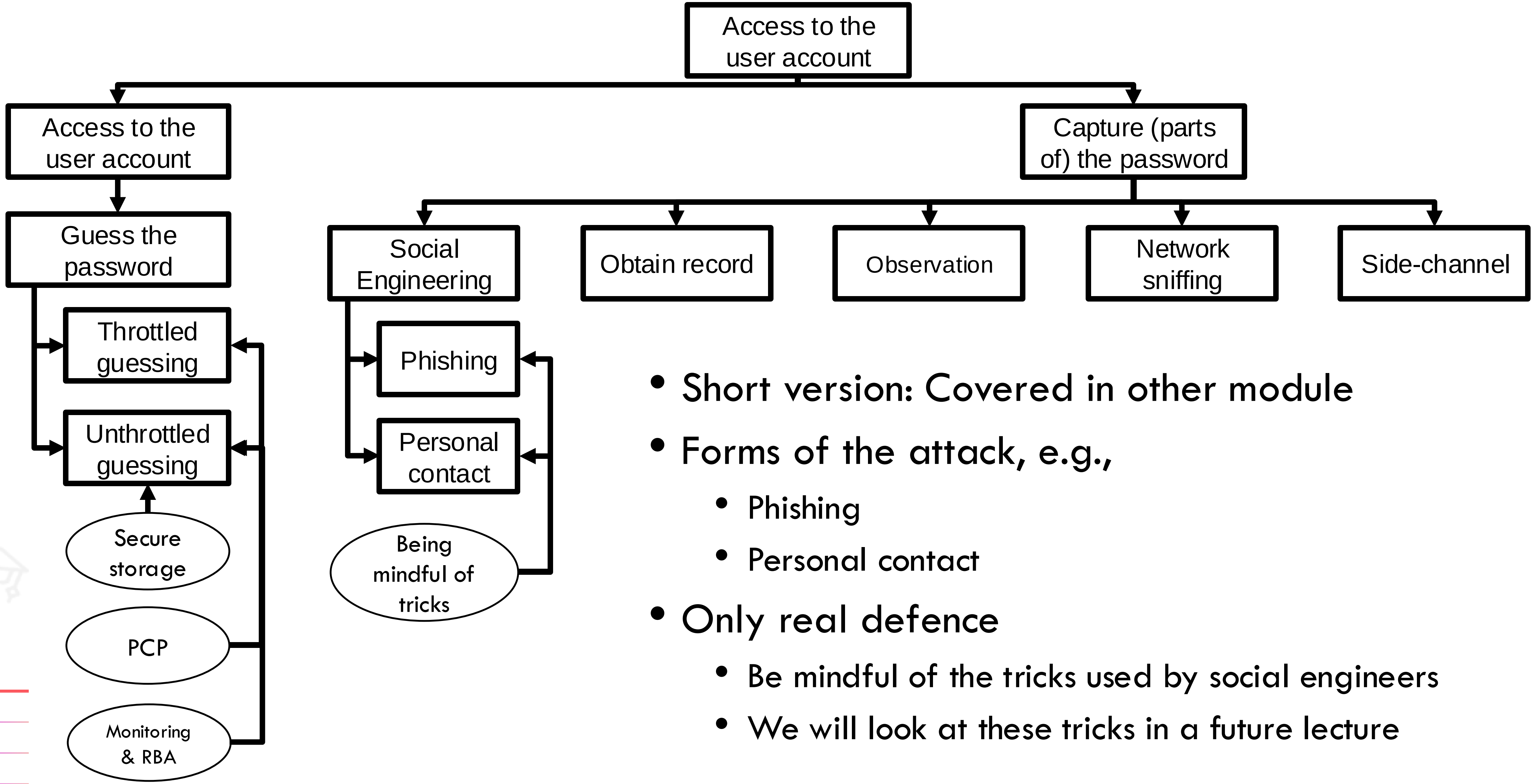


THE ATTACK TREE SO FAR



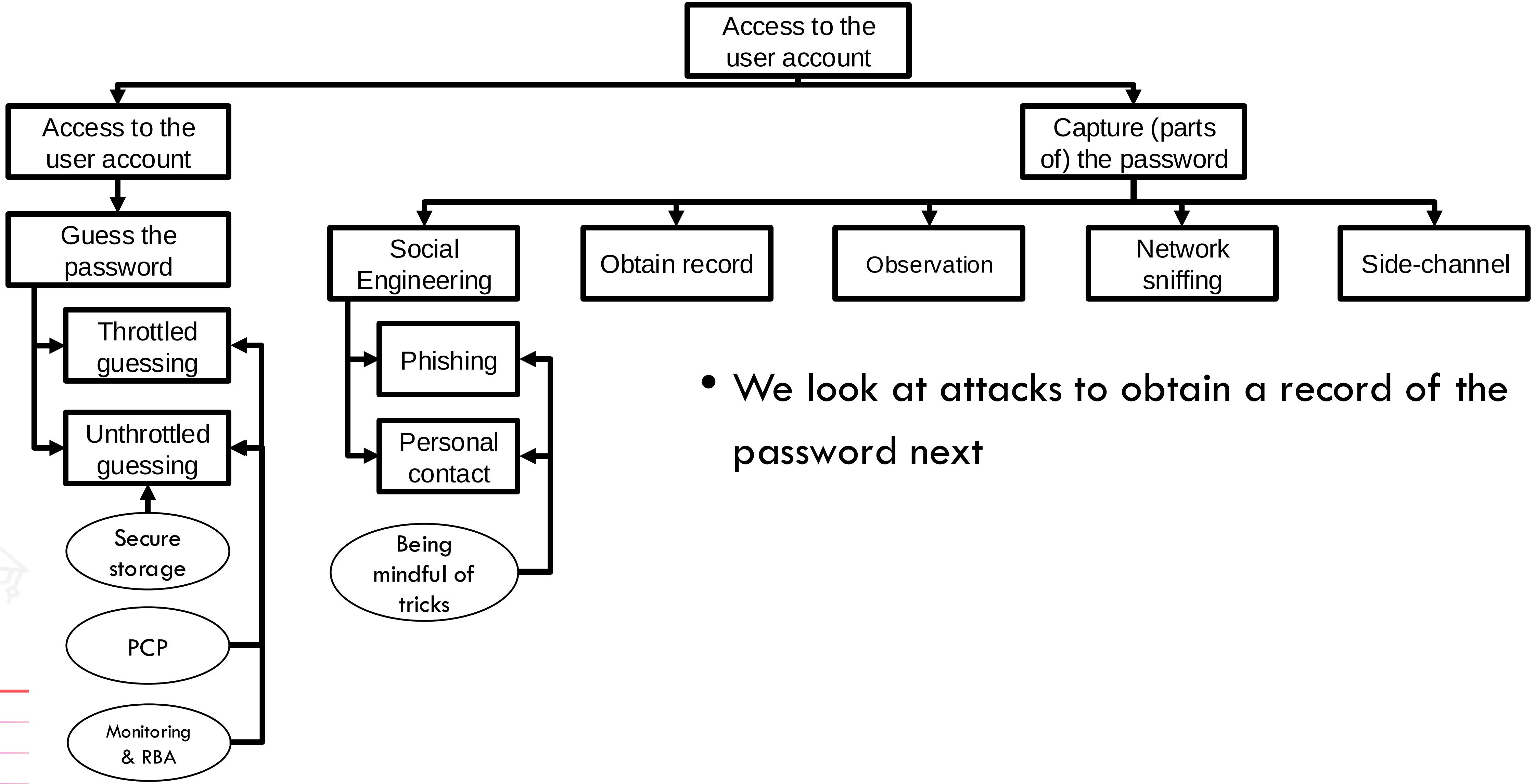
- We will look at the capture sub-tree next

SOCIAL ENGINEERING ATTACKS



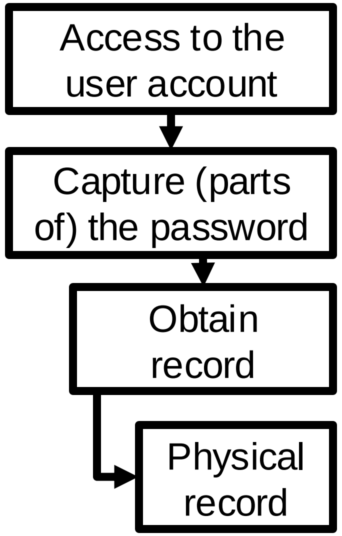
- Short version: Covered in other module
- Forms of the attack, e.g.,
  - Phishing
  - Personal contact
- Only real defence
  - Be mindful of the tricks used by social engineers
  - We will look at these tricks in a future lecture

THE ATTACK TREE SO FAR



- We look at attacks to obtain a record of the password next

OBTAINING A RECORD – PHYSICAL RECORD



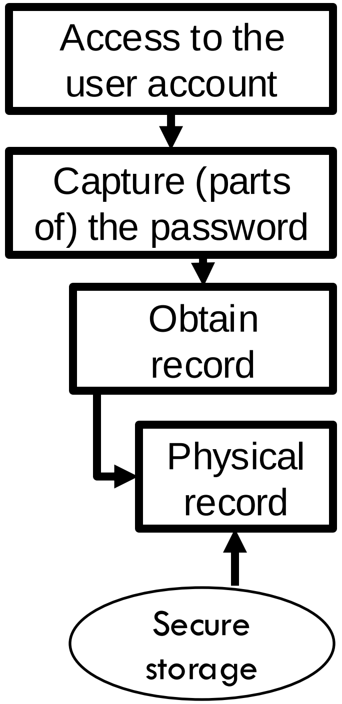
- The attacker exploits physical records
- Can pretend to be maintenance staff or similar
- Or just happens by accident

OBTAINING A RECORD – PHYSICAL RECORD

Access to the user account
→ Capture (parts of) the password
→ Obtain record
→ Physical record

- The attacker exploits physical records
- Can pretend to be maintenance staff or similar
- Or just happens by accident

OBTAINING A RECORD – PHYSICAL RECORD

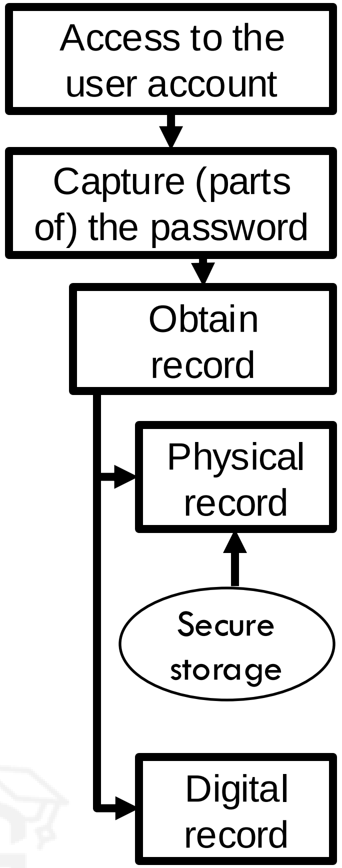


- The attacker exploits physical records
- Can pretend to be maintenance staff or similar
- Or just happens by accident

- Defences
  - Physical copies need to be safely stored
  - Might also be copy you did not create, e.g., letter with PIN

- Overall, securely stored physical copies are better than reused passwords
  - Reused password online, stored copy in you home

OBTAINING A RECORD – DIGITAL RECORD



- Unencrypted digital storage by users is basically the same as a Post-it note on the monitor
- Worst case
  - Unencrypted file
  - Synced vis cloud
  - E.g., text file via Dropbox, OneNote notebook, etc.
- Might also become problem later on
  - Device or harddrive sold and not cleaned

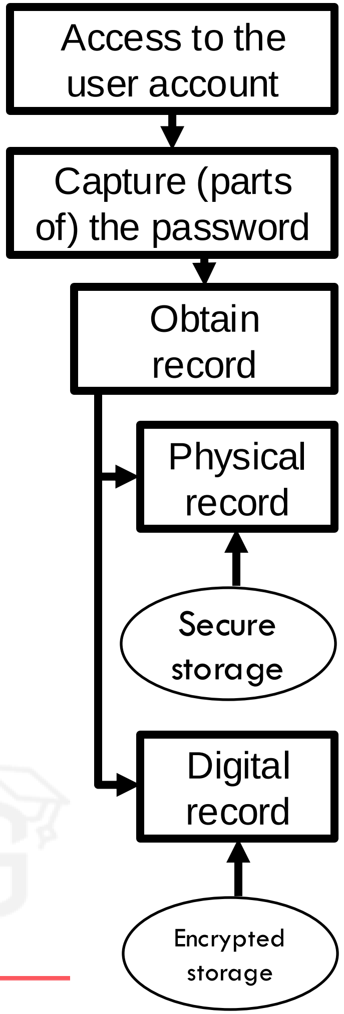
Survey: 40% of hard drives bought on eBay hold personal, corporate data

Buyers found data on everything from corporate spreadsheets to e-mails and photos

By Lucas Mearian

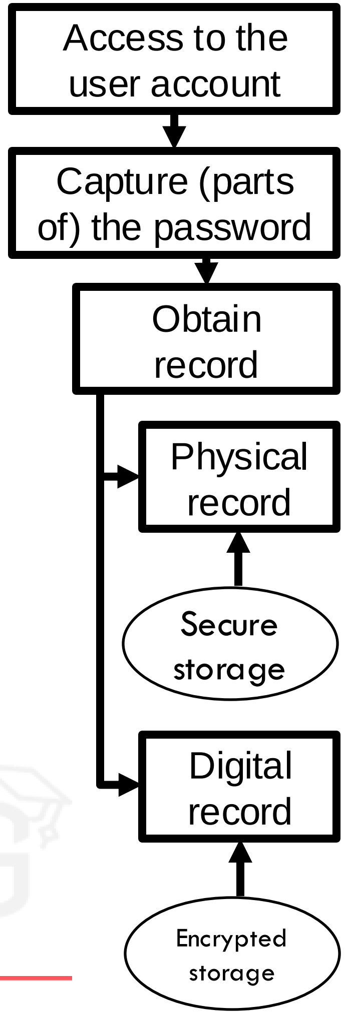
Senior Reporter, Computerworld | FEB 10, 2009 12:00 PM PST

OBTAINING A RECORD – DIGITAL RECORD



- Unencrypted digital storage by users is basically the same as a PIN note on the monitor
- Defences
  - Store digital records only encrypted
  - Easiest way to do so: Password manager
    - Also provides
      - Strong individual passwords
      - Resilience to phishing attacks when used in browser
      - Resilience to shoulder-surfing attacks
      - Encrypted sync accross devices
  - If a password was shared (e.g., via messagers) change it
  - Passwords send via email need to be changed asap

HOW DO PASSWORD MANAGERS PERFORM?



●= offers the benefit; ○= almost offers the benefit; *no circle* = does not offer the benefit.
= better than passwords; = worse than passwords; *no background pattern* = no change.

