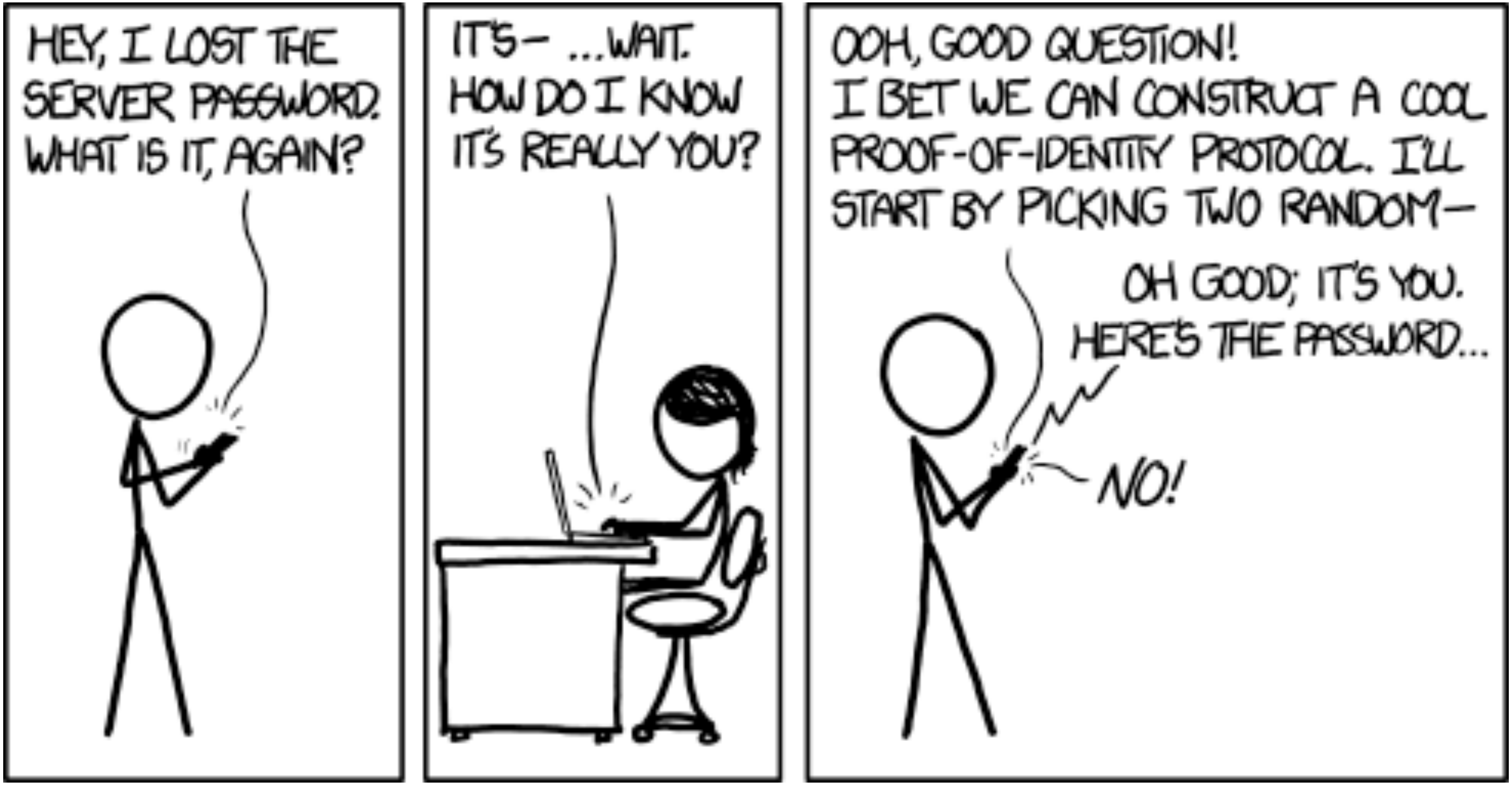
CS 55:
Security and Privacy

Identification and Authentication


HEY, I LOST THE SERVER PASSWORD. WHAT IS IT, AGAIN?
IT'S— ...WAIT. HOW DO I KNOW IT'S REALLY YOU?
OOH, GOOD QUESTION! I BET WE CAN CONSTRUCT A COOL PROOF-OF-IDENTITY PROTOCOL. I'LL START BY PICKING TWO RANDOM—
OH GOOD; IT'S YOU. HERE'S THE PASSWORD...
NO!

Source: https://xkcd.com/1121/

Anything your computer can do for you
it can potentially do for someone else
- Alan Cox

Source: *Security Engineering*, 3rd Edition, Anderson

Big idea: allow legitimate users in, keep others out

Discussion

What is the difference between:

- Identification
- Authentication
- Authorization?

Agenda

➡ 1. Lessons from my military days

2. Identification, Authentication and Authorization

3. Multi-factor authentication

What do to if you are ever a hostage

Rescuers have an identification problem

Image: https://www.buildings.com/article-details/articleid/14893/title/hostage-prevention-101/viewall/true

NEOs

Rescuers have an authentication (and authorization) problem

Image: https://www.ready.marines.mil/Stay-Informed/Emergency-Actions/Noncombatant-Evacuation-Operations-NEO/

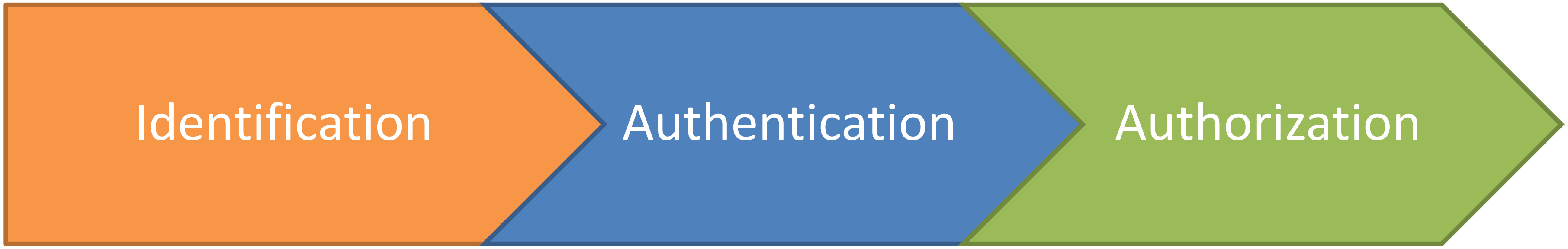
Agenda

1. Lessons from my military days

2. Identification, Authentication and Authorization

3. Multi-factor authentication

Access proceeds from Identification to Authentication to Authorization



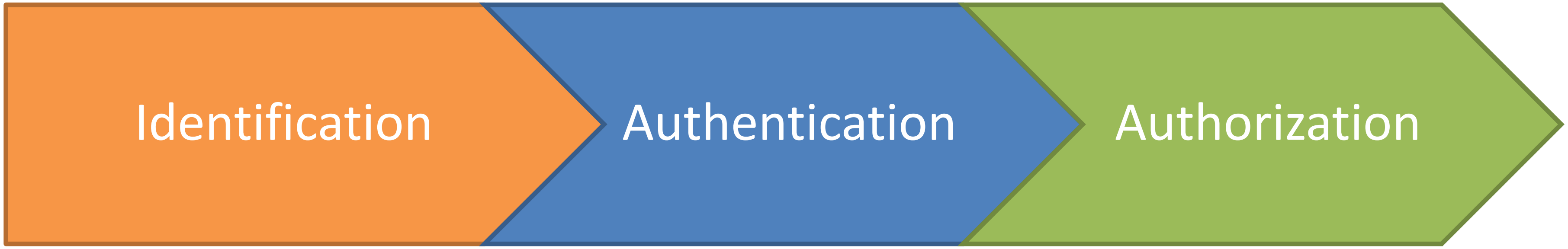
Users claims their identity; they asset who they are

Example: provide a user ID or biometric

Identity is public (anyone can claim to be a person)

Adapted from https://securityboulevard.com/2020/06/authentication-vs-authorization-defined-whats-the-difference-infographic

Access proceeds from Identification to Authentication to Authorization



Identification

Authentication

Authorization

Users claims their identity; they asset who they are

Example: provide a user ID or biometric

Identity is public (anyone can claim to be a person)

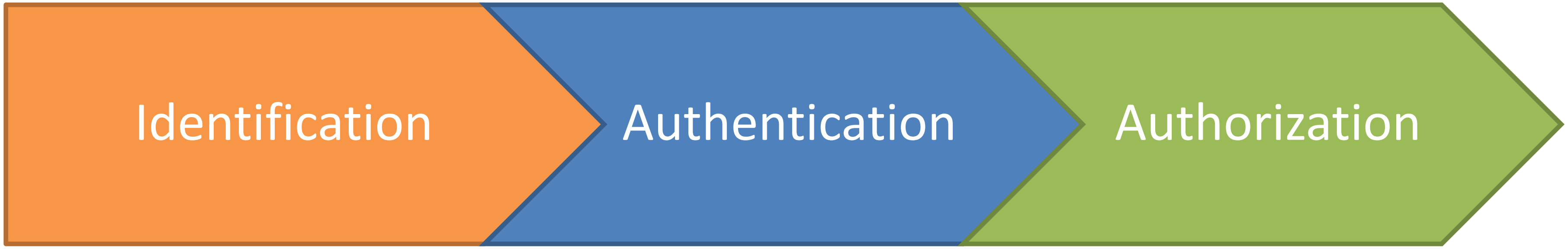
Verifying users by confirming they are who they say they are