Usability

Scheme	Described in section	Reference	*Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*
Web passwords	III	[13]			●		●	●	○	●
Firefox	IV-A	[22]	○	●	○	○	●	●	●	
LastPass		[42]	○	●	○	○	●	●	●	○

Deployability

Scheme	Described in section	Reference	*Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*
Web passwords	III	[13]	●	●	●	●	●	●
Firefox	IV-A	[22]	●	●	●		●	●
LastPass		[42]	●	○	●		●	

Security

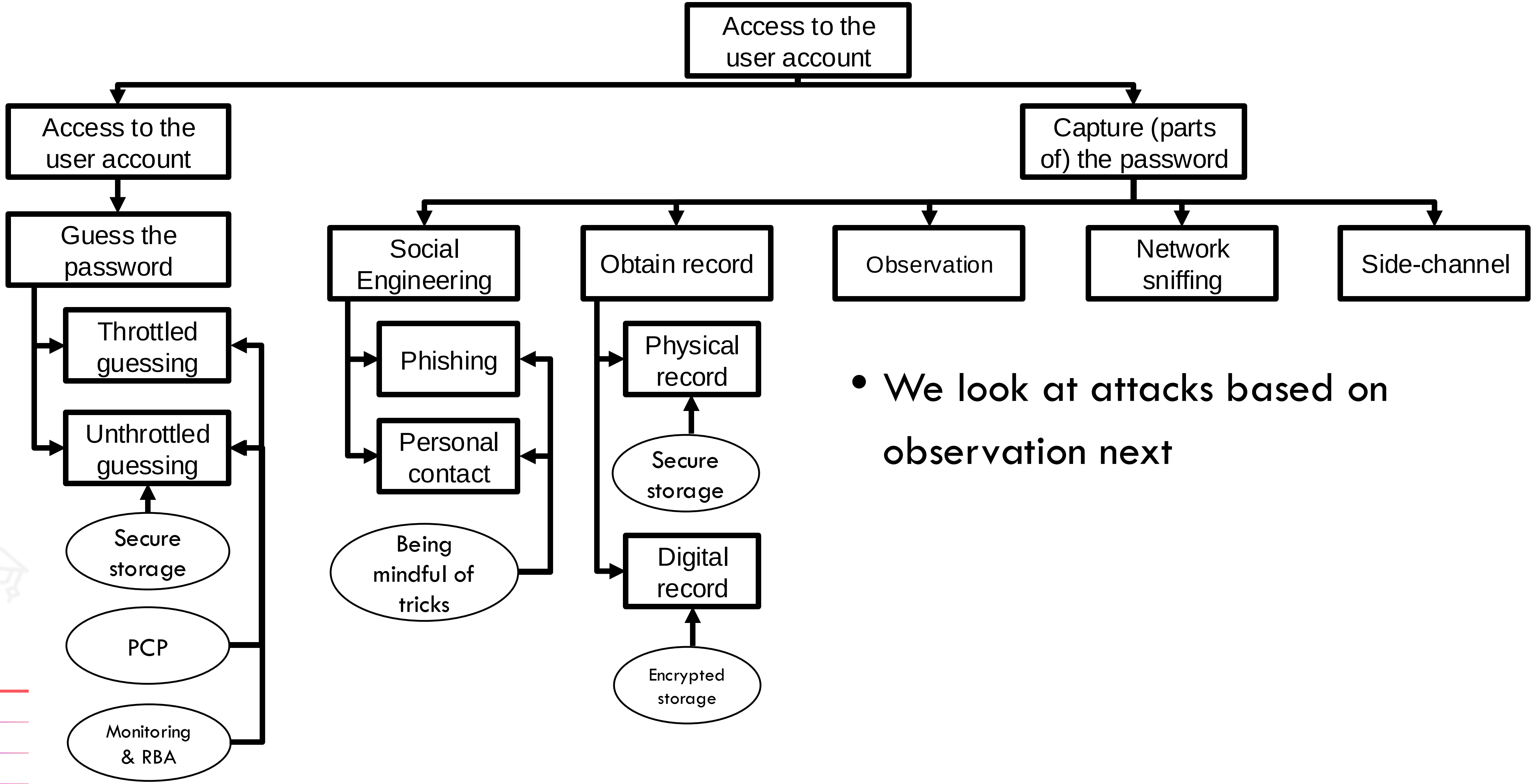
Scheme	Described in section	Reference	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Throttled-Guessing*	*Resilient-to-Unthrottled-Guessing*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*No-Trusted-Third-Party*	*Requiring-Explicit-Consent*	*Unlinkable*
Web passwords	III	[13]		○						●	●	●	●
Firefox	IV-A	[22]	○	○					●	●	●	●	●
LastPass		[42]	○	○	○	○		○	●	●		●	●

Better in terms of usability, except for recovery

Strictly worse in terms of deployability

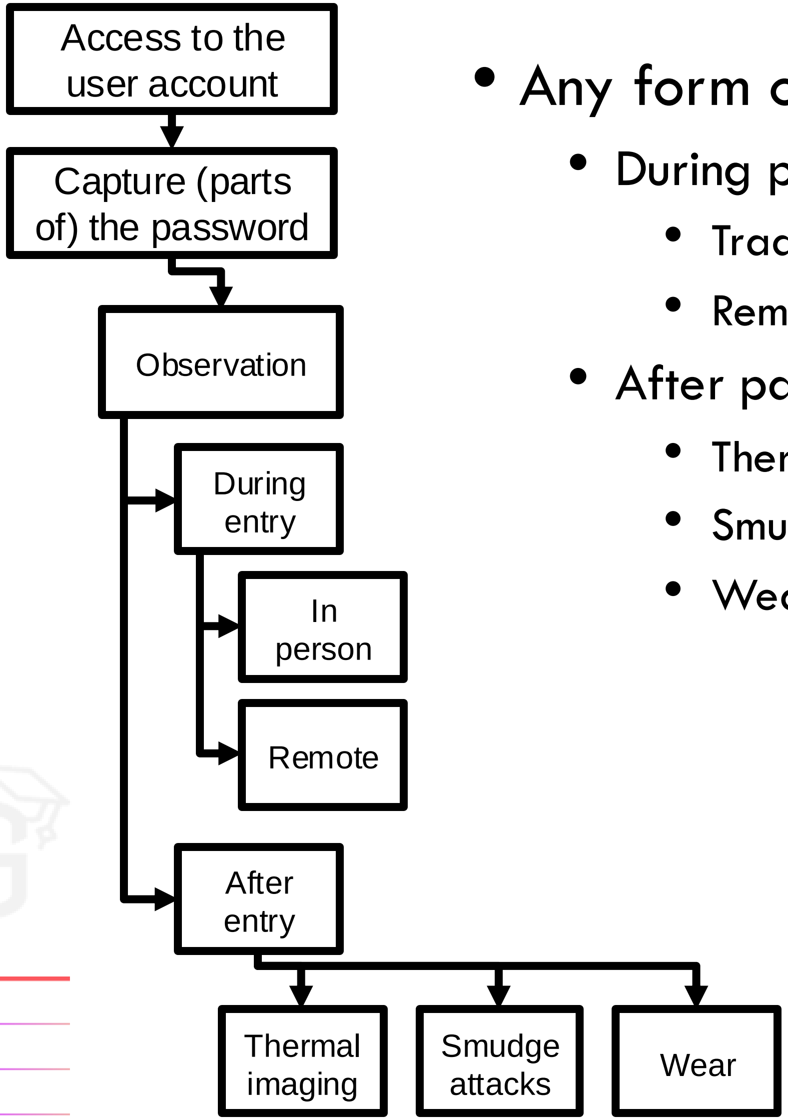
Better against several attacks, but we need to trust 3rd party

THE ATTACK TREE SO FAR



- We look at attacks based on observation next

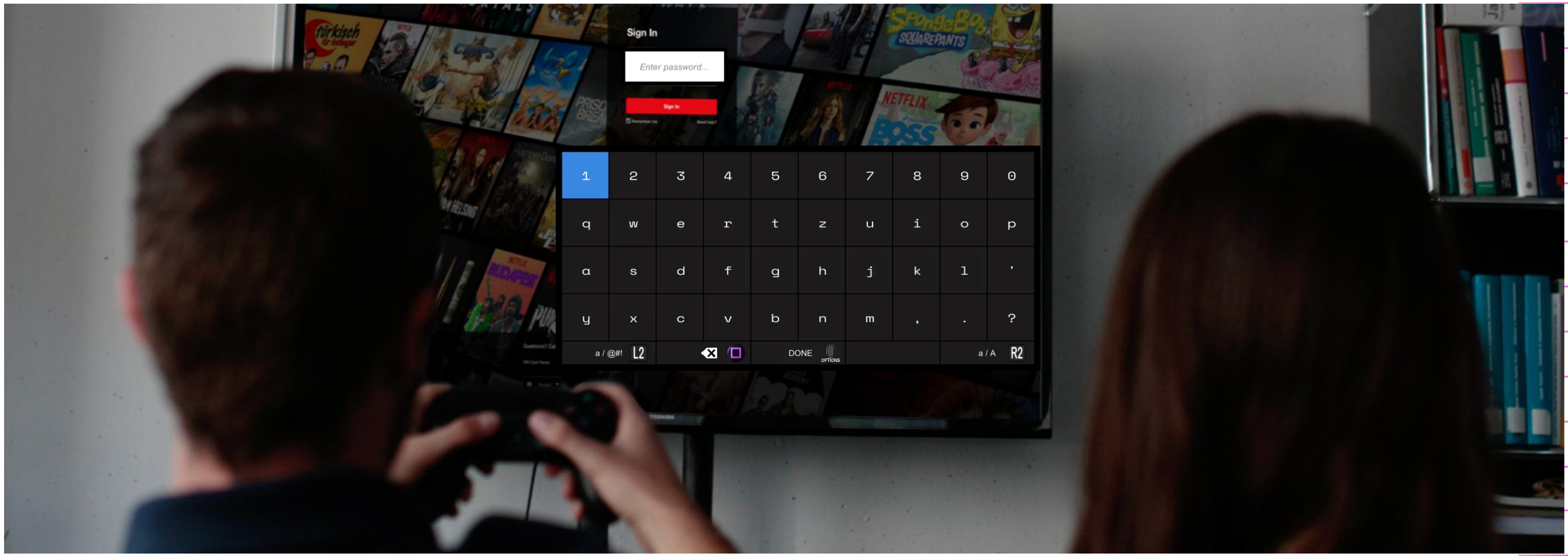
OBSERVATION ATTACKS – BASICS



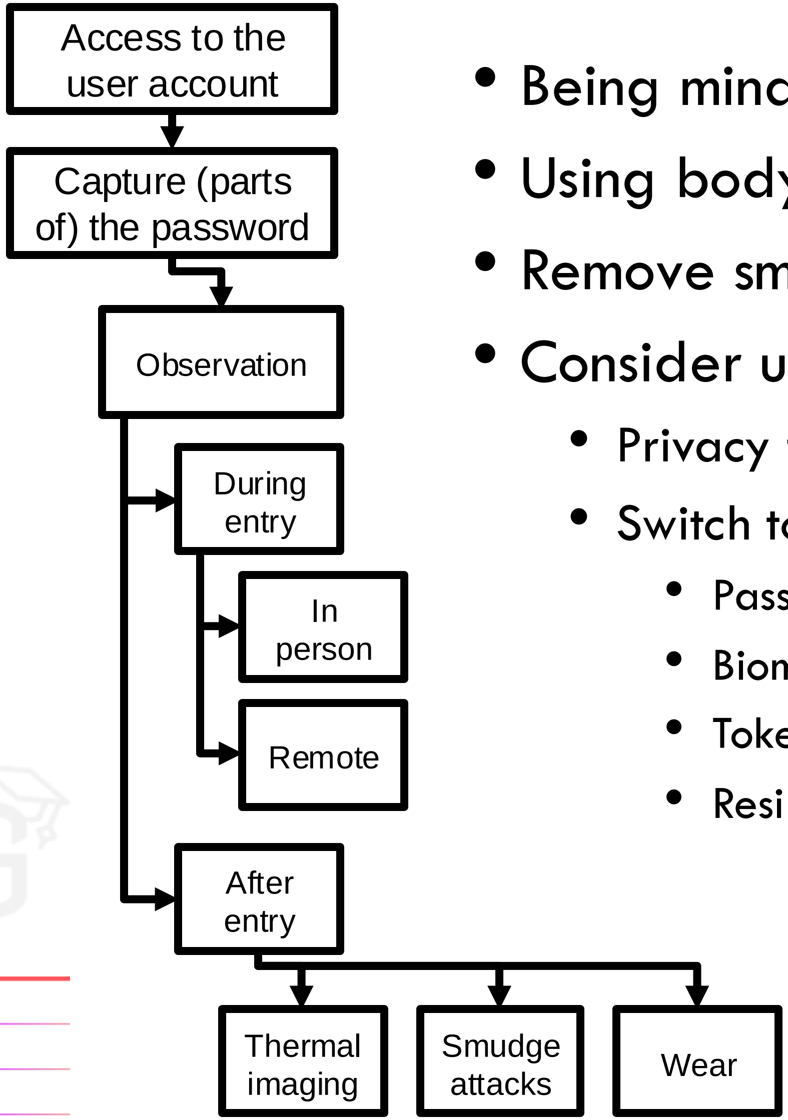
- Any form of physical observation
  - During password entry
    - Traditional shoulder-surfing in person (also with camera, mic, etc.)
    - Remote (e.g., with camera)
  - After password entry
    - Thermal imaging
    - Smudge attacks
    - Wear of input device

OBSERVATION ATTACKS – SOCIAL INFLUENCE

- Access to the user account
  - Capture (parts of) the password
    - Observation
      - During entry
        - In person
        - Remote
      - After entry
        - Thermal imaging
        - Smudge attacks
        - Wear

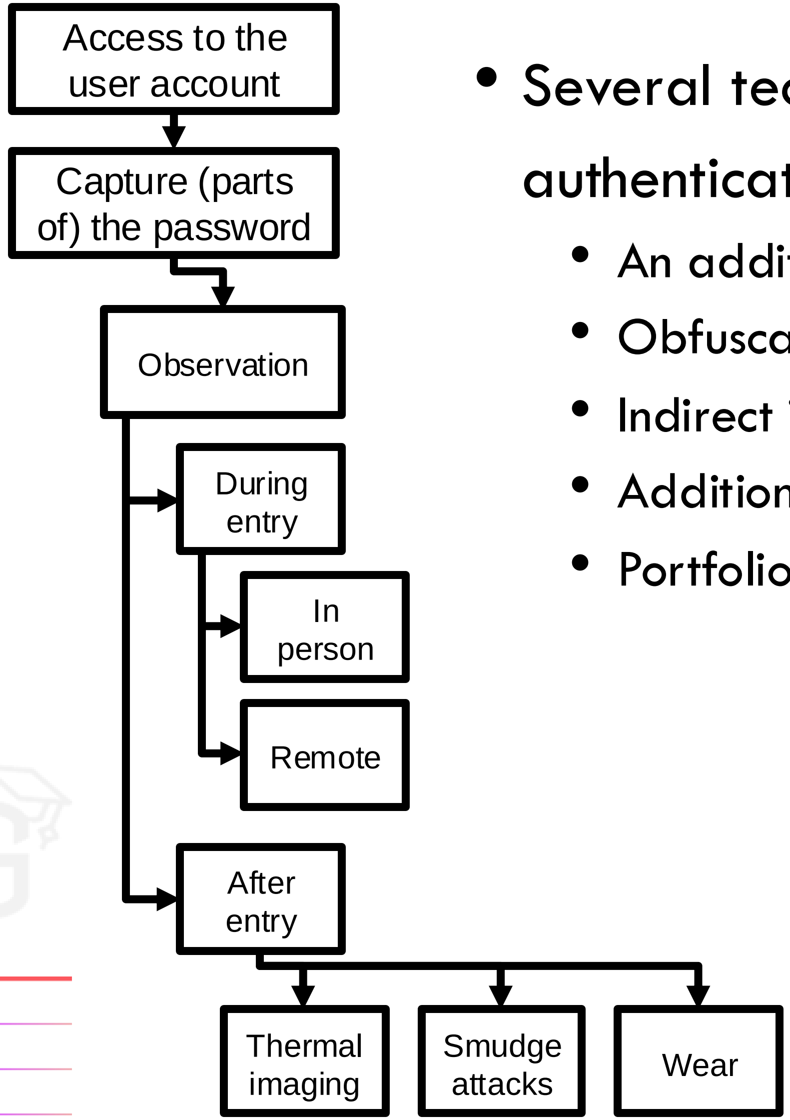


OBSERVATION ATTACKS – DEFENCES FOR USERS



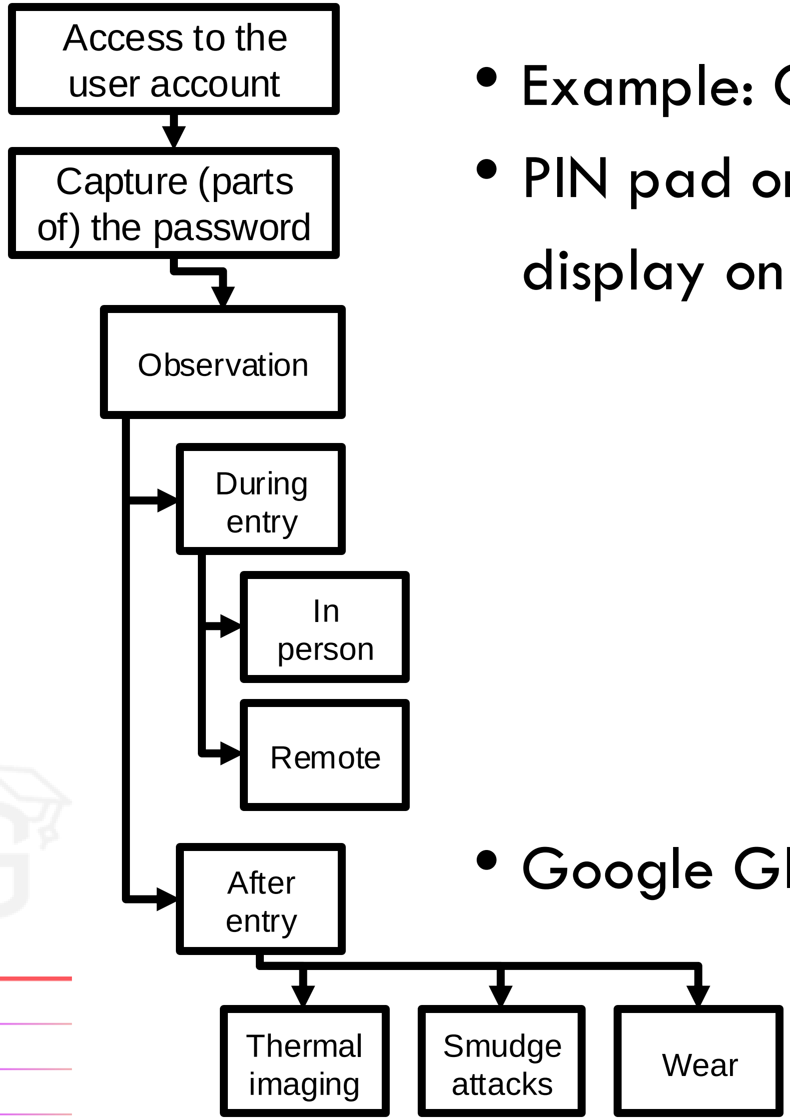
- Being mindful of surroundings
- Using body to shield screens
- Remove smudges from touchscreens
- Consider using technologies that assist in defending
  - Privacy filter for phone
  - Switch to resistant authentication scheme
    - Password managers
    - Biometrics
    - Tokens
    - Resilient knowledge-based authentication scheme

OBSERVATION ATTACKS – DEFENCES IN DESIGN

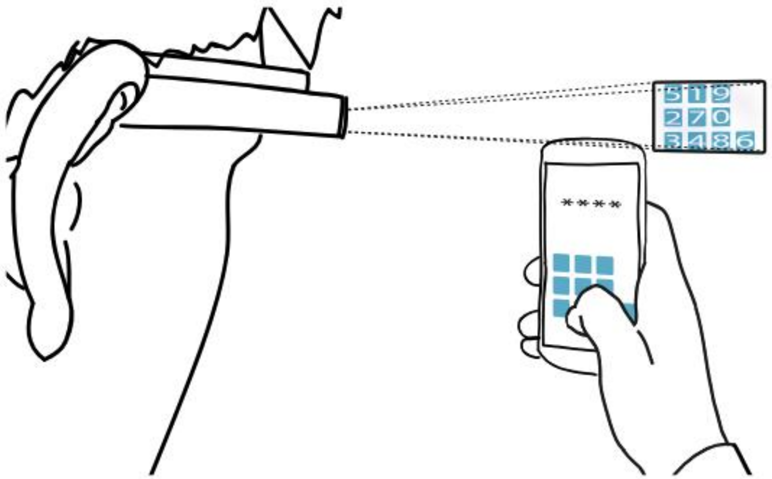


- Several techniques can be used to design knowledge-based authentication schemes with resilience to observation attacks
  - An additional covert channel for the user
  - Obfuscation of the user's input through distractors
  - Indirect input
  - Additional factor
  - Portfolio Authentication

OBSERVATION ATTACKS – COVERT CHANNELS



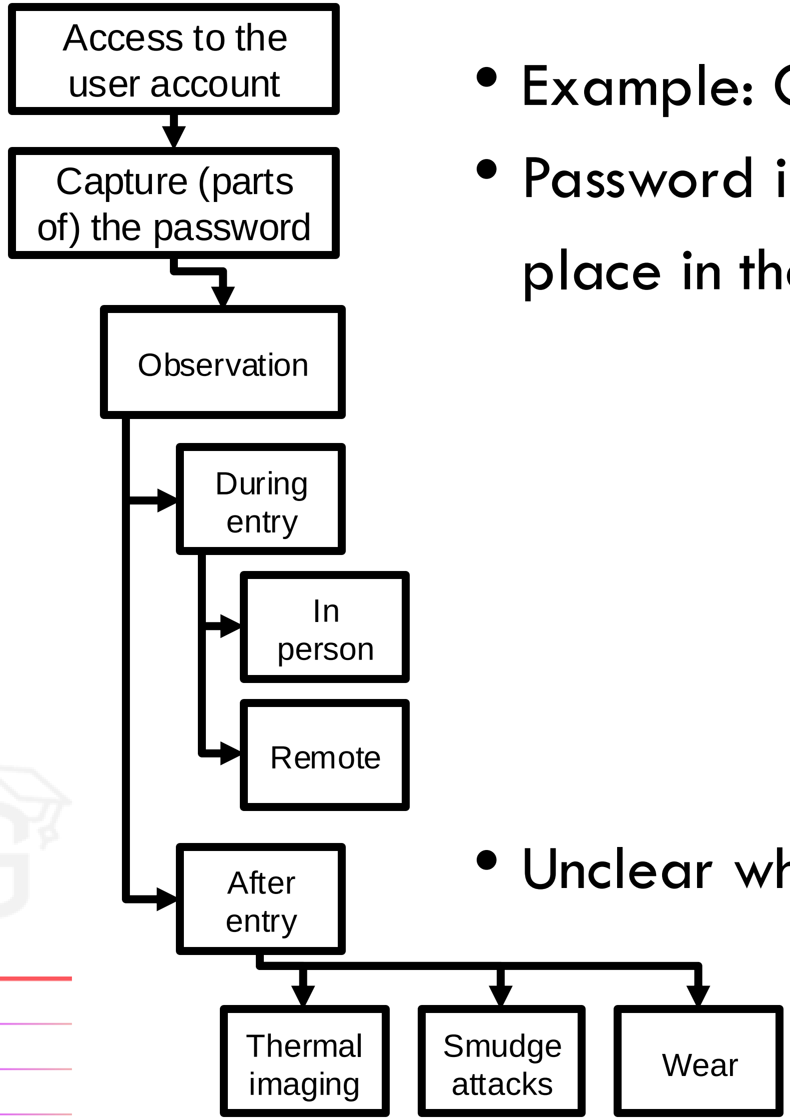
- Example: Glass Unlock (Winkler et al. 2015)
- PIN pad on phone has no numbers, instead numbers are shown in display on Google Glass AR headset



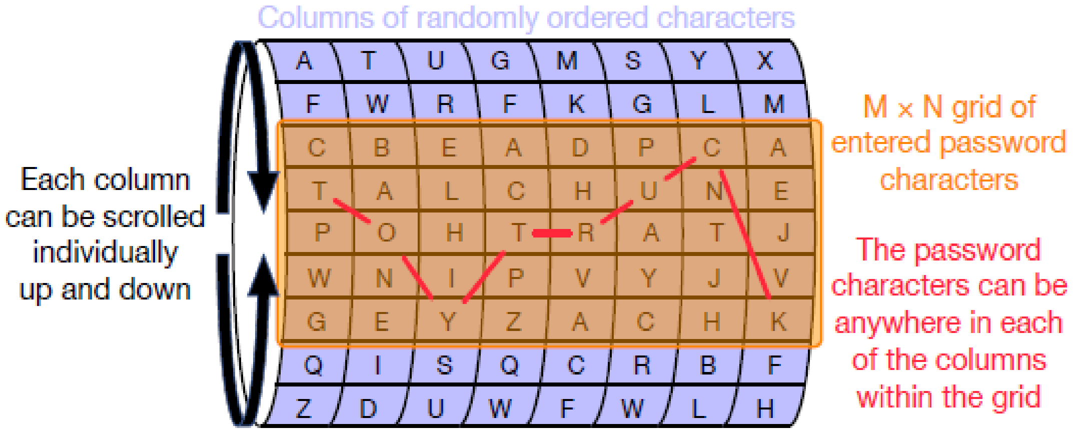
- Google Glass display is covert channel to user

Winkler, Christian, et al. "Glass unlock: Enhancing security of smartphone unlocking through leveraging a private near-eye display." Proceedings of the 33rd annual acm conference on human factors in computing systems. 2015.

OBSERVATION ATTACKS – OBFUSCATION



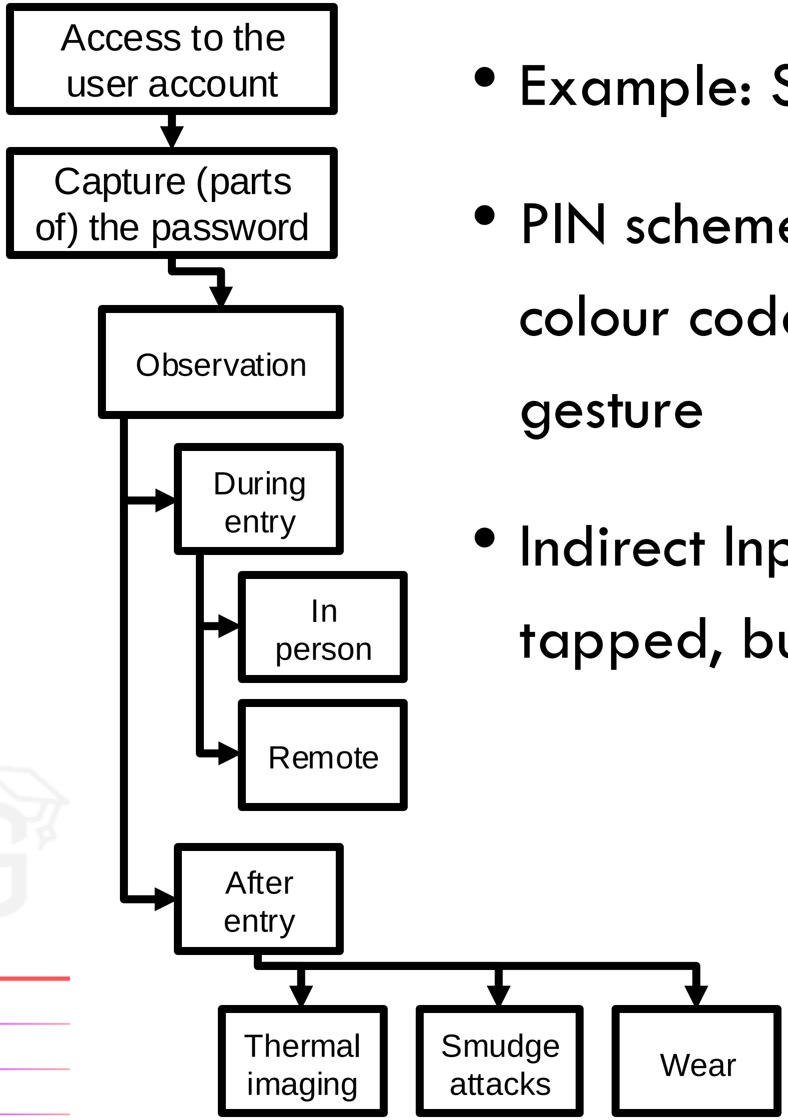
- Example: Grid scheme (Kim et al. 2011)
- Password is entered using “wheels”, multiple characters for each place in the password visible



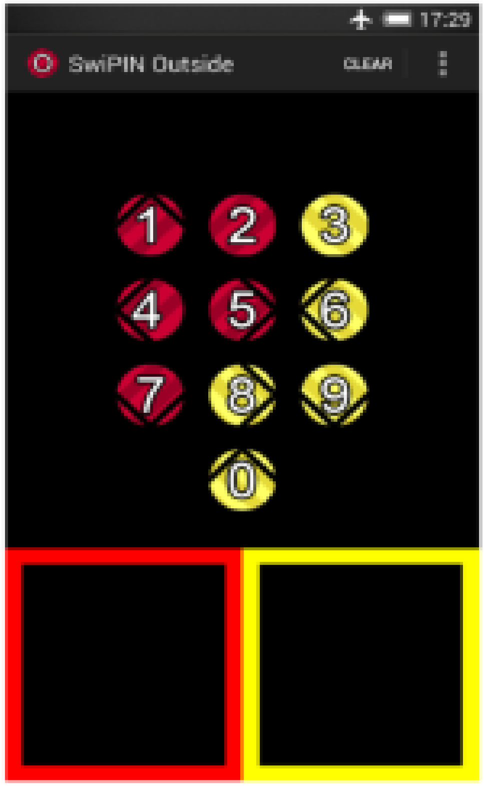
- Unclear which character is the one in the password

Kim, Sung-Hwan, et al. "A new shoulder-surfing resistant password for mobile environments." Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication. 2011.