Could be done by confirming password matches user ID

Authentication is private

Adapted from https://securityboulevard.com/2020/06/authentication-vs-authorization-defined-whats-the-difference-infographic

Access proceeds from Identification to Authentication to Authorization



Users claims their identity; they asset who they are

Example: provide a user ID or biometric

Identity is public (anyone can claim to be a person)

Verifying users by confirming they are who they say they are

Could be done by confirming password matches user ID

Authentication is private

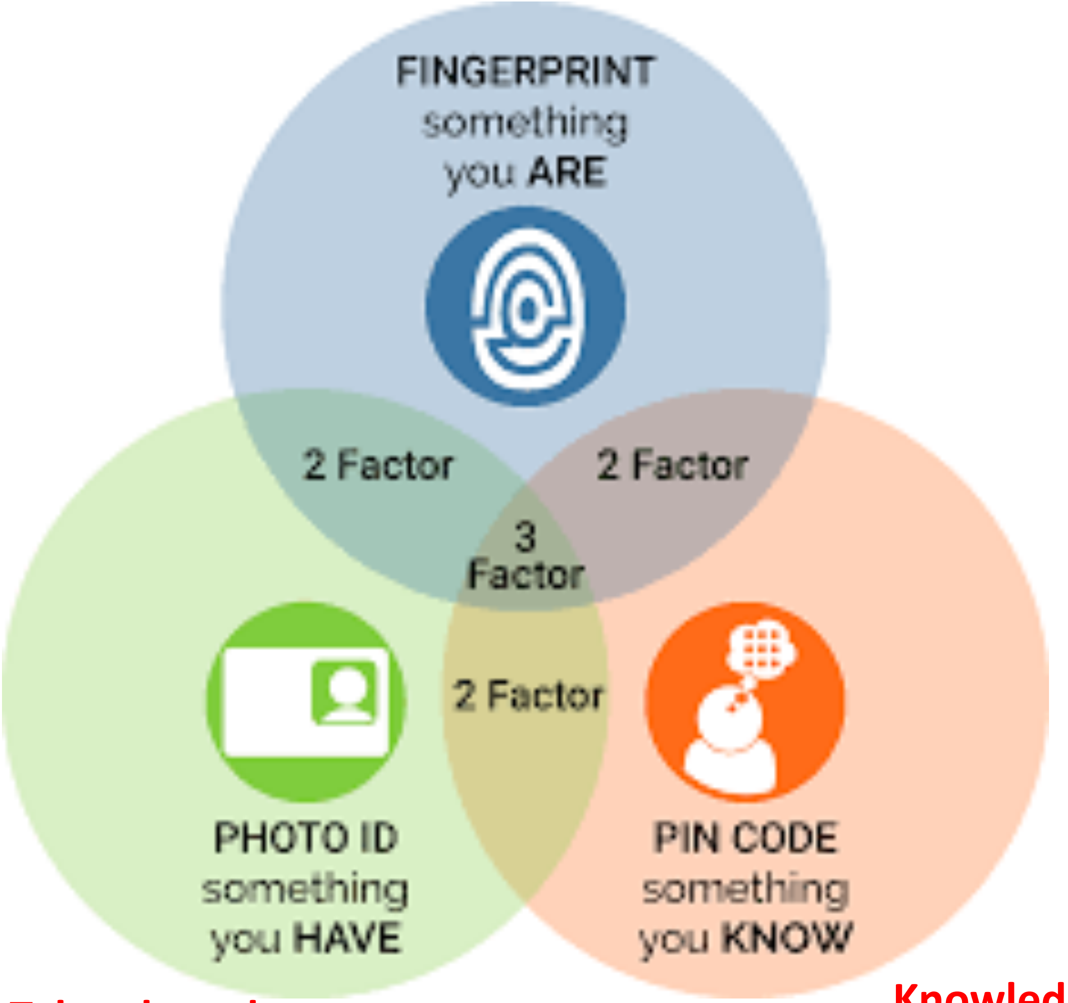
Validating the roles, permissions, and privileges assigned to a user

Performed after authentication to grant or deny access rights to users for resources

Adapted from https://securityboulevard.com/2020/06/authentication-vs-authorization-defined-whats-the-difference-infographic

Authentication is often based on something you KNOW, HAVE, or ARE

Biometric-based



Token-based

Knowledge-based

Multi-factor authentication uses two or more of these

Examples you use?

https://www.borer.co.uk/borer-technology/multi-factor-identity-authentication/

Discussion: what are the shortcommings of using passwords for authentication

Password shortcomings

What are some issues with using passwords for authentication?

- Easily guessed or hard to remember
- Must remember password for multiple systems (leads to reuse)
- Users write them down (sticky note easily observed)
- Recall is harder than recognition
- Password recovery issues (easy if you know people, harder online)
- Disclosure: once someone else knows password, they can use it or change the password to a new one!
- Cannot forget password on demand (rubber hose attack)
- Can lead to loss (Bitcoin wallet – forget password on bitcoin gone)

Should passwords be changed frequently?

Passwords can be guessed given enough time, counter measures are possible

Password guessing approaches:

- Dictionary/rainbow table attacks
- Inferring likely passwords for a particular user (OSINT)
- Credential stuffing (attacks password re-use)
- Brute force

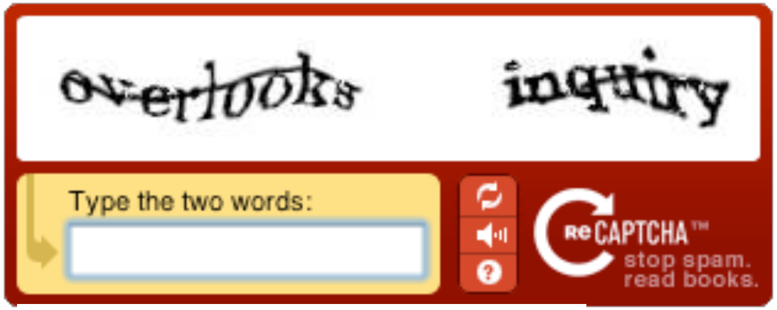
Counter measures

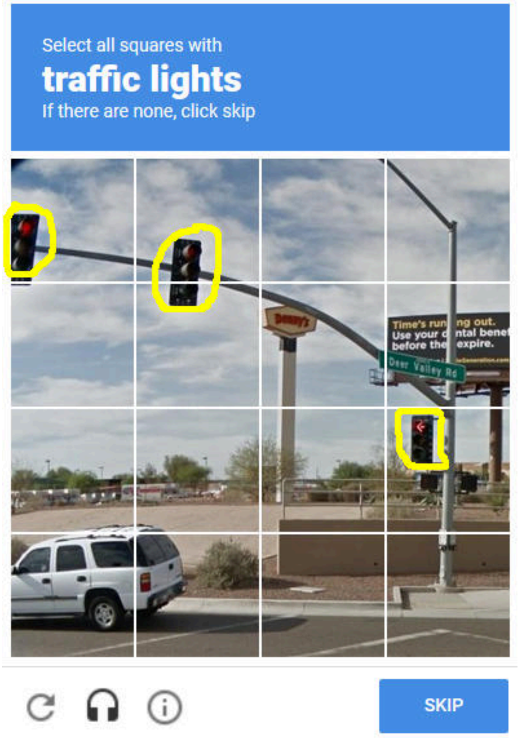
- DO NOT STORE PASSWORD IN PLAIN TEXT, store password hash
- Use hash function that is slow to compute (not bad for users)
- Add salt (defeat rainbow tables) and pepper (defeat dictionary)
- Provide exponential back off/lock out for online guessing (but this could be turned into a DOS attack!)
- Use CAPTCHA along with each guess

Adapted from *The Craft of System Security* by Smith and Marchesini

CAPTCHAs can help default automate attacks against online systems

Completely Automated Public Turing test to tell Computers and Humans Apart




Select all squares with
traffic lights
If there are none, click skip

http://www.captcha.net/

Original design relied on fact that humans can easily read text with distractors, but machines could not