OBSERVATION ATTACKS – INDIRECT INPUT

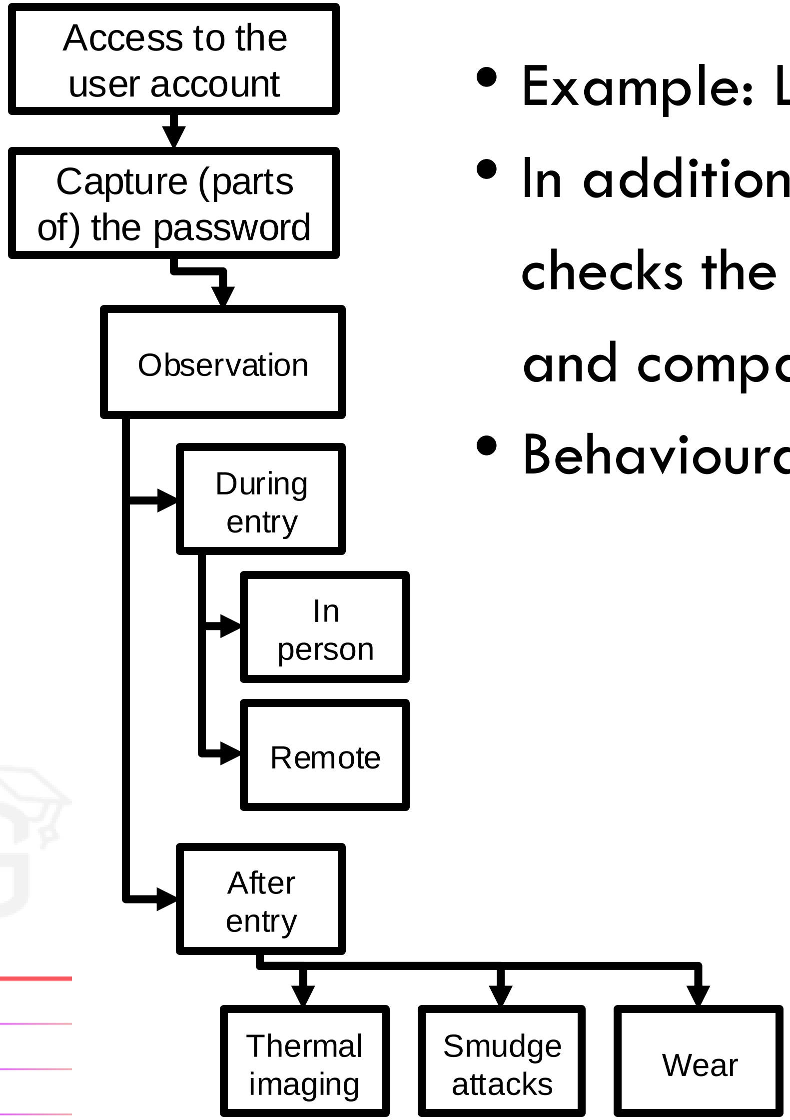


- Example: Swipin (von Zezschwitz et al. 2015)
- PIN scheme where numbers are colour coded and require specific gesture
- Indirect Input: Number is not directly tapped, but instead the field below

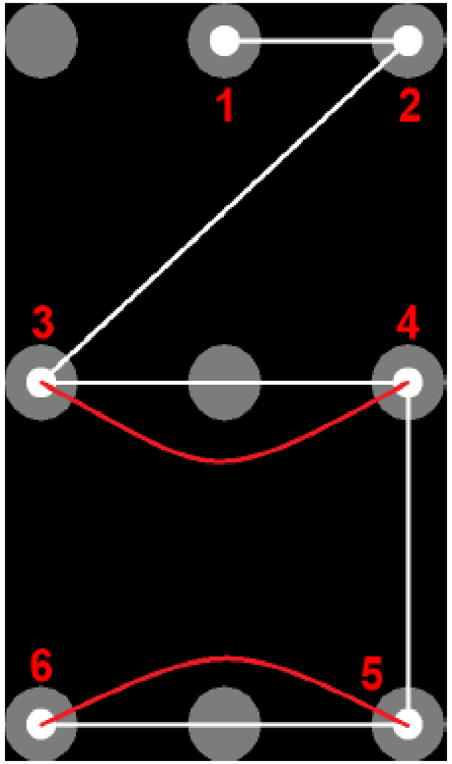


Von Zezschwitz, Emanuel, et al. "Swipin: Fast and secure pin-entry on smartphones." Proceedings of the 33rd annual acm conference on human factors in computing systems. 2015.

OBSERVATION ATTACKS – ADDITIONAL FACTORS

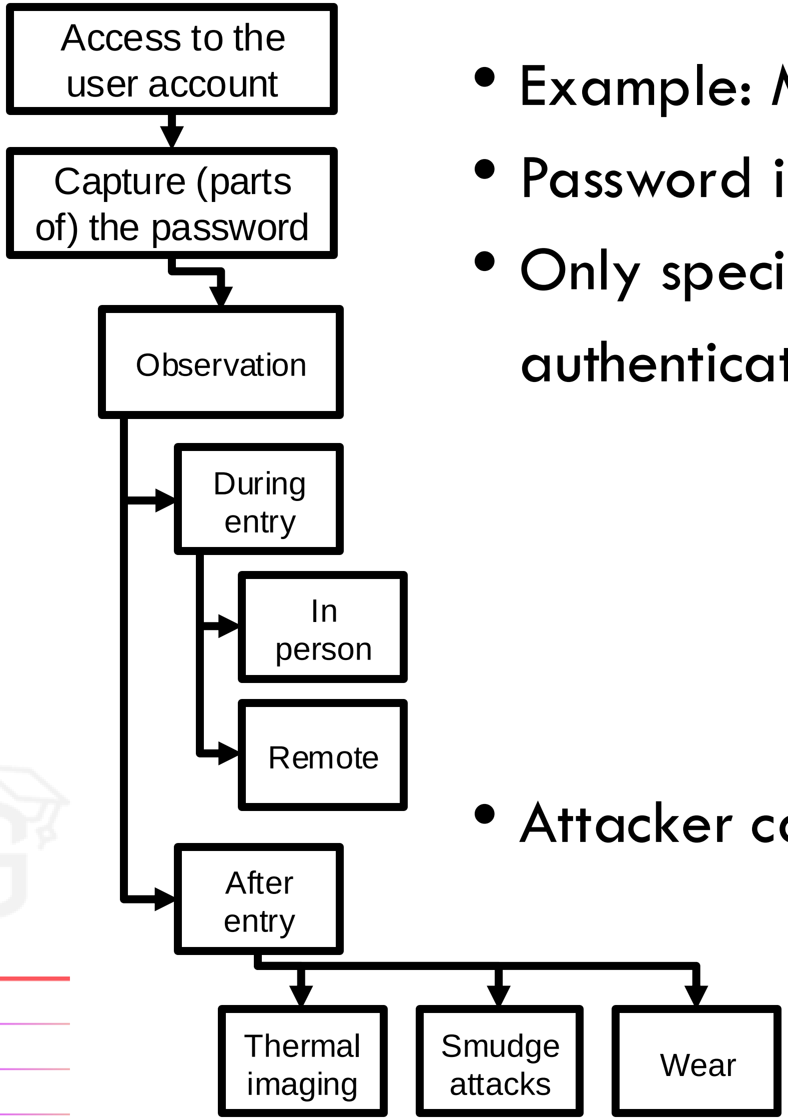


- Example: Lock pattern biometrics (De Luca et al. 2012)
- In addition to pattern the scheme checks the input characteristics and compares to user samples
- Behavioural biometric

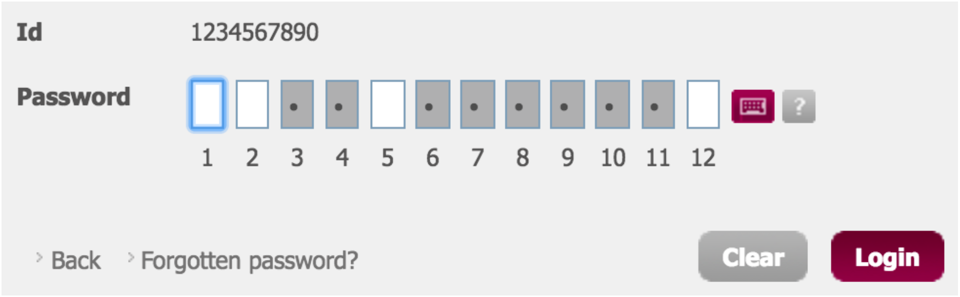


De Luca, Alexander, et al. "Touch me once and i know it's you! implicit authentication based on touch screen patterns." proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2012.

OBSERVATION ATTACKS – PORTFOLIO AUTHENTICATION

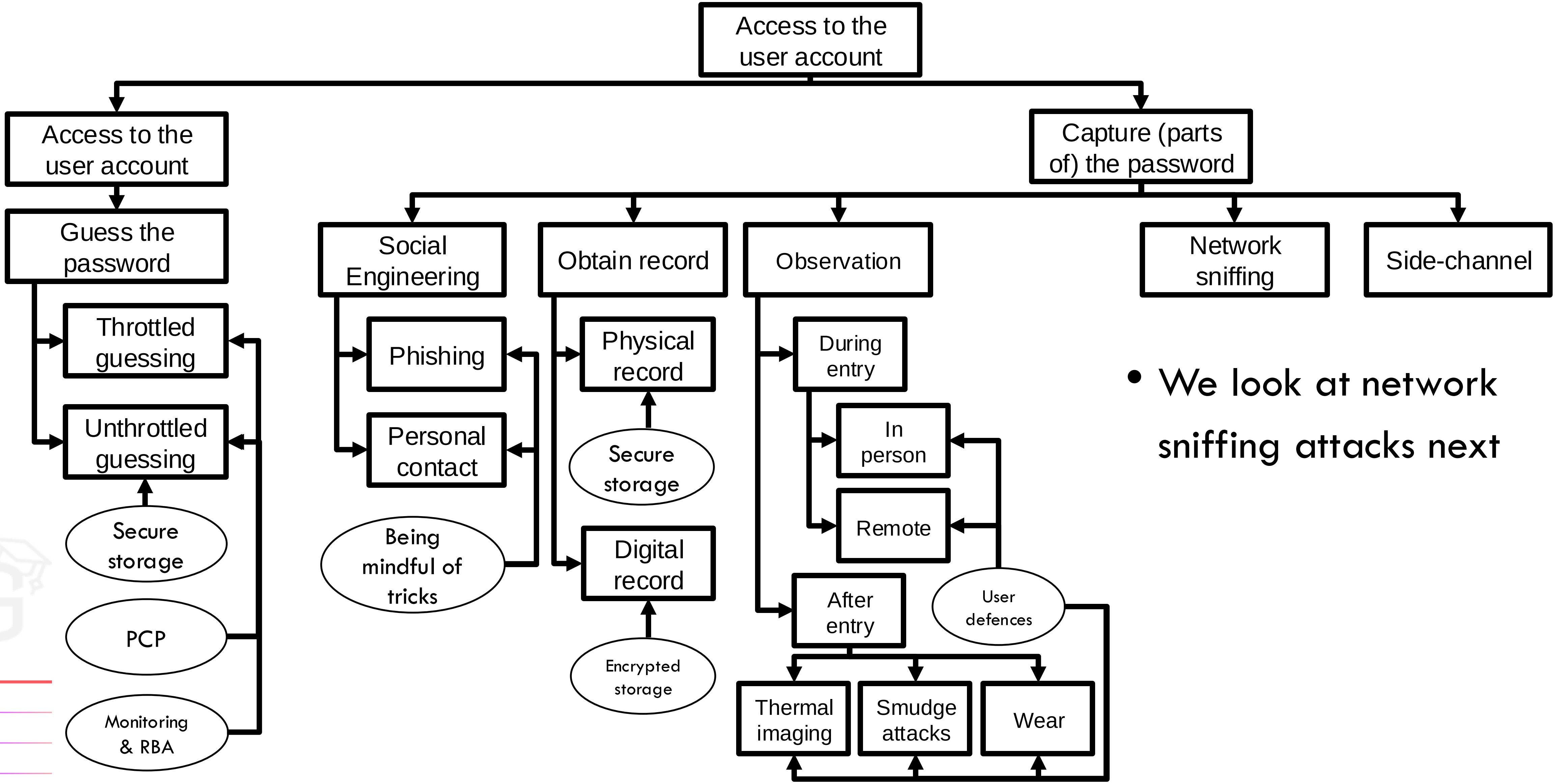


- Example: Masked passwords
- Password is portfolio of elements
- Only specific subset of password has to be entered for each authentication attempt



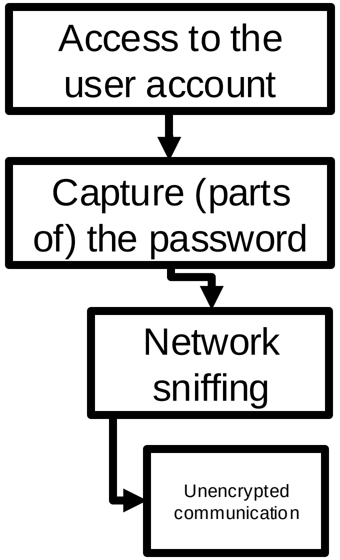
- Attacker cannot capture entire password in one observation

THE ATTACK TREE SO FAR

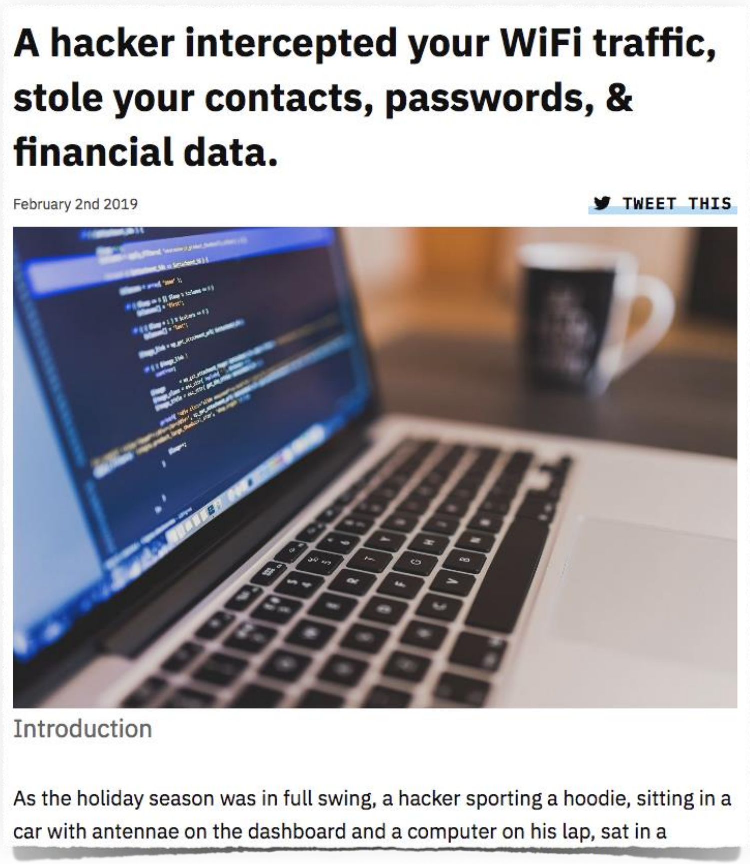


- We look at network sniffing attacks next

NETWORK SNIFFING – UNENCRYPTED COMMUNICATION



- Secrets should never be transmitted unencrypted
- Otherwise attackers can sniff
  - Passwords
  - Compromises account long-term
- Session IDs
  - Compromises account as long as session is active
  - That is why some sites have such aggressive time-outs of the login



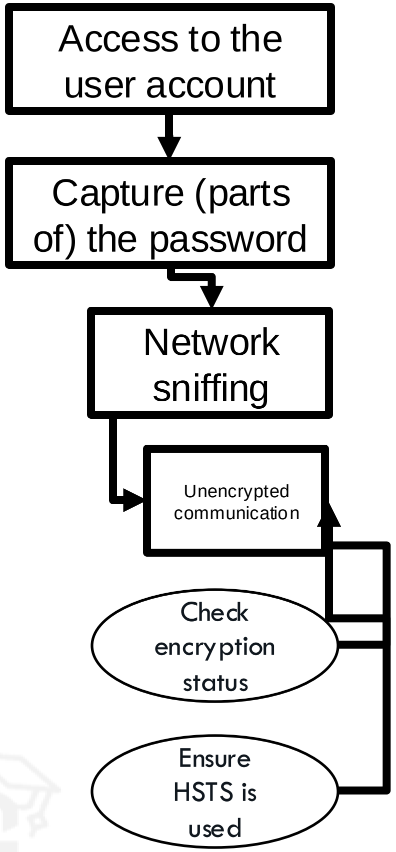
A hacker intercepted your WiFi traffic, stole your contacts, passwords, & financial data.

February 2nd 2019 TWEET THIS

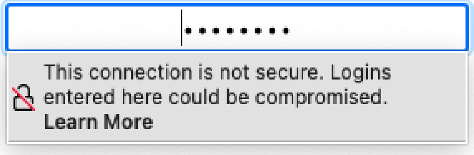
Introduction

As the holiday season was in full swing, a hacker sporting a hoodie, sitting in a car with antennae on the dashboard and a computer on his lap, sat in a

NETWORK SNIFFING – UNENCRYPTED COMMUNICATION



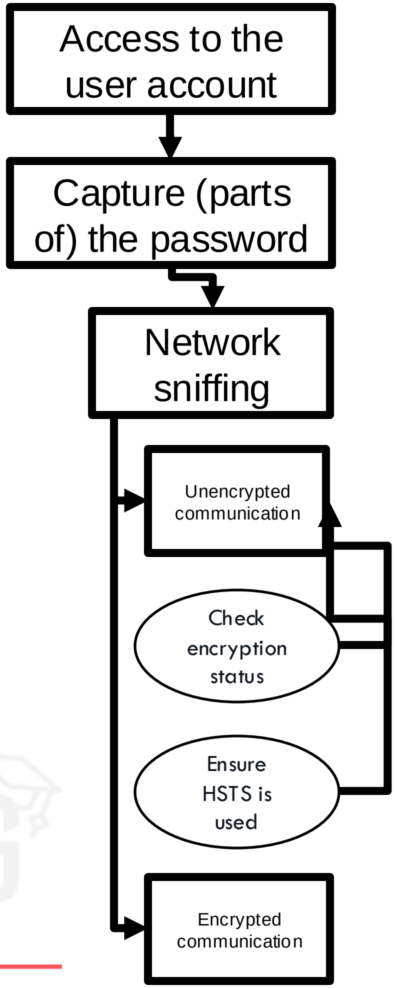
- Defence: Making sure encryption is used, e.g., TLS
  - Web services should use HTTP Strict Transport Security
  - Modern browsers have indicators



- You can try out how it looks in your browser using the website https://badssl.com/
  - Allows simulating a wide variety of TLS errors
  - Offers sites without TLS

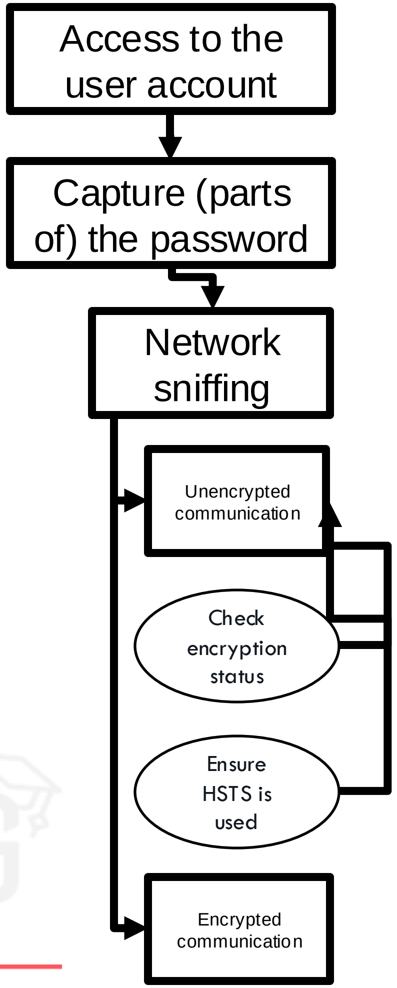
- **Q:** A company could also decide to block all non TLS traffic. Is that generally a good idea?

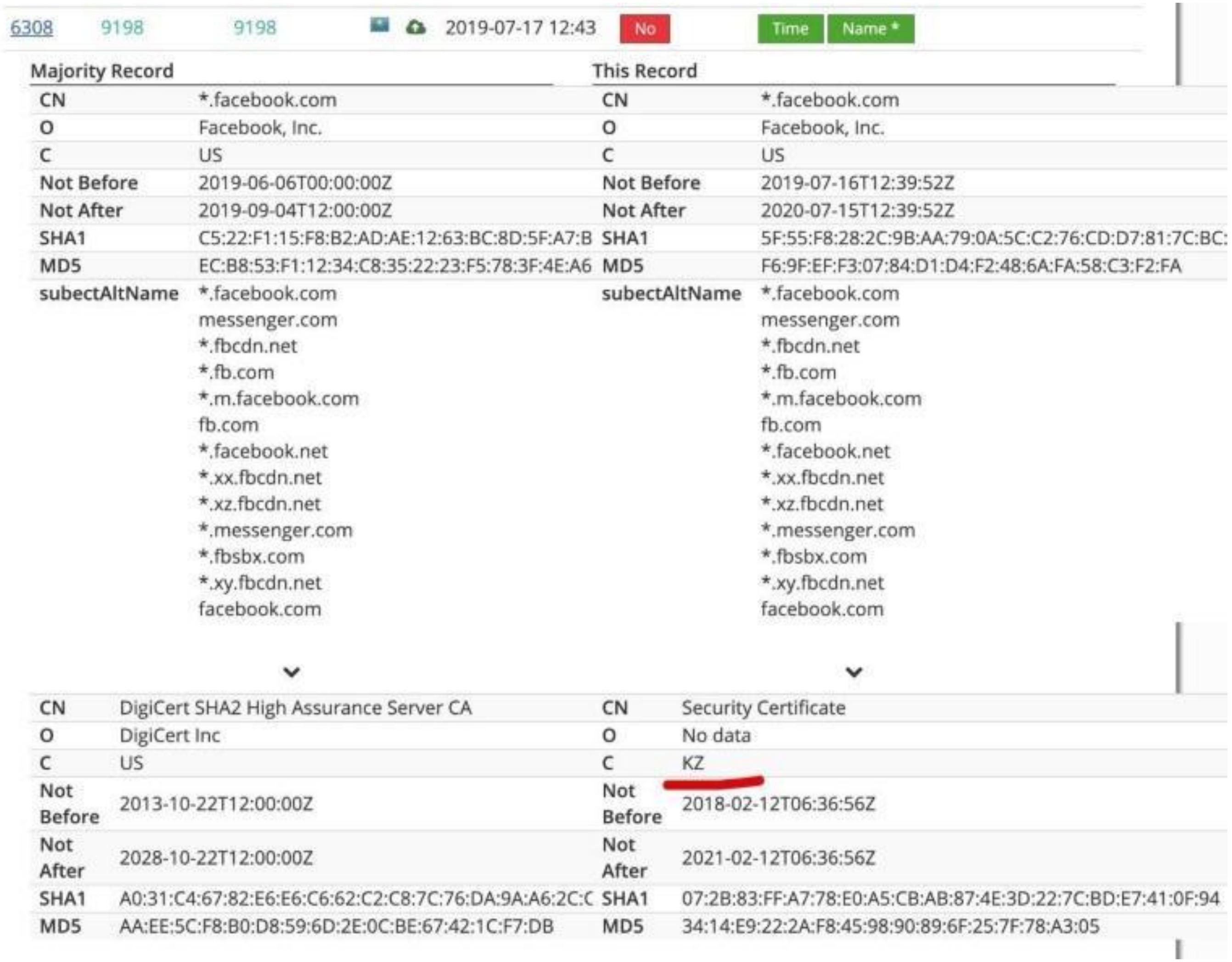
NETWORK SNIFFING – ENCRYPTED COMMUNICATION



- To sniff encrypted traffic, the attacker must break into the encryption
  - Attacker must impersonate remote web service
  - Victims thinks they communicate with the web-server but instead it is the attacker

NETWORK SNIFFING – ENCRYPTED COMMUNICATION



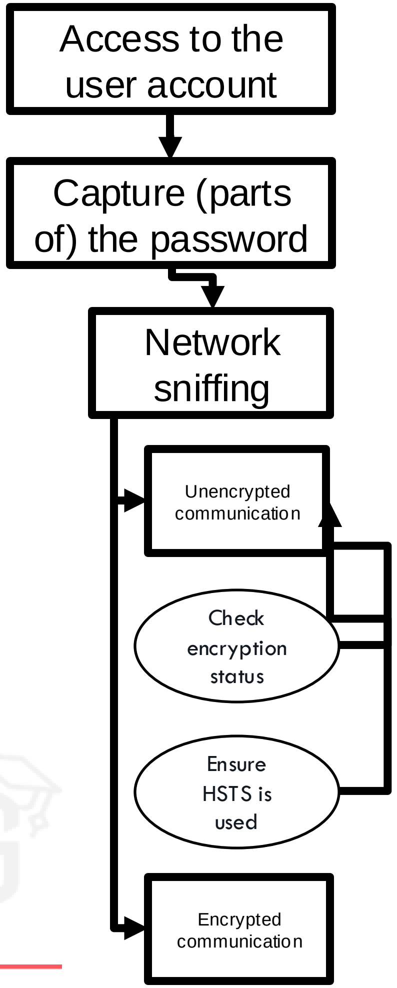


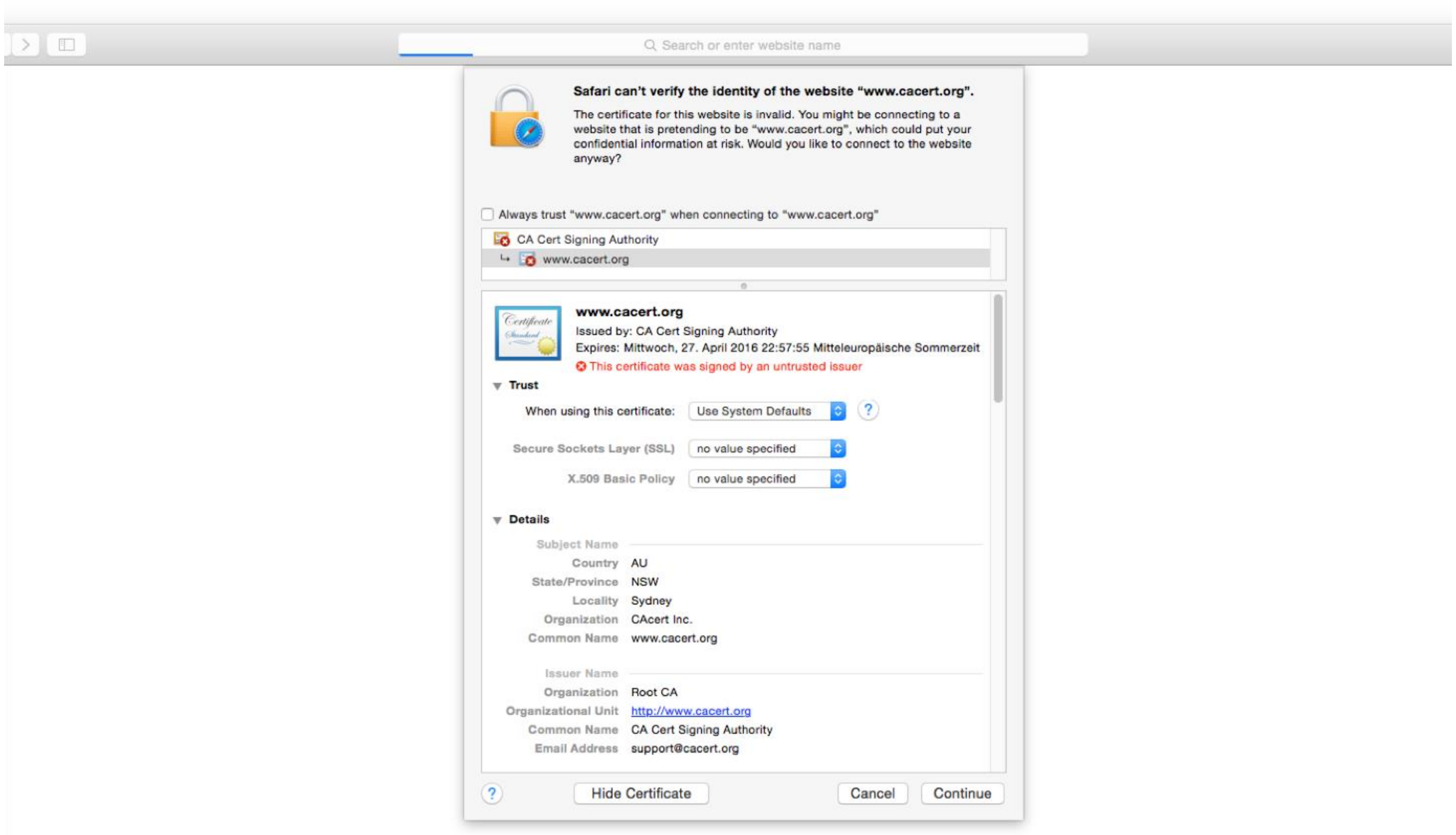
6308 9198 9198 2019-07-17 12:43 No Time Name *