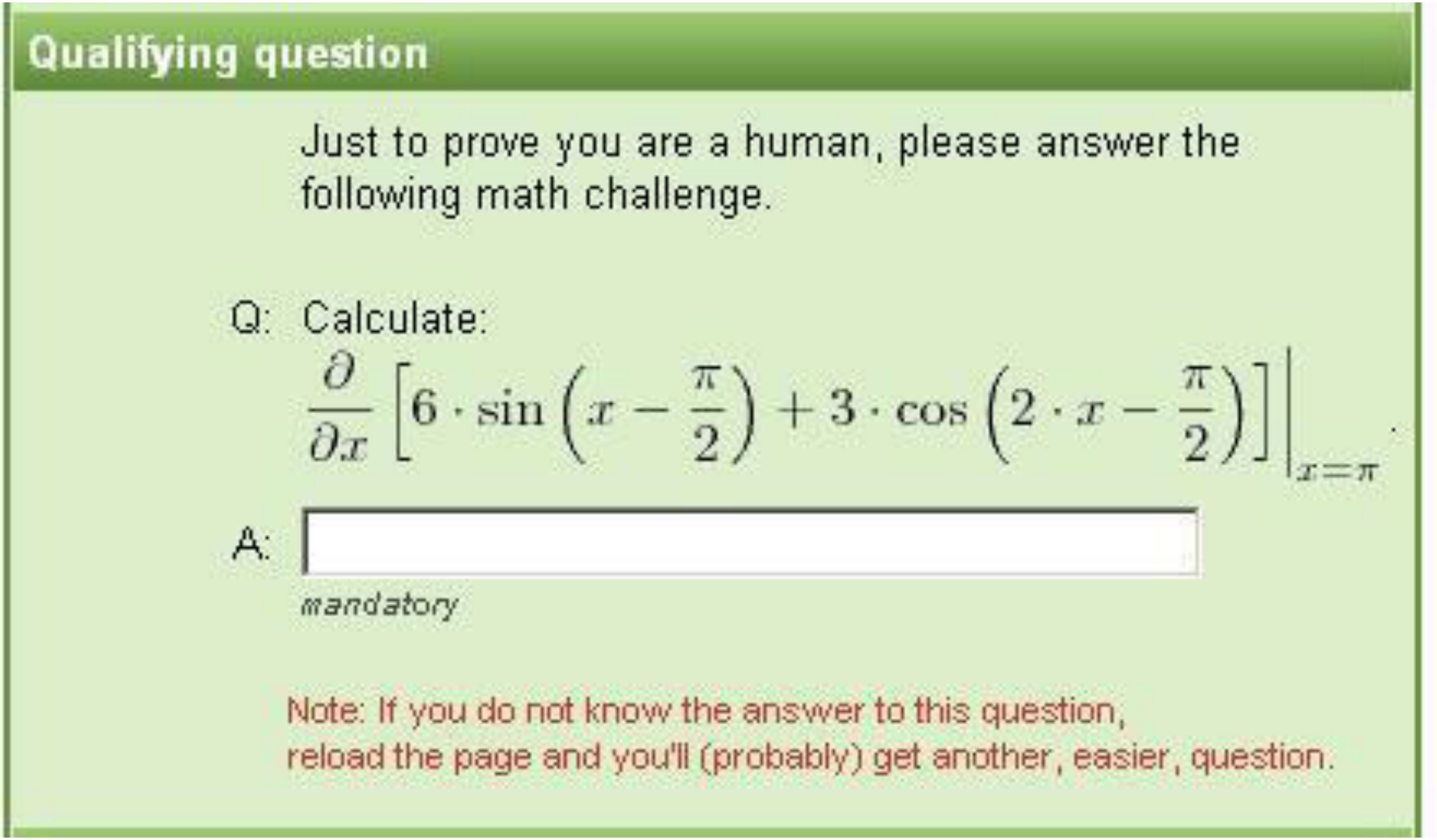
- Amazon Mechanical Turk
- ML advances reducing effectiveness

re-CAPTCHA design asks users to identify objects in image

- Harder for machines (for now)
- Helps Google with image recognition ML

Uses?

Here is another version of a CAPTCHA…



Qualifying question

Just to prove you are a human, please answer the following math challenge.

Q: Calculate:

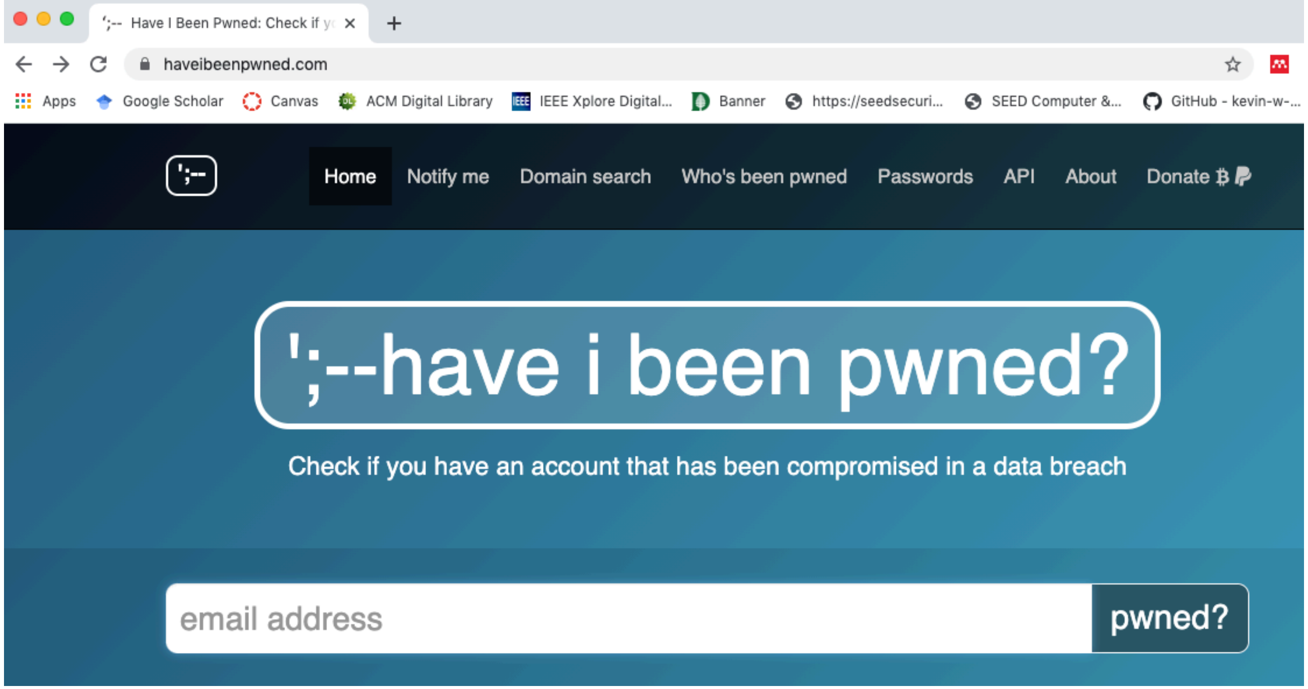
$$\frac{\partial}{\partial x}\left[6\cdot\sin\left(x-\frac{\pi}{2}\right)+3\cdot\cos\left(2\cdot x-\frac{\pi}{2}\right)\right]\Bigg|_{x=\pi}.$$

A:

mandatory

Note: If you do not know the answer to this question,
reload the page and you'll (probably) get another, easier, question.

Haveibeenpwned.com can check if a password has been in a breach



';--have i been pwned?

Check if you have an account that has been compromised in a data breach

email address

pwned?

Passwords from

- 481 breaches
- 10,199,352,448 user accounts
- 572,611,621 passwords

Check if email[1] or password[2] has been pwned

Has API you can use in your sites for:

- User registration – check if password burned
- Password change – check if new password burned
- Login – check if password is newly burned

[1] https://haveibeenpwned.com/
[2] https://haveibeenpwned.com/Passwords
Data as of Oct 17, 2020

Troy Hunt offers some useful advice regarding authentication and passwords

Authentication should be more than a binary state

- If the user has tried to login 3 times, show a captcha, lock after 5 attempts
- If logging in with a new browser from a new country, perhaps don't give unfettered access to everything

Longer passwords are (usually) stronger

- Don't limit passwords to say 8-10 characters, why limit at 10? NIST says at least 64 characters (it all hashes down to a fixed length anyway)

Special characters

- All printable characters (including space) should be allowed in a password
- Should not impose other composition rules (e.g., requiring a mix of different character types or prohibiting consecutively repeated characters) for memorized secrets (goes against conventional wisdom, but "Password!" would be ok)

Do not use password hints (e.g., my name, usual, password, email)

Use password managers

- They pick strong, random passwords
- Do not re-use passwords

Adapted from https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/

Troy Hunt offers some useful advice regarding authentication and passwords

Do not mandate password changes

- People just increment a number at the end of their password
- Change when you have a suspicion of compromise

Notify users of abnormal behavior

- Example: Dropbox emails you when a new computer accesses your files

Block previously breeched passwords

- Can use haveibeenpwnd.com API to check

Use multi-factor authentication

Adapted from https://www.troyhunt.com/passwords-evolved-authentication-guidance-for-the-modern-era/

OWASP has additional advice for developers

Use Bcrypt unless you have a good reason not to

- Bcrypt has been vetted (do not roll your own crypto!)
- Takes a long time to compute hash (ok for one user, bad for adversary trying millions of possibilities)

Set a reasonable work factor for your system

- Work factor = number of iterations of hashing algorithm
- Too low: doesn’t slow down adversaries enough
- Too high: takes too long for users
- Somewhere around 10 to 12 generally recommended

Use a salt (modern algorithms do this for you automatically)

- Each user assigned a different random string
- Append to password before hashing to defeat rainbow tables
- (salt stored in plaintext in database)

Consider using a pepper to provide an additional layer of security

- Secret value appended to password+salt to defeat dictionary attacks

Adapted from https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

Password entry systems can leak information unnecessarily

```
Welcome to XYU Computing Services
Enter username: foople
*** Unknown username – Retry

Enter username:
```

Adapted from Prof Palmer CS55 lecture notes

Password entry systems can leak information unnecessarily

```
Welcome to XYU Computing Services
Enter username: foople
Enter password: *******
*** Incorrect password
*** Attempt 1 of 3

Enter username:
```