Majority Record		This Record	
CN	*.facebook.com	CN	*.facebook.com
O	Facebook, Inc.	O	Facebook, Inc.
C	US	C	US
Not Before	2019-06-06T00:00:00Z	Not Before	2019-07-16T12:39:52Z
Not After	2019-09-04T12:00:00Z	Not After	2020-07-15T12:39:52Z
SHA1	C5:22:F1:15:F8:B2:AD:AE:12:63:BC:8D:5F:A7:B	SHA1	5F:55:F8:28:2C:9B:AA:79:0A:5C:C2:76:CD:D7:81:7C:BC:
MD5	EC:B8:53:F1:12:34:C8:35:22:23:F5:78:3F:4E:A6	MD5	F6:9F:EF:F3:07:84:D1:D4:F2:48:6A:FA:58:C3:F2:FA
subectAltName	*.facebook.com messenger.com *.fbcdn.net *.fb.com *.m.facebook.com fb.com *.facebook.net *.xx.fbcdn.net *.xz.fbcdn.net *.messenger.com *.fbsbx.com *.xy.fbcdn.net facebook.com	subectAltName	*.facebook.com messenger.com *.fbcdn.net *.fb.com *.m.facebook.com fb.com *.facebook.net *.xx.fbcdn.net *.xz.fbcdn.net *.messenger.com *.fbsbx.com *.xy.fbcdn.net facebook.com

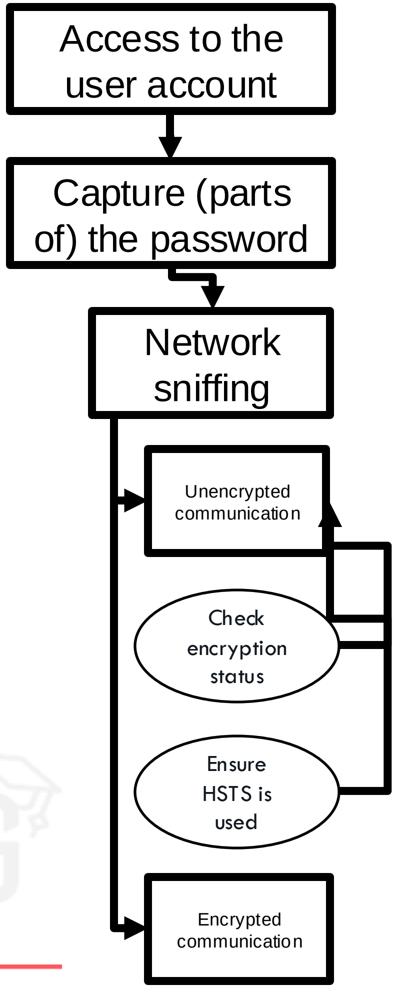
CN	DigiCert SHA2 High Assurance Server CA	CN	Security Certificate
O	DigiCert Inc	O	No data
C	US	C	KZ
Not Before	2013-10-22T12:00:00Z	Not Before	2018-02-12T06:36:56Z
Not After	2028-10-22T12:00:00Z	Not After	2021-02-12T06:36:56Z
SHA1	A0:31:C4:67:82:E6:E6:C6:62:C2:C8:7C:76:DA:9A:A6:2C:C	SHA1	07:2B:83:FF:A7:78:E0:A5:CB:AB:87:4E:3D:22:7C:BD:E7:41:0F:94
MD5	AA:EE:5C:F8:B0:D8:59:6D:2E:0C:BE:67:42:1C:F7:DB	MD5	34:14:E9:22:2A:F8:45:98:90:89:6F:25:7F:78:A3:05

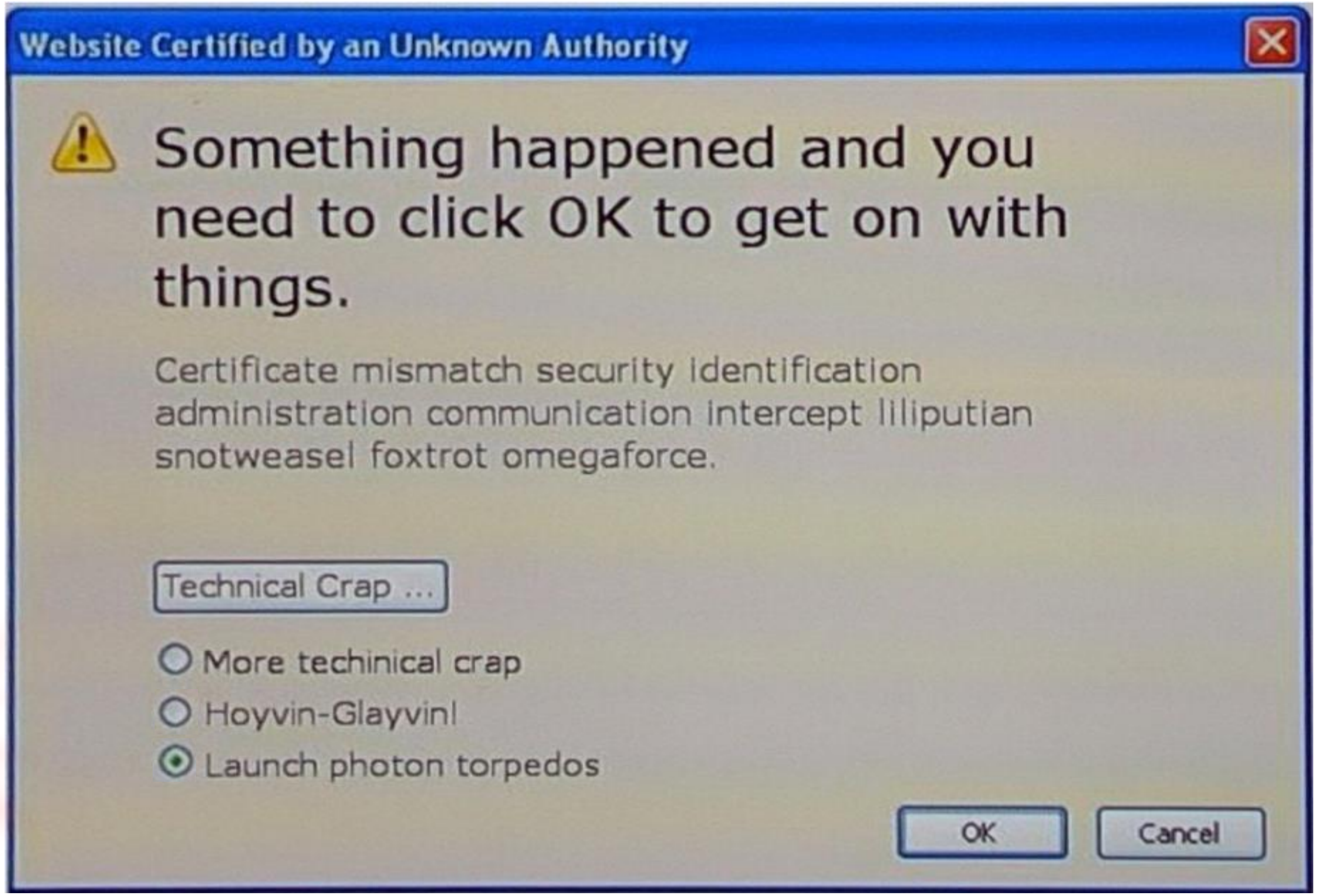
NETWORK SNIFFING – ENCRYPTED COMMUNICATION



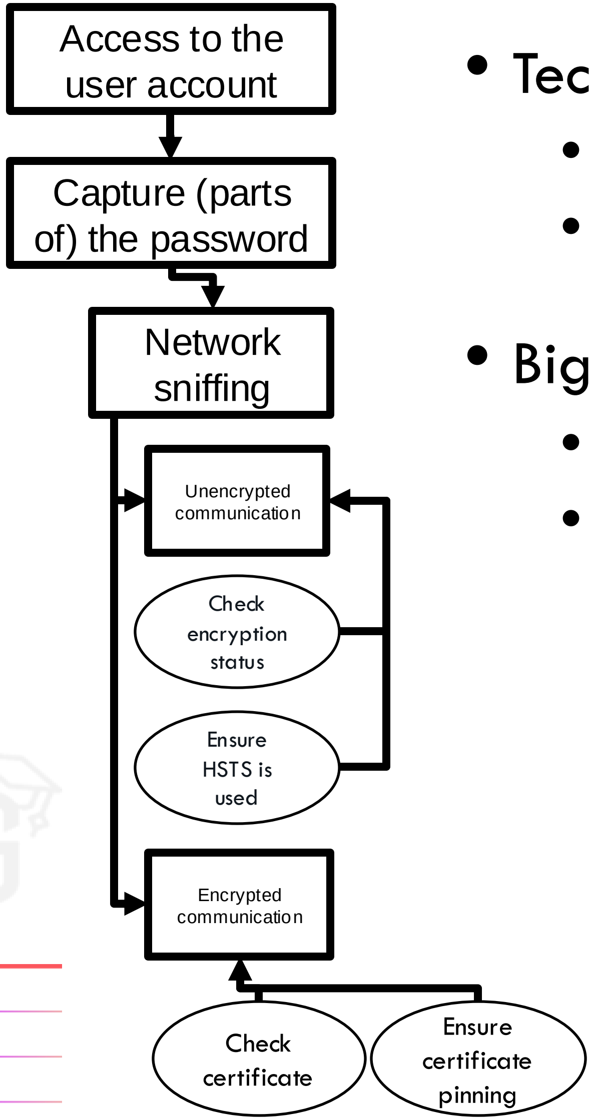


NETWORK SNIFFING – ENCRYPTED COMMUNICATION



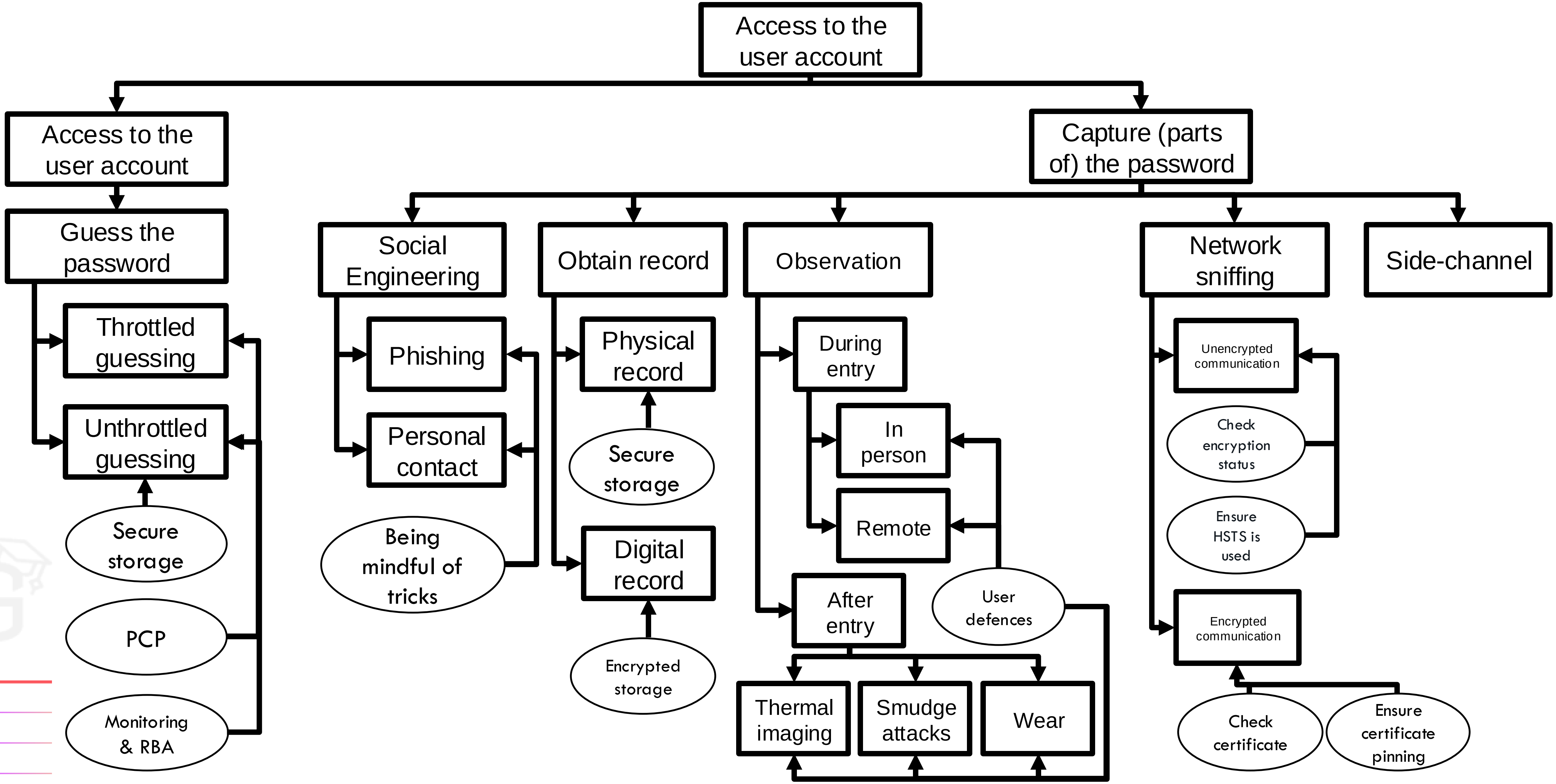


NETWORK SNIFFING – ENCRYPTED COMMUNICATION

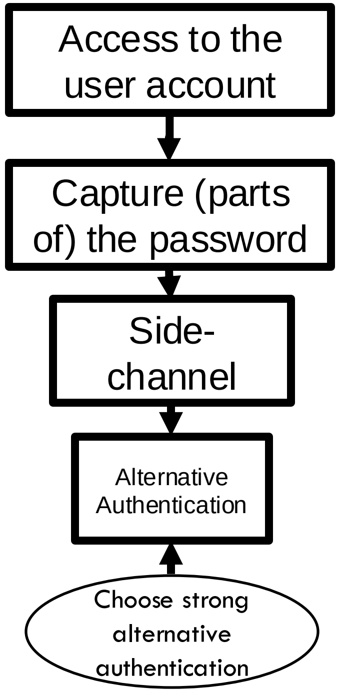


- Technical measure by web services
  - Certificate pinning
  - Client that has visited previously will only accept the pinned certificate
- Biggest problem in practice
  - Most errors are due to misconfiguration, >90%
  - E.g., the BSI, the German governmental authority on cybersecurity forgot to renew their certificate in 2018…
    - Everyone who visited their website saw an error message