Adapted from Prof Palmer CS55 lecture notes

Password entry systems can leak information unnecessarily

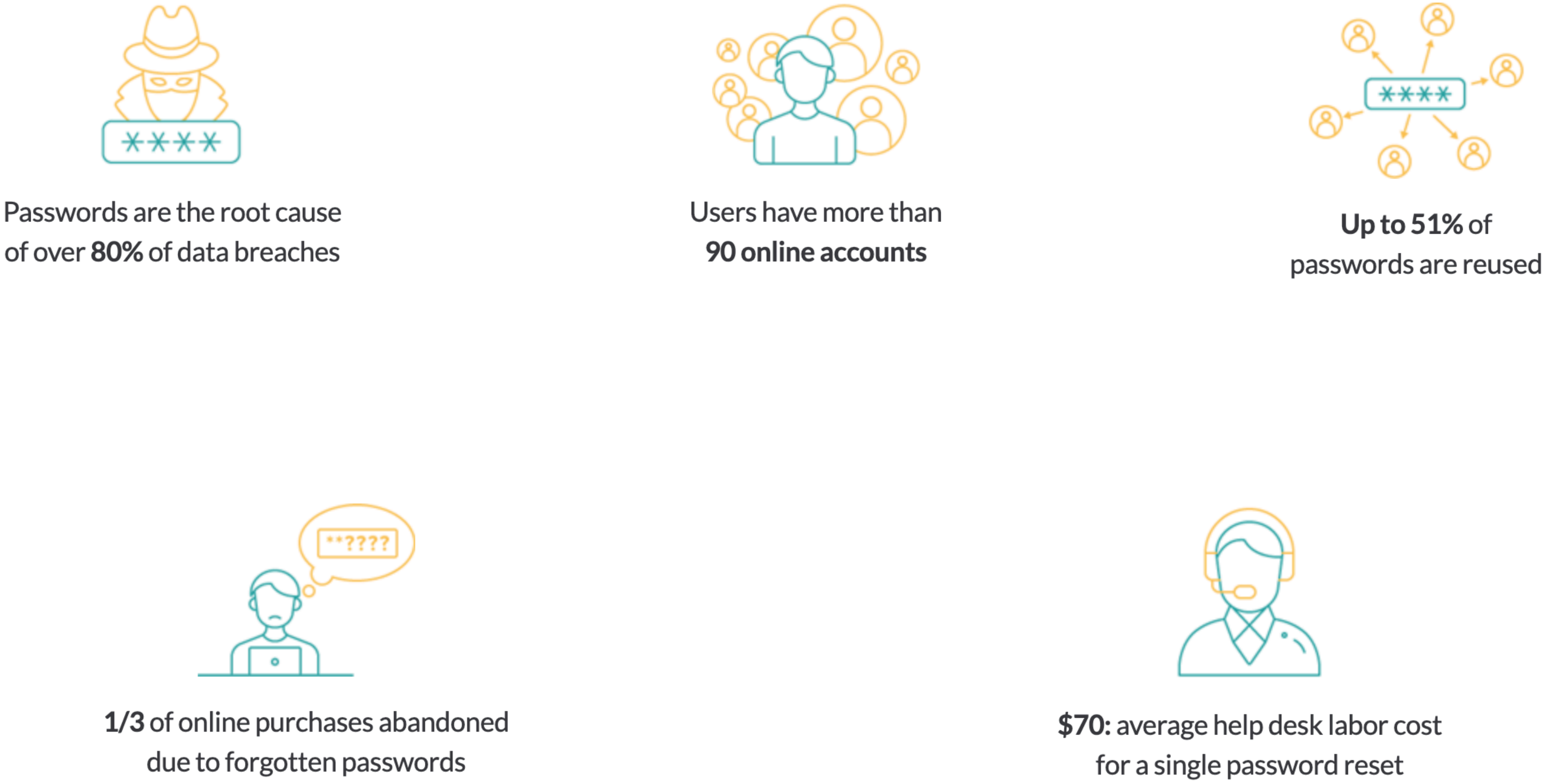
```
Welcome to XYU Computing Services
Enter username: foople
Enter password: *******
*** Authentication failed
*** Attempt 1 of 3

Enter username: foople
Enter password: ********
*** Authentication succeeded

$ _
```

Adapted from Prof Palmer CS55 lecture notes

Discussion

Will passwords ever go away?

What would be needed for them to go away?



Passwords are the root cause of over **80%** of data breaches

Users have more than **90 online accounts**

Up to 51% of passwords are reused

1/3 of online purchases abandoned due to forgotten passwords

$70: average help desk labor cost for a single password reset

Image: https://fidoalliance.org/what-is-fido/

Agenda

1. Lessons from my military days
2. Identification, Authentication and Authorization
3. Multi-factor authentication

Multi-factor authentication often uses tokens you HAVE

Static tokens

PANAMA TASK FORCE COMMAND POST PASS No. 491
Quarry Heights
This will identify Date: May 12, 1943.
Name: [illegible]
Org.: Sig. Serv. Co.
[illegible]
Issued by: [illegible]
Rank & Org.: [illegible]

Certificate of Birth

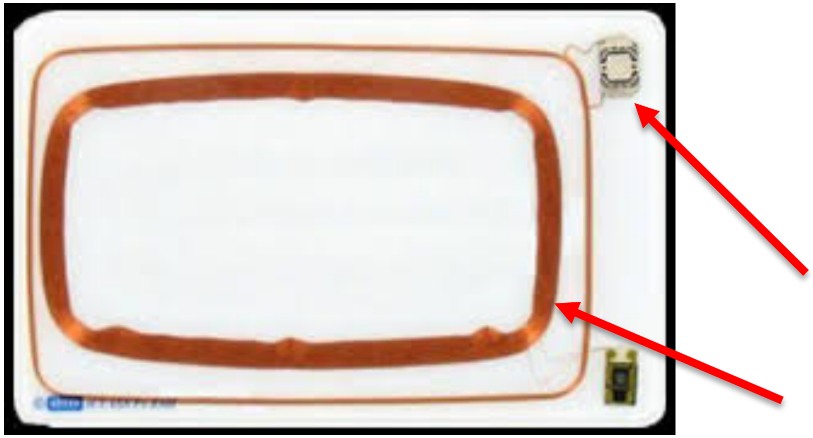
Static tokens do not change value over time

Ideally

- Difficult to duplicate
- Often issued by an authority (vets who gets a token)
- Easily recognized as valid
- Easily revoked

Adapted from Prof Palmer CS55 lecture notes

Example: proximity cards are often used for physical access control



Small chip provides ID

Antenna gathers power from reader

Prox cards

- Short range
- Cards are passive
  - No power in card itself
  - Card powered by reader
- Cards often only have an ID (not more computational power)
- Reader reads card ID
  - Checks if access is allowed
  - Opens door if authorized access

Adapted from https://www.professormesser.com/security-plus/sy0-501/access-control-technologies/

Prox cards can be captured by a mobile battery powered reader

https://www.youtube.com/watch?v=etUdZFeQgmw&start=87

Prox cards can be captured by a mobile battery powered reader

Hunt Pad Attacks!
RFID
Taking the long-range reader on the offensive!

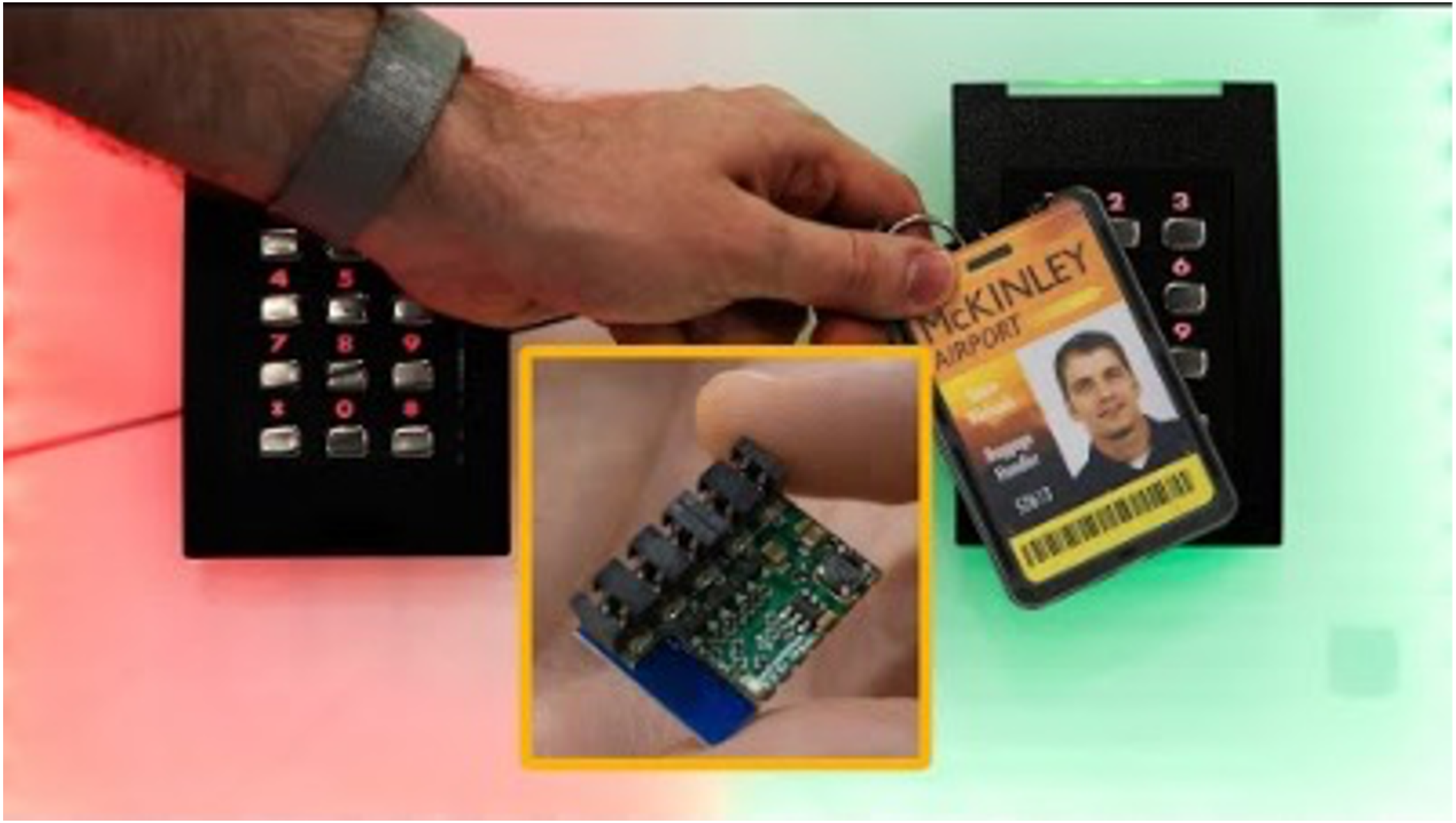
https://www.youtube.com/watch?v=etUdZFeQgmw&start=87

Countermeasures?

Credentials can also be harvested with an ESPKey

https://www.youtube.com/watch?v=Ccm1caB6bao&start=642

Credentials can also be harvested with an ESPKey



https://www.youtube.com/watch?v=Ccm1caB6bao&start=642

Countermeasures?

Smart cards are sometimes used for access to computers

Smart cards

- Card has integrated circuit and digital certificate
- Must have physical access to computer and have smart card
- Normally used with PIN (something you know) or biometric (something you are)
- US Government uses this technology (PIV – Personal Identity Verification; DOD calls it CAC – Common Access Card)
- Credit cards are another example

Adapted from https://www.professormesser.com/security-plus/sy0-501/access-control-technologies/

Dynamic tokens can change value over time

Dynamic tokens

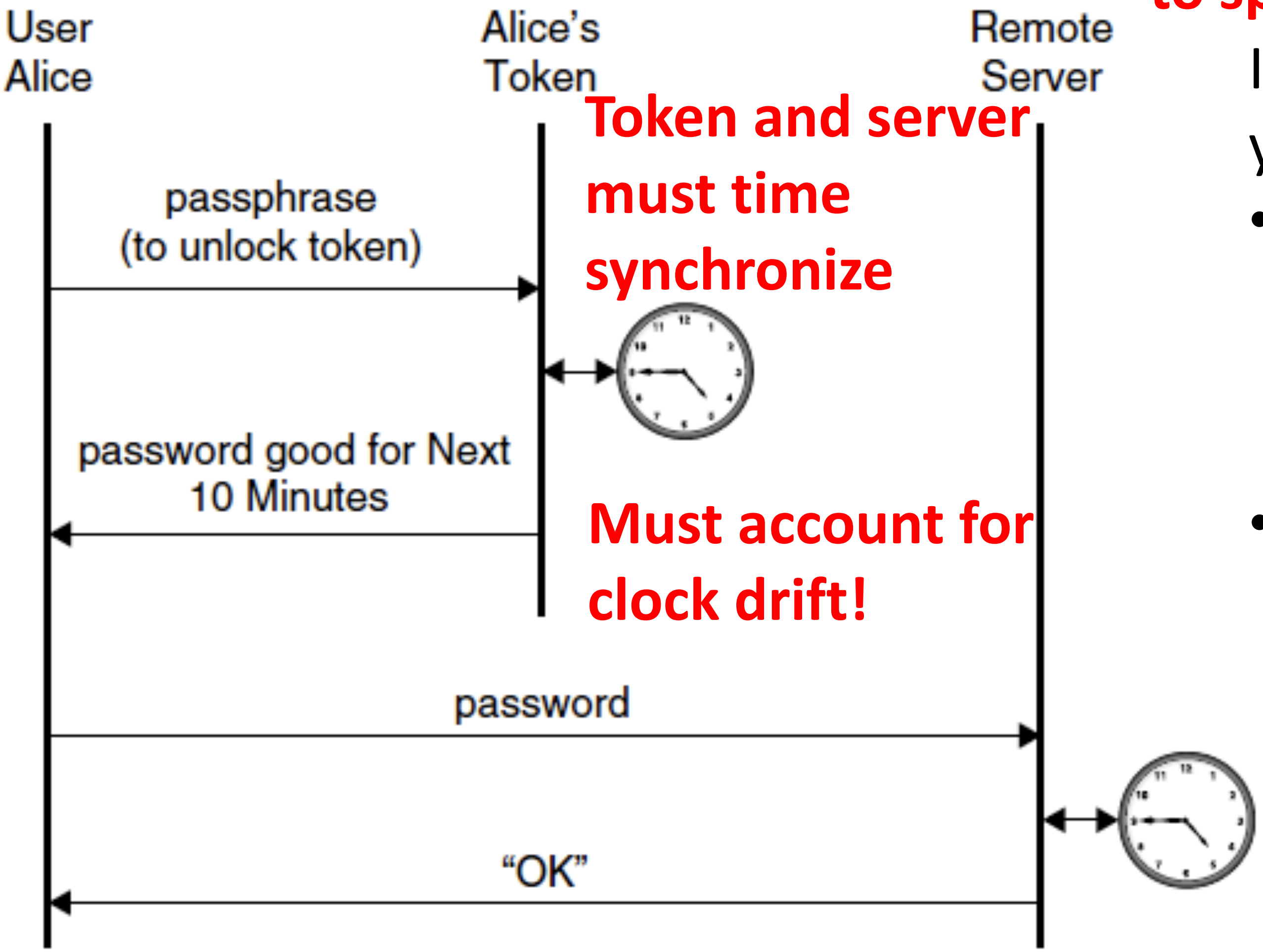
Dynamic tokens have some computational capabilities

Change internal state over time

Adapted from Prof Palmer CS55 lecture notes

Dynamic tokens are often combined with something you KNOW

Simplified one-time password with clock



Token and server must time synchronize

Must account for clock drift!

Might also restrict access to specific time of day

Involves something you

- **HAVE**
  token that changes password at fixed interval
- **KNOW**
  password to access token

Dartmouth's Duo works somewhat similarly using PKI

Adapted from Prof Palmer CS55 lecture notes

Token failures can still result

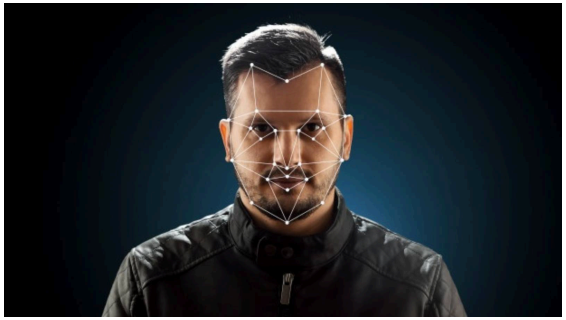
Possible token failures

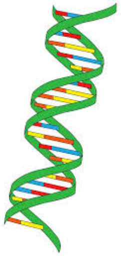
- Lost
- Stolen
- Duplicated
- Broken
- Revoked but used anyway
- Hacked

Adapted from Prof Palmer CS55 lecture notes

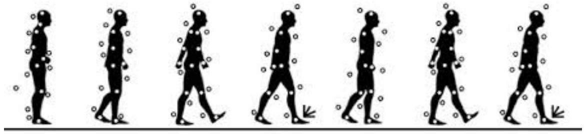
Biometrics use physiological or behavioral characteristics about you