THE ATTACK TREE SO FAR

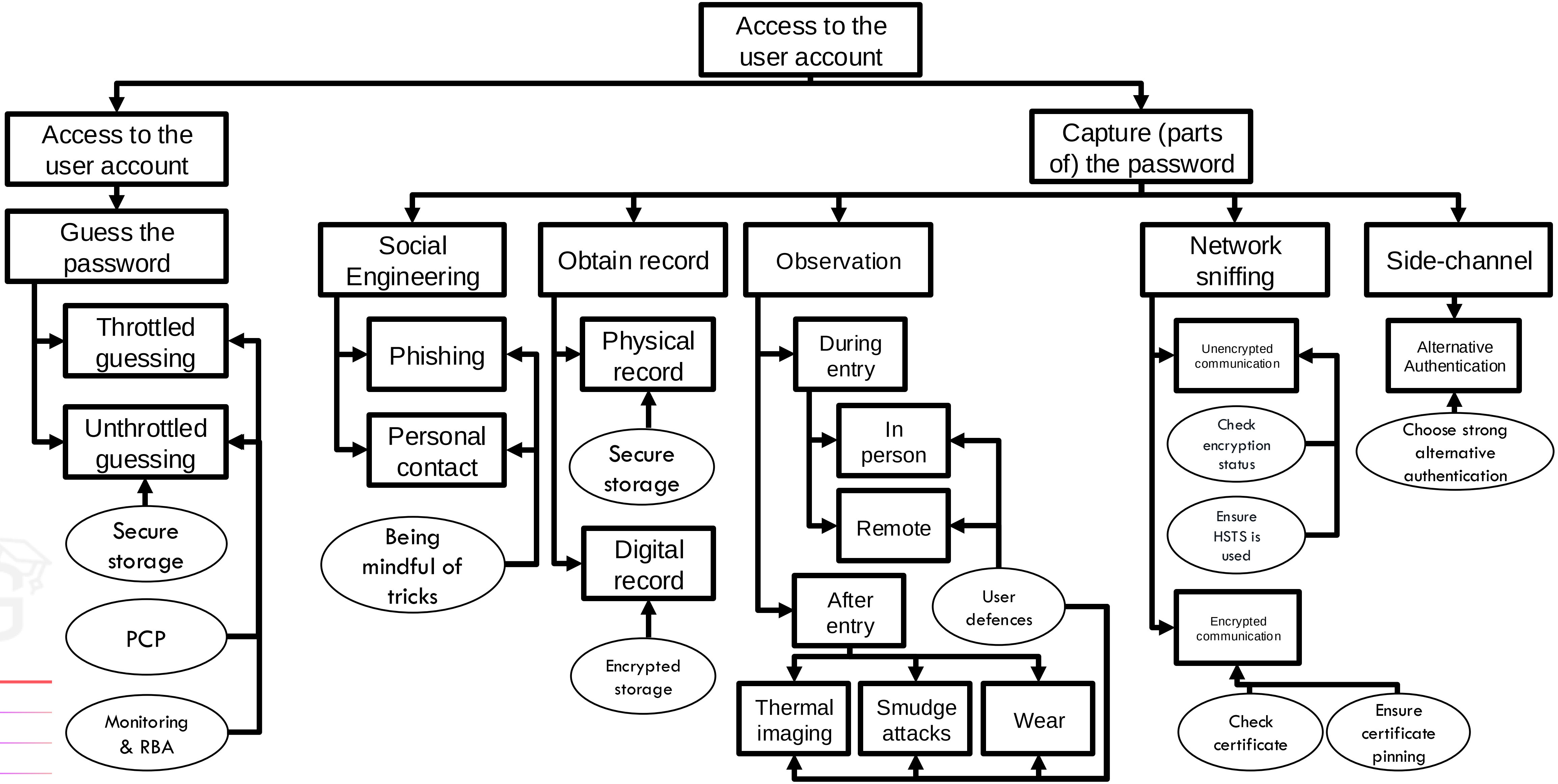


SIDE CHANNELS – ALTERNATIVE AUTHENTICATION SCHEMES



- Most web services allow resetting the password through an alternative authentication scheme, e.g.,
  - Link in email
  - Security questions
- If this alternative authentication scheme is weaker than the primary authentication scheme, it can be exploited by an attacker

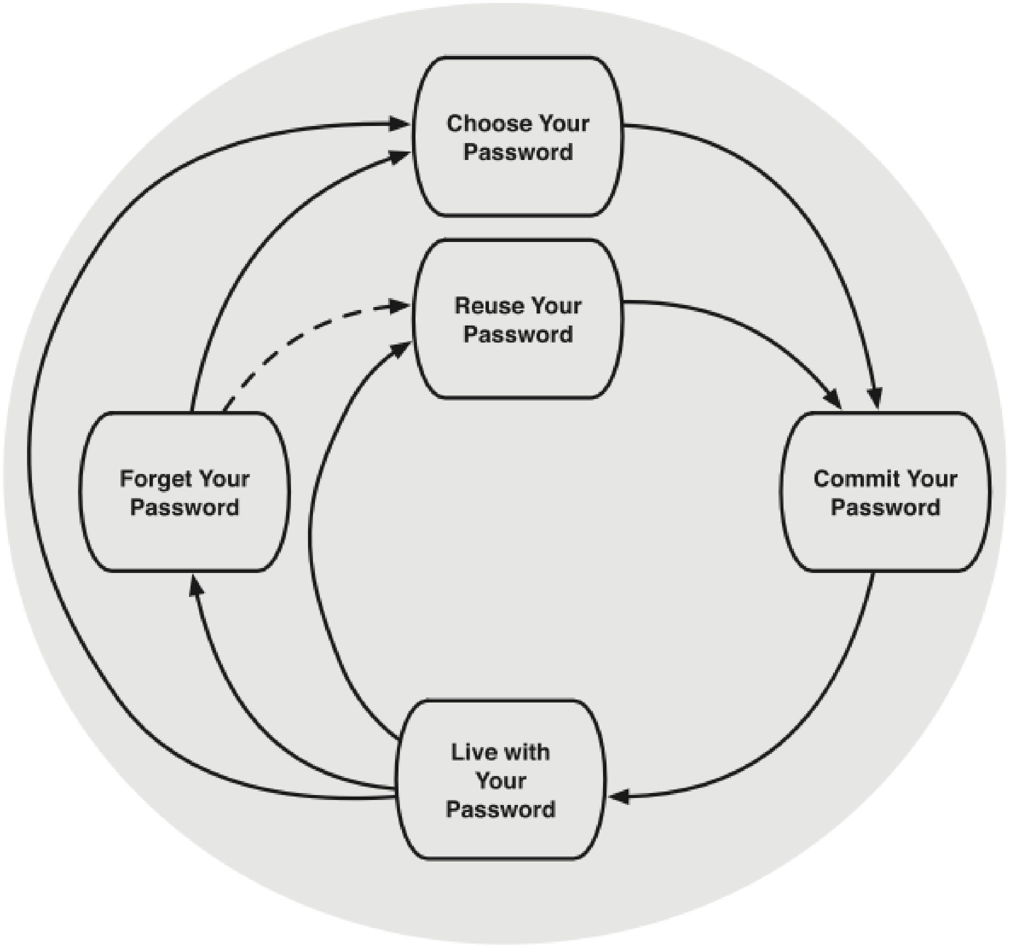
THE FINAL ATTACK TREE



WHAT CAN PEOPLE DO TO THWART ATTACKS?

THE PASSWORD LIFE-CYCLE

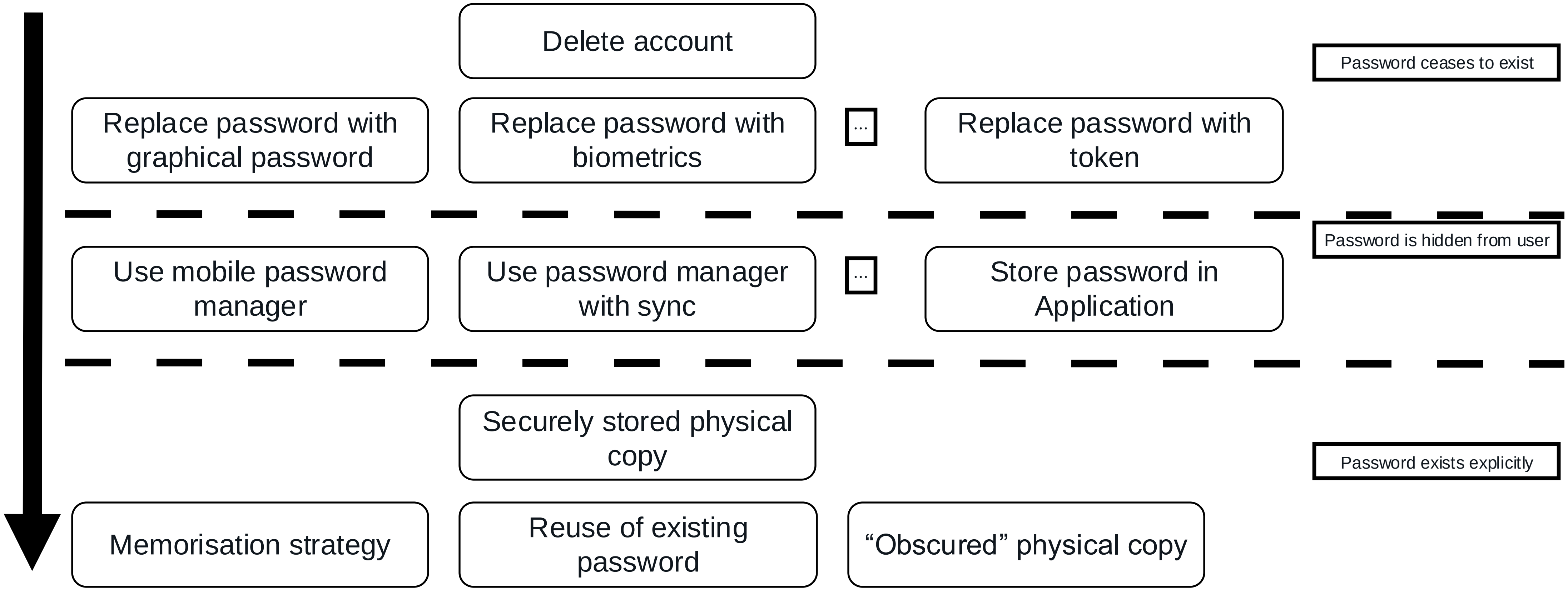
- Interviews with lay-users and experts (Stobert & Biddle 2018)



GENERAL ADVICE

- Budget well and spent effort where it makes sense
  - E.g., do not unnecessarily change passwords
- Humans can handle about 4-7 passwords
  - Most important passwords are
    - Primary email account
    - Logins for devices
    - Vault password for password manager
    - Password for Single-Sign-On
  - Only the most unimportant accounts should have reused passwords (if at all)
  - Think about which attacks are relevant for which accounts
    - E.g., is shoulder surfing relevant for the online banking password you are only using at home?
- Best is to consciously decide on own strategy

FINDING YOUR OWN STRATEGY – WHAT ARE THE OPTIONS



FINDING YOUR OWN STRATEGY – THE LONG WAY

This part of the page you can show to others whom you trust to receive feedback regarding your assessment. However, you should always cut off the right side of the page beforehand.

This part of the page can be kept in a safe place as a backup for emergencies.

Basic Information	Website/Purpose mein-paketservice.de User name (if applicable) max.muster	Website/Purpose User name (if applicable)	Website/Purpose User name (if applicable)	Website/Purpose User name (if applicable)
Data and actions	- Meine aktuelle Adresse - Vergangene Adressen - Adressen der Empfänger - Kreditkartendaten (VISA)			
Possible solutions	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☒ saved in password manager ☐ saved in mobile App ☐ written down	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☐ saved in password manager ☐ saved in mobile App ☐ written down	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☐ saved in password manager ☐ saved in mobile App ☐ written down	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☐ saved in password manager ☐ saved in mobile App ☐ written down
Chosen solutions	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☒ saved in password manager ☐ saved in mobile App ☐ written down	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☐ saved in password manager ☐ saved in mobile App ☐ written down	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☐ saved in password manager ☐ saved in mobile App ☐ written down	☐ Account can be deleted ☐ Two-factor-authentication offered Password can be ☐ replaced with SSO ☐ saved in password manager ☐ saved in mobile App ☐ written down
Requirements if password must be memorized	Security ☐ Resist “Targetted Guessing” ☐ Resist “Untargetted Guessing” ☐ Resist “System Break-In” Others ☐ Must be changed periodically ☐ Is shared with others	Security ☐ Resist “Targetted Guessing” ☐ Resist “Untargetted Guessing” ☐ Resist “System Break-In” Others ☐ Must be changed periodically ☐ Is shared with others	Security ☐ Resist “Targetted Guessing” ☐ Resist “Untargetted Guessing” ☐ Resist “System Break-In” Others ☐ Must be changed periodically ☐ Is shared with others	Security ☐ Resist “Targetted Guessing” ☐ Resist “Untargetted Guessing” ☐ Resist “System Break-In” Others ☐ Must be changed periodically ☐ Is shared with others
Passwords	Old password Pakete123 New password UquohgeeOouLeeghahna Notes on how the password was chosen Zufällig erzeugt durch Passwortmanager	Old password New password Notes on how the password was chosen	Old password New password Notes on how the password was chosen	Old password New password Notes on how the password was chosen

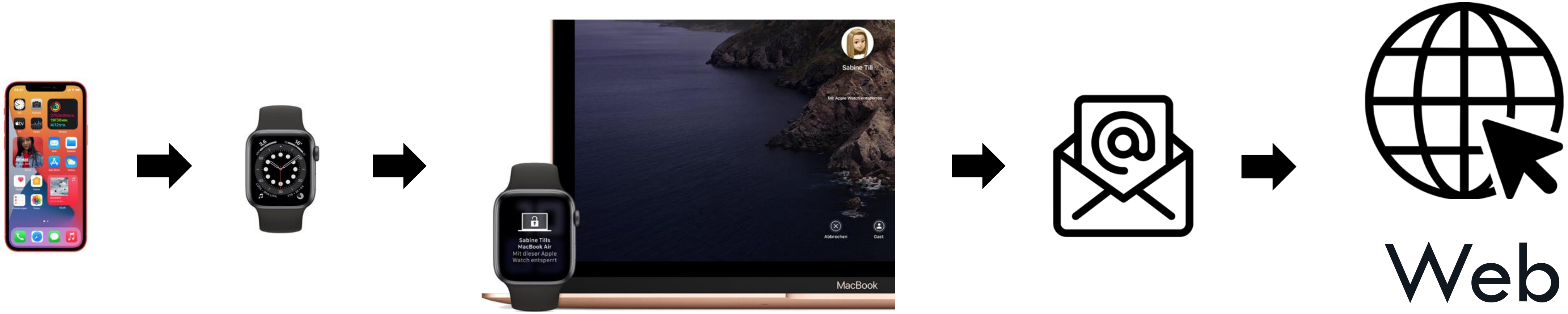
HOW TO COMPOSE PASSWORDS

- There is no simple way to just create memorable passwords, you must find your own, also makes them harder to guess (in case that is a relevant attack
- Passphrases: Many people use the same and you can build a dictionary (Yang et al. 2016)
  - E.g., To be or not to be, The quick brown fox jumps over the lazy dog
- Concatenating words, a.k.a. “CorrectHorseBatteryStaple”
  - “System-assigned passphrases performed similarly to system-assigned passwords of similar entropy across the usability metrics we examined. [...] However, passphrases took significantly longer for participants to enter, and appear to require error-correction to counteract entry mistakes.” (Shay et al. 2012)
  - “Five-word passwords also appear to negatively impact usability, only 39.7 % of participants successfully recalled their password after two weeks.” (Wu et al. 2022)