Physiological



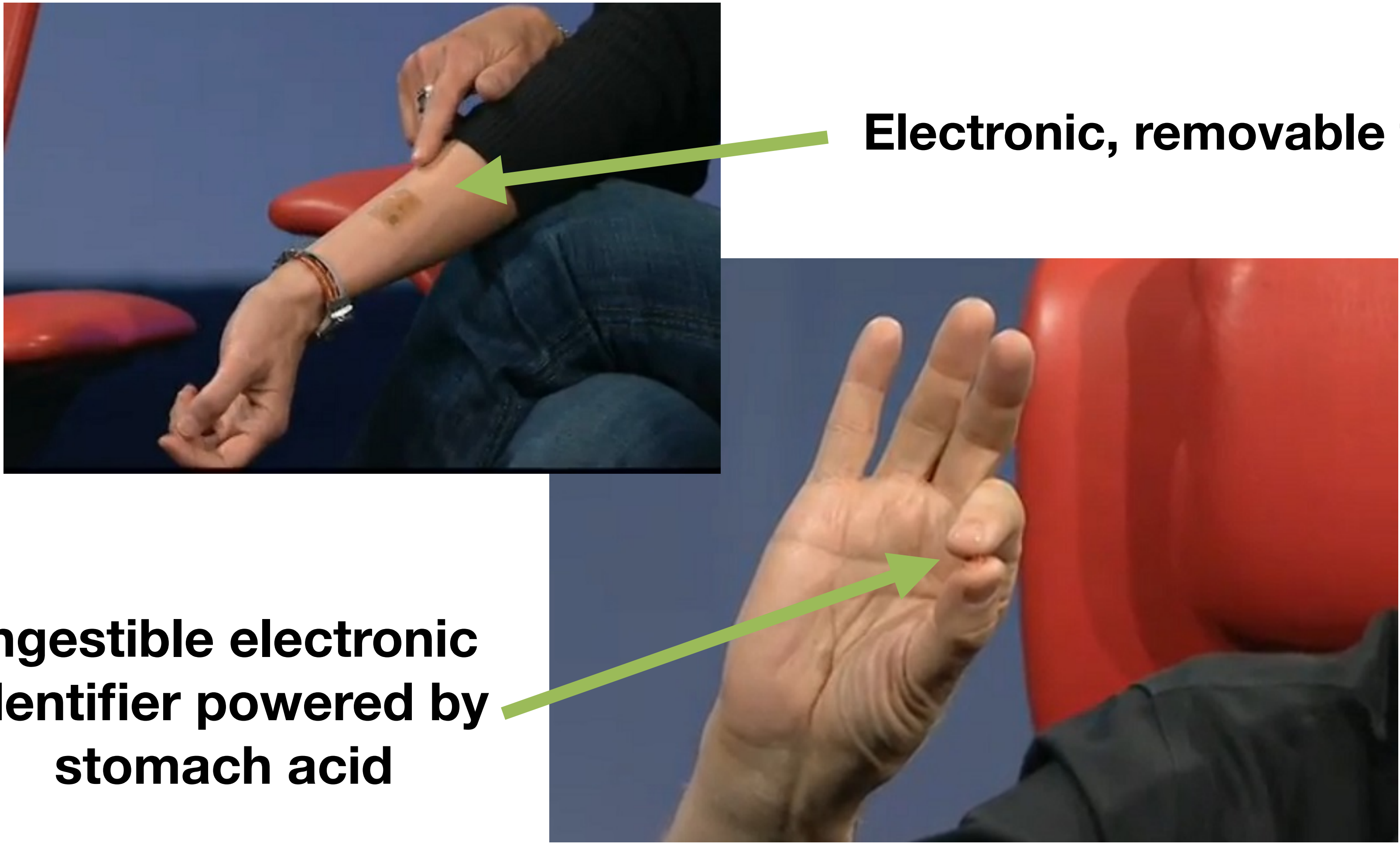


Behavioral



Adapted from Prof Palmer CS55 lecture notes

Some blur the line between tokens and biometrics



Electronic, removable tattoos

Ingestible electronic identifier powered by stomach acid

Adapted from Prof Palmer CS55 lecture notes

Biometrics authentication result in one of four cases

	The subject IS the person they claimed to be	The subject IS NOT the person they claimed to be
Test result is Positive: MATCH	a) TRUE POSITIVE	**False Accept Rate (FAR)** b) FALSE POSITIVE
Test result is Negative: NO MATCH	**False Reject Rate (FRR)** c) FALSE NEGATIVE	d) TRUE NEGATIVE

Dichotomous test: there is either a match or there is not a match

Adapted from *Security In Computing* by Pfleeger

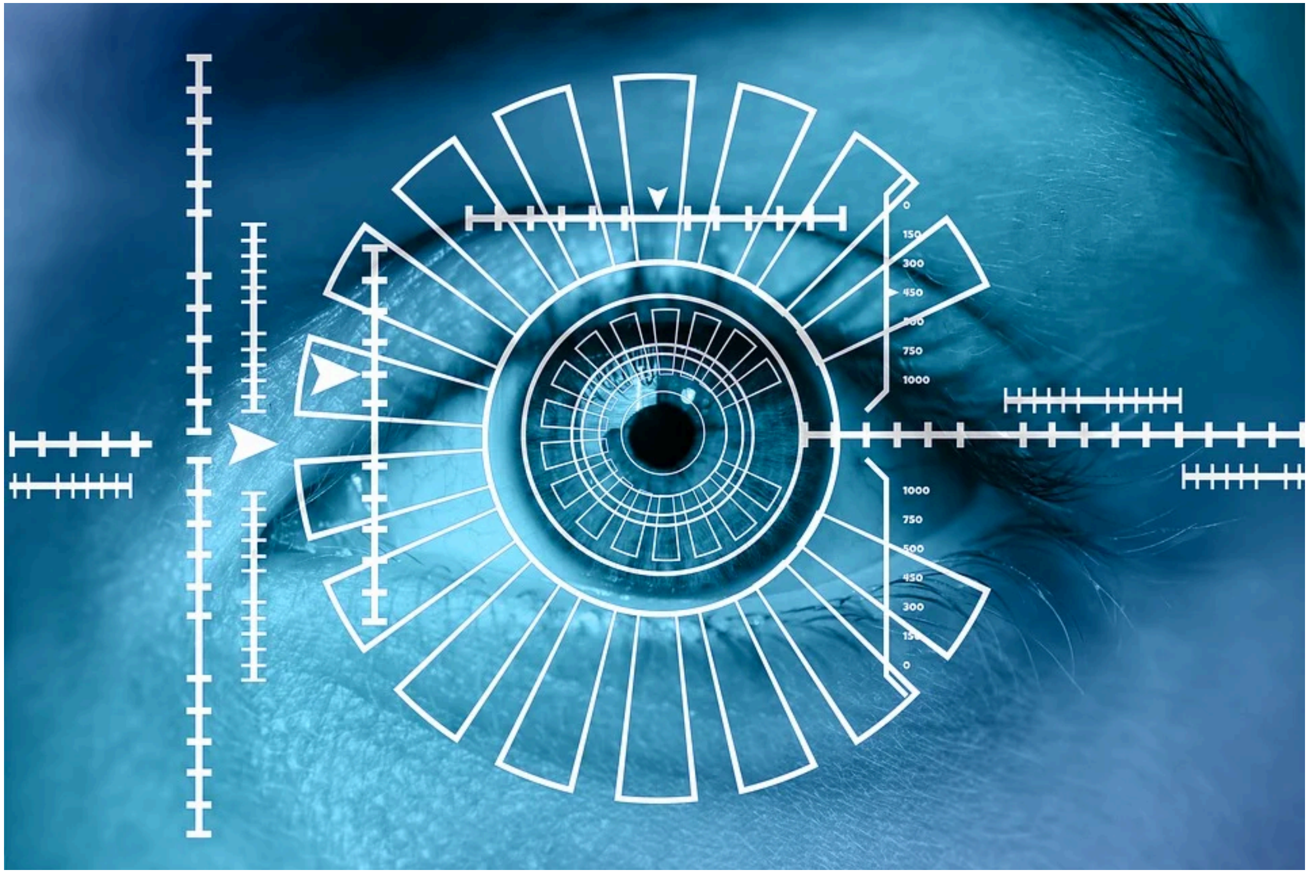
Sometimes it is tough to accurately authenticate a user

Adapted from Prof Palmer CS55 lecture notes

Aside from false readings, there can be problems with biometrics

Biometric problems

- Intrusive
- Can be expensive
- Single point of failure
- Sampling error
- Speed
- Forgery

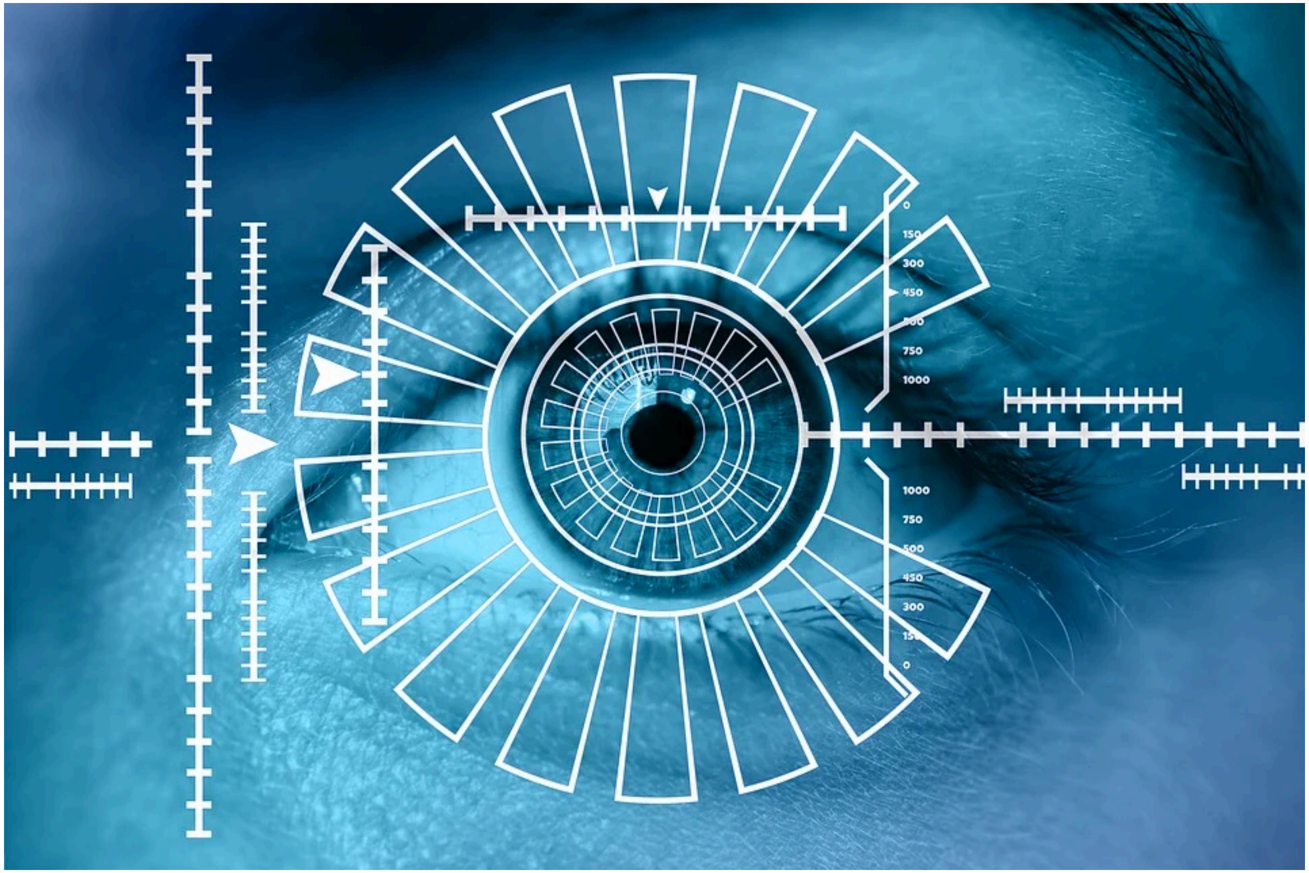


Adapted from *Security In Computing* by Pfleeger, Pfleeger, and Margulies

Aside from false readings, there can be problems with biometrics

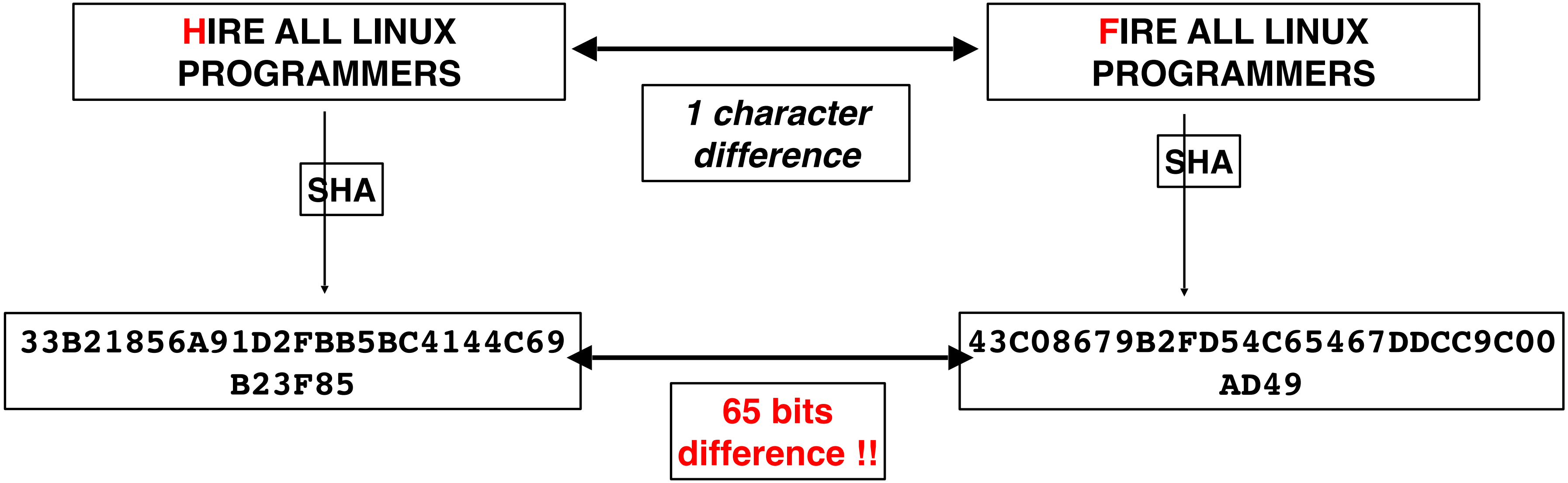
Biometric problems

- Intrusive
- Can be expensive
- Single point of failure
- Sampling error
- Speed
- Forgery
- **Not easily cancellable**



Adapted from *Security In Computing* by Pfleeger, Pfleeger, and Margulies

Can we hash a biometric such as a fingerprint?



HIRE ALL LINUX PROGRAMMERS

FIRE ALL LINUX PROGRAMMERS

1 character difference

SHA

SHA

33B21856A91D2FBB5BC4144C69 B23F85

43C08679B2FD54C65467DDCC9C00 AD49

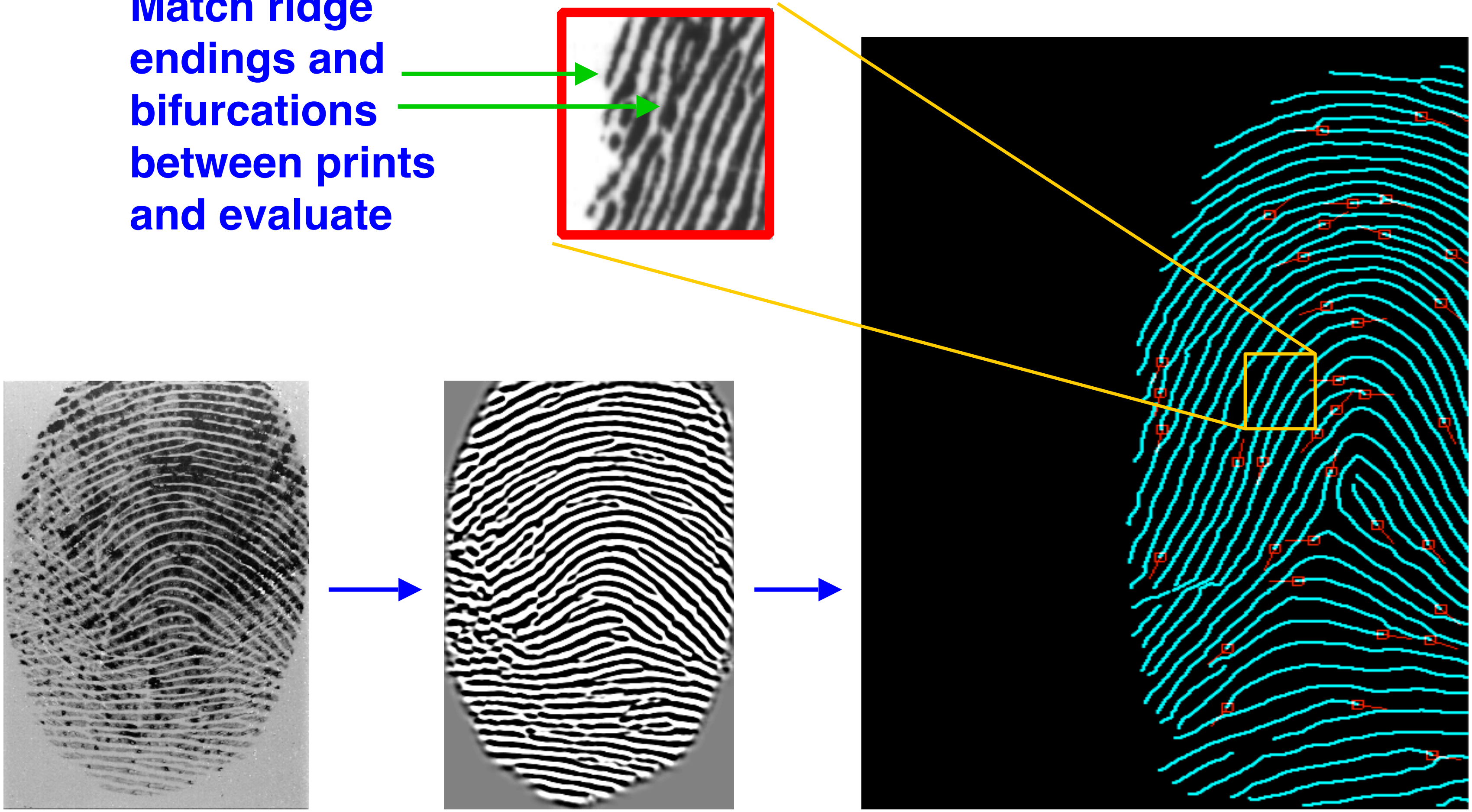
65 bits difference !!

We hash passwords so that if they are stolen the adversary does not get the plain text

What about hashing a biometric?