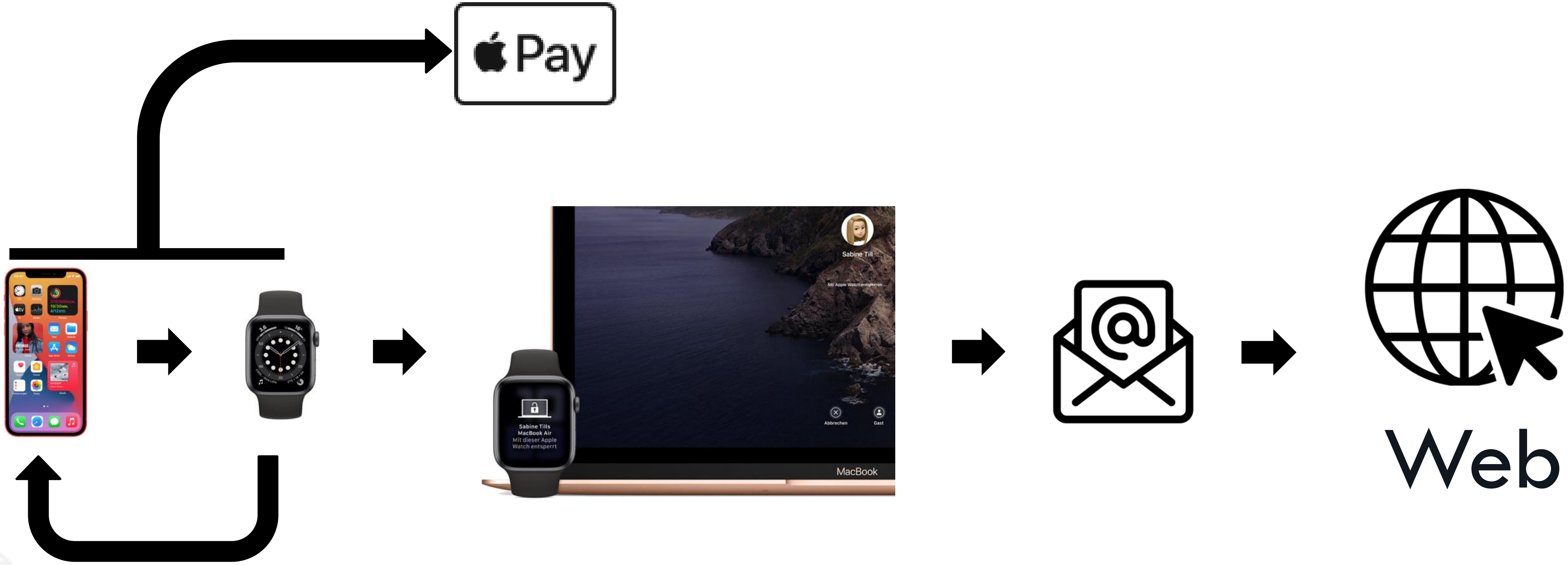
Yang, Weining, et al. "An empirical study of mnemonic sentence-based password generation strategies." Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016.
Shay, Richard, et al. "Correct horse battery staple: Exploring the usability of system-assigned passphrases." Proceedings of the eighth symposium on usable privacy and security. 2012.
Wu , Xiaoyuan, et al. ” User Perceptions of Five-Word Passwords.” Proceedings of the 38th Annual Computer Security Applications Conference. 2022.

MIND AUTHENTICATION CHAINS

MIND AUTHENTICATION CHAINS



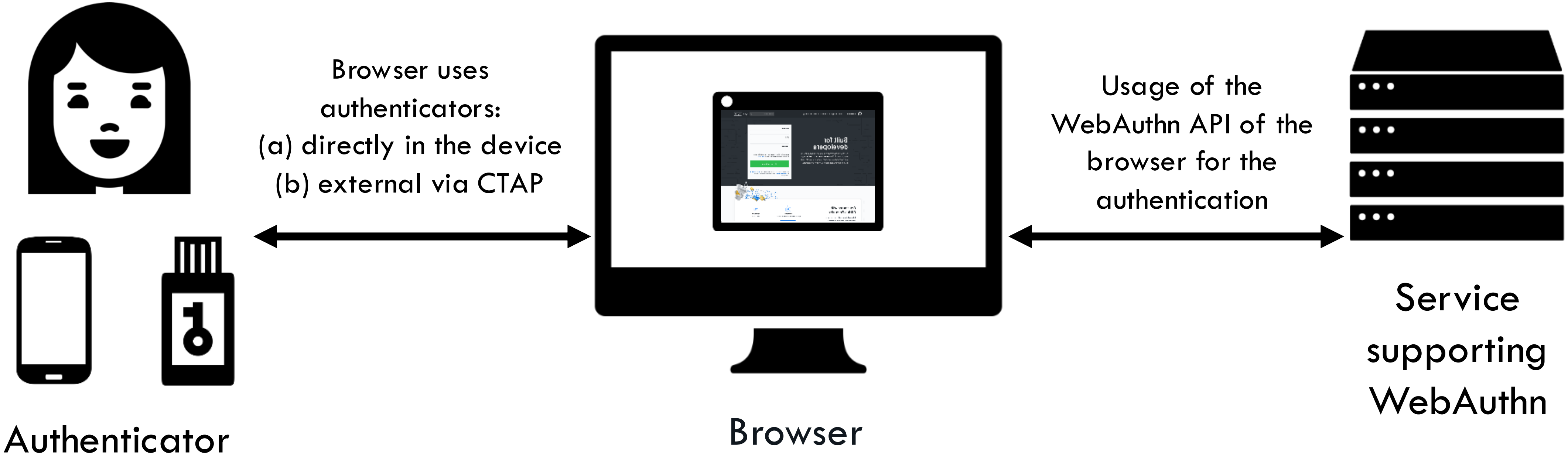
MIND AUTHENTICATION CHAINS



WHAT COULD THE FUTURE OF AUTHENTICATION LOOK LIKE?

FIDO2 – THE FUTURE OF AUTHENTICATION ON THE WEB

- This future is almost here, all major platform support it
- FIDO2 = CTAP + WebAuthn

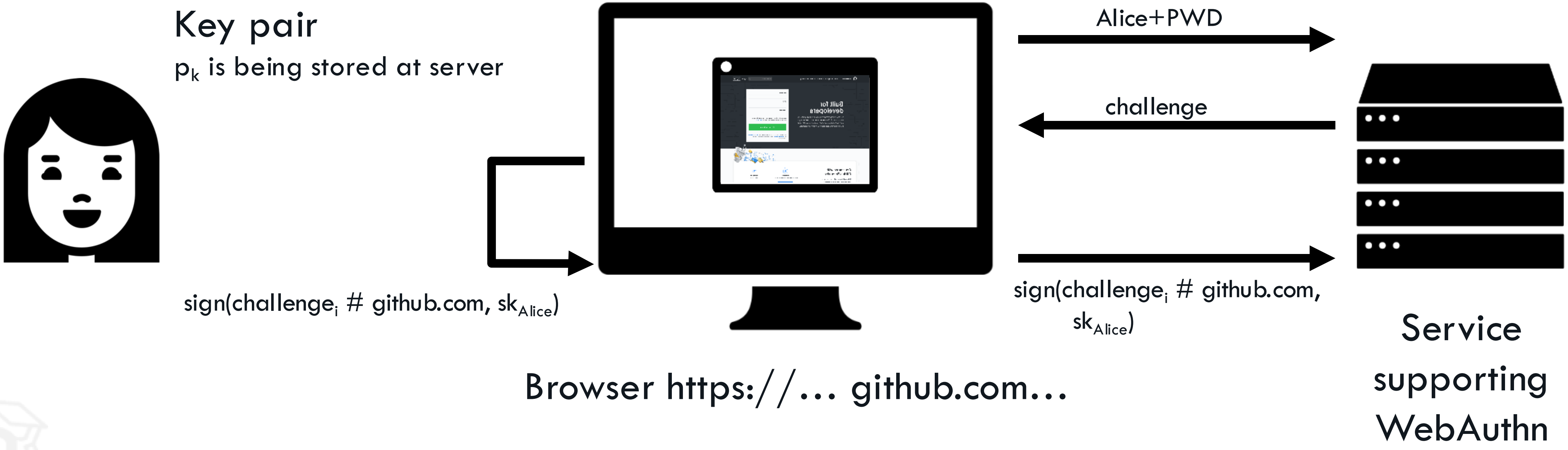


WEBAUTHN

- Standard for unidirectional authentication
  - Two use cases
    - Passwordless authentication based on public keys
    - MFA
  - Based on asymmetric cryptography and a challenge response protocol

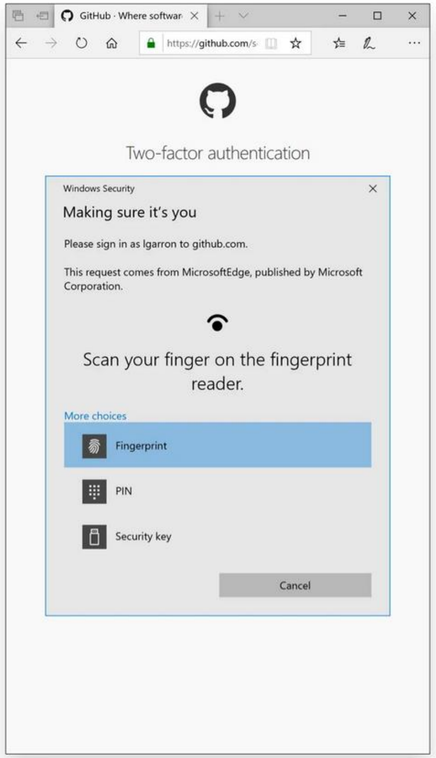
- Defining advantage: Server does not hold any password data (only public keys)
  - No secrets can be stolen from server ⇢ Guessing attacks impossible

WEBAUTHN – MFA PROCEDURE

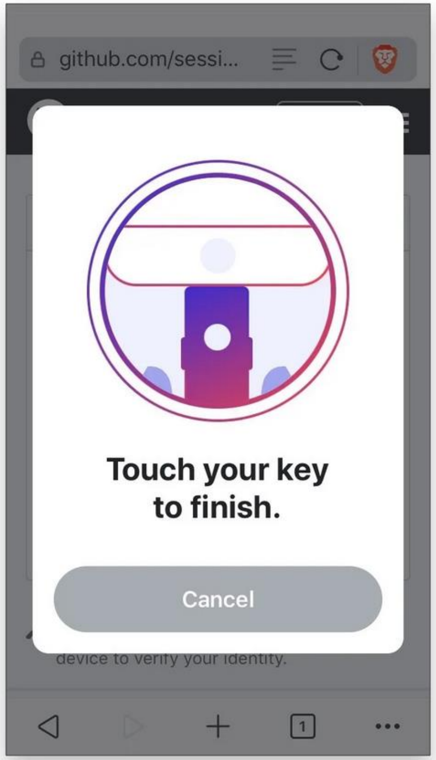


WEBAUTHN IN PRACTICE

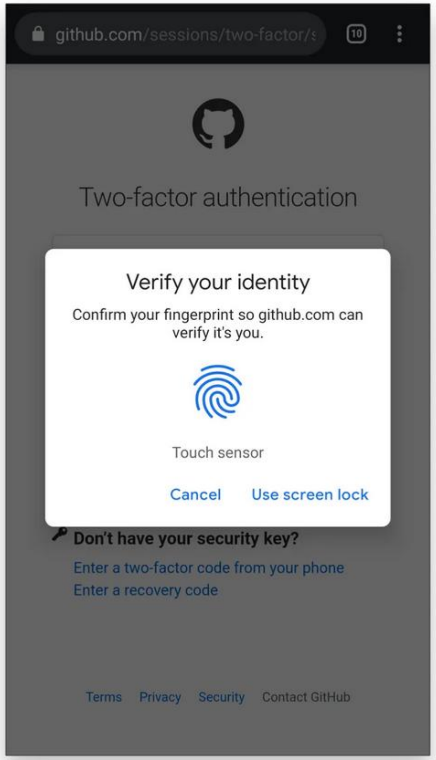
Windows MS Edge



iOS Brave



Android Chrome



You can see the status of supported browsers at: https://github.com/webauthn-open-source/fido2-webauthn-status

WEBAUTHN SUMMARY

- Basically authentication by possession (device that has private key)
- Advantages
  - Fast authentication
  - No secret information at the server
  - Long random secrets
  - Protects against phishing, since challenge includes domain
- Disadvantages
  - If private key is lost immediate access to account
  - If private key is lost, might be difficult to regain access

FIDO2 IN COMPARISON

Scheme	Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-trusted-Third-Party	Requiring-Explicit-Consent	Unlinkable
	Usability								**Deployability**						**Security**										
Password	○	○	●	○	●	●	◐	●	●	●	●	●	●	●	○	◐	○	○	○	○	○	●	●	●	●
1FA	●	● (shaded)	○ (shaded)	●	●	●	●	○ (shaded)	●	◐	○ (shaded)	● (shaded)	● (shaded)	● (shaded)	● (shaded)	● (shaded)	● (shaded)	● (shaded)	●	● (shaded)	● (shaded)	○	● (shaded)	●	● (shaded)

● = offers benefit; ◐ = almost offers benefit; ○ = does not offer benefit
■ = depends only on FIDO2 standard and is fixed for all authenticators; otherwise, depends purely or mostly on the authenticator device

Lyastani, Sanam Ghorbani, et al. “Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication.” 2020 IEEE Symposium on Security and Privacy. 2020.

FIND OUT MORE

https://gamess.dk/

WHO IS BEHIND GameSS

Partners behind the project

SDU

DTU

IT UNIVERSITY OF CPH

AALBORG UNIVERSITET

Happy 42

Collaborators

ERHVERVSAKADEMI AARHUS

KEA KØBENHAVNS ERHVERVSAKADEMI

Dania ERHVERVSAKADEMI

UCN

UCL Erhvervsakademi og Professionshøjskole

Supported by

Uddannelses- og Forskningsministeriet

Ministry of Higher Education and Science Denmark

WHO IS BEHIND GameSS

Partners behind the project

SDU

DTU

IT UNIVERSITY OF CPH

AALBORG UNIVERSITET

Happy 42

Collaborators

ERHVERVSAKADEMI AARHUS

KEA KØBENHAVNS ERHVERVSAKADEMI

Dania ERHVERVSAKADEMI

UCN

UCL Erhvervsakademi og Professionshøjskole

Uddannelses- og Forskningsministeriet

Ministry of Higher Education and Science Denmark