Adapted from Prof Palmer CS55 lecture notes

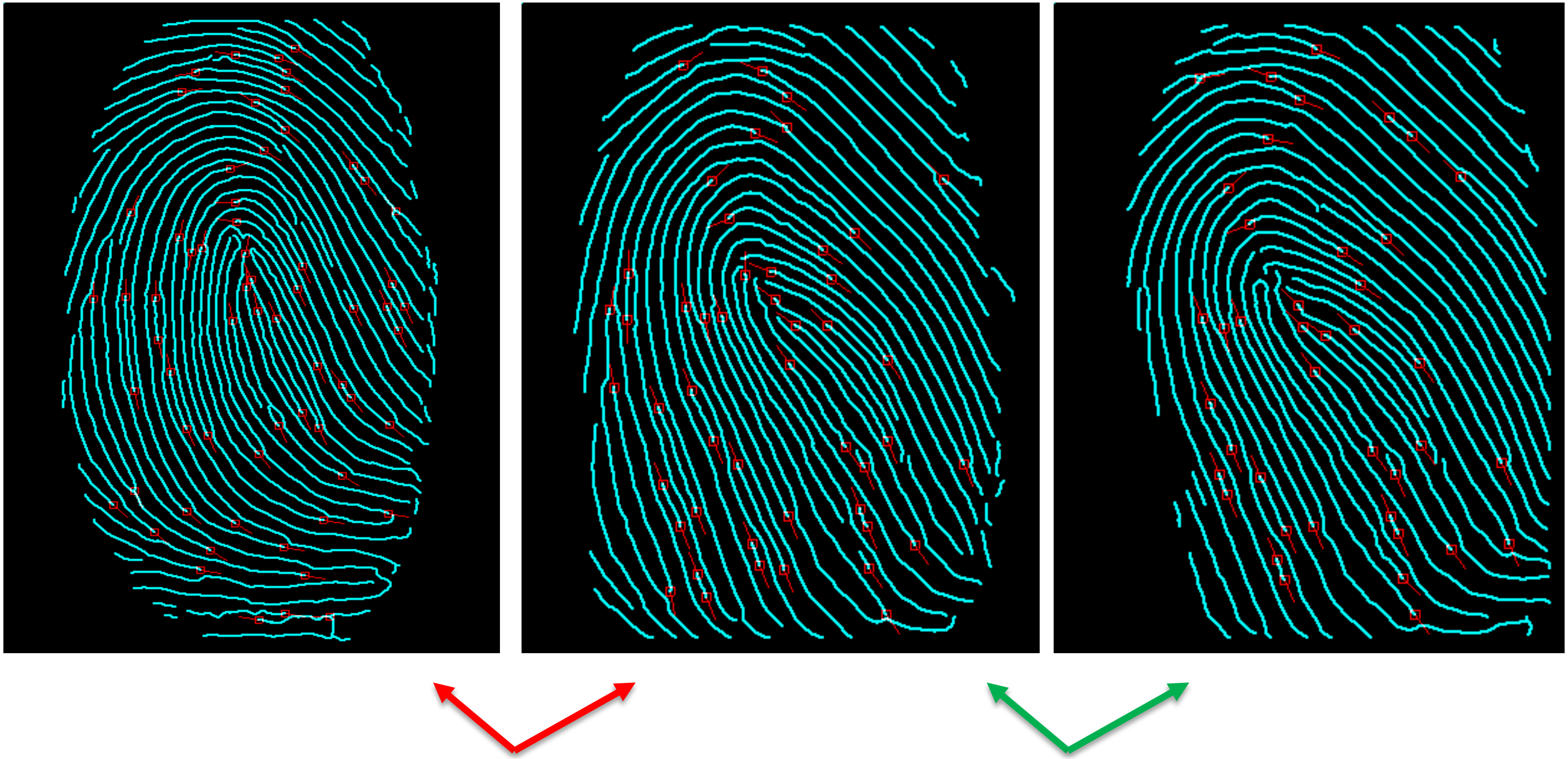
Fingerprints are primarily matched by "minutiae"



Match ridge endings and bifurcations between prints and evaluate

Adapted from Prof Palmer CS55 lecture notes

Small changes in minutiae identify individuals



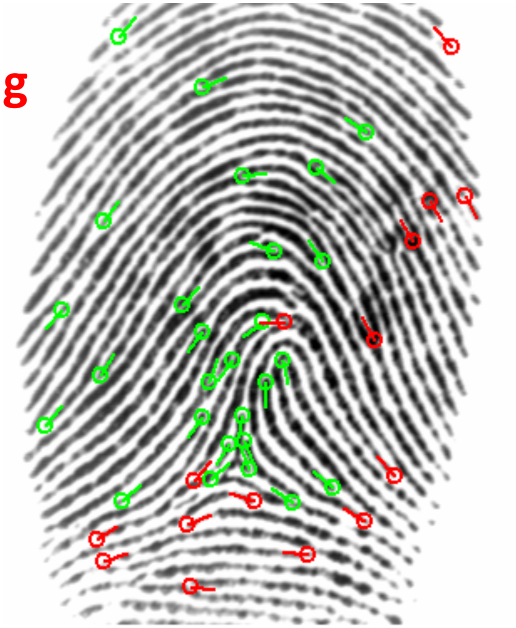
No match

Match

Adapted from Prof Palmer CS55 lecture notes

Goal: authenticate fingerprint if enough points match

When enrolling user capture their points



one-way hash

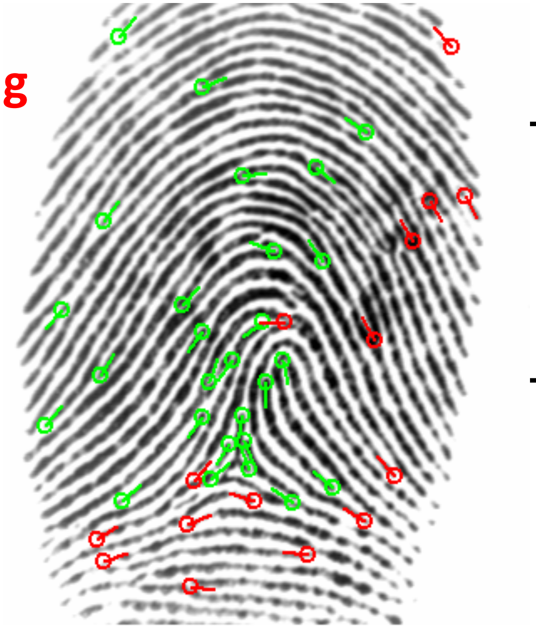
`F313C86188DDE96bD48AD58CDECDB9E8`

Hash points and save to database

Adapted from Prof Palmer CS55 lecture notes

Goal: authenticate fingerprint if enough points match

When enrolling user capture their points

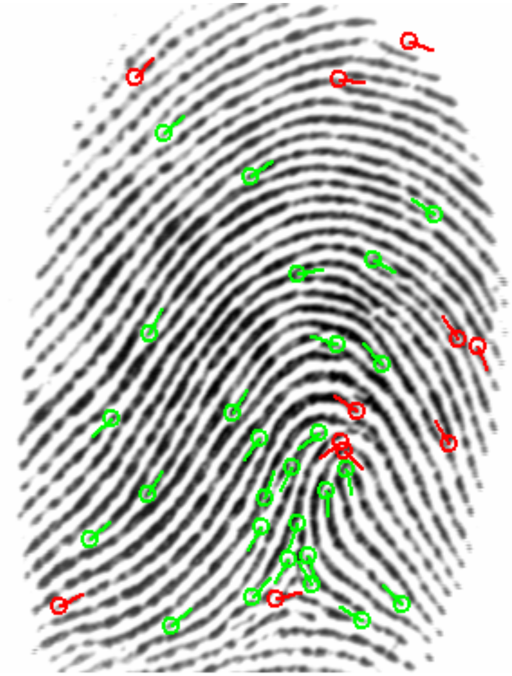


26 points match

OK

15 points don’t match

To authenticate later, measure points and compare with database



Goal: accept if enough points match

one-way hash

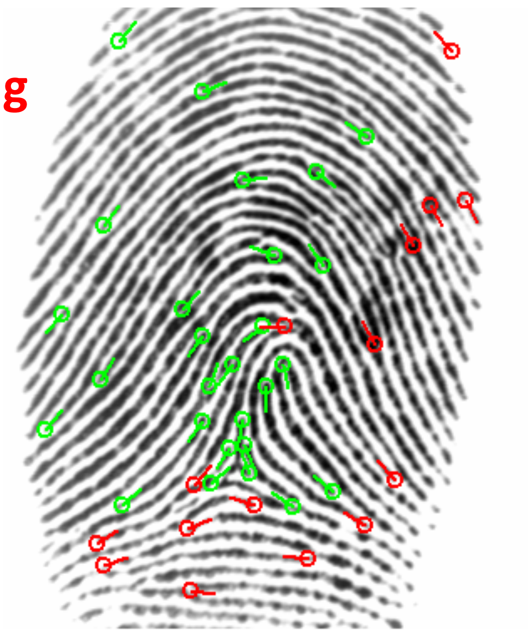
F313C86188DDE96bD48AD58CDECDB9E8

Hash points and save to database

Adapted from Prof Palmer CS55 lecture notes

It is often difficult to hash biometrics

When enrolling user capture their points

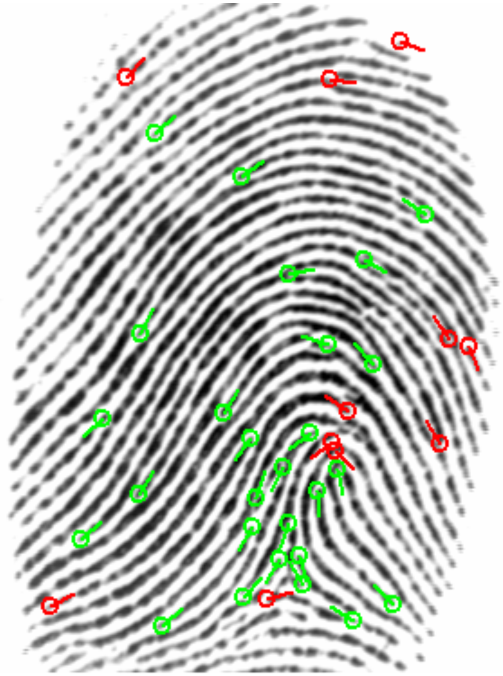


26 points match

OK

15 points don’t match

To authenticate later, measure points and compare with database



Goal: accept if enough points match

Hash and compare

one-way hash

one-way hash

F313C86188DDE96bD48AD58CDECDB9E8

80BC979099C2FA643E4C5432A03E01B8

Hash points and save to database

Not even close! (by design)

It is often difficult to hash biometrics

Adapted from Prof Palmer CS55 lecture notes

Once a user is authenticated security controls can limit what they can do

Technical controls

Used to limit the impact or prevent a security incident, may log events

- Controls implemented using systems
- Operating system controls
- Firewalls, IPS/IDS

Administrative controls

Controls that determine how people act

- Security policies
- Standard operating procedures

Physical controls

Limit access to physical areas

- Locks
- Fences
- Mantraps

WARNING
Restricted Area
It is unlawful to enter this area without permission of the Installation Commander.
Sec. 21, Internal Security Act of 1950; 50 U.S.C. 797
While on this Installation all personnel and the property under their control are subject to search.
Use of deadly force authorized.

Adapted from https://www.professormesser.com/security-plus/sy0-501/security-controls-2